HUMAN INTERACTIONS ON ONLINE SOCIAL MEDIA
Collecting and Analyzing Social Interaction Networks

Fredrik Erlandsson
Human Interactions on Online Social Media

Fredrik Erlandsson
Human Interactions on Online Social Media
Collecting and Analyzing Social Interaction Networks

Fredrik Erlandsson

Doctoral Dissertation in Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN
Abstract

Online social media, such as Facebook, Twitter, and LinkedIn, provides users with services that enable them to interact both globally and instantly. The nature of social media interactions follows a constantly growing pattern that requires selection mechanisms to find and analyze interesting data. These interactions on social media can then be modeled into interaction networks, which enable network-based and graph-based methods to model and understand users’ behaviors on social media. These methods could also benefit the field of complex networks in terms of finding initial seeds in the information cascade model. This thesis aims to investigate how to efficiently collect user-generated content and interactions from online social media sites. A novel method for data collection that is using an exploratory research, which includes prototyping, is presented, as part of the research results in this thesis.

Analysis of social data requires data that covers all the interactions in a given domain, which has shown to be difficult to handle in previous work. An additional contribution from the research conducted is that a novel method of crawling that extracts all social interactions from Facebook is presented. Over the period of the last few years, we have collected 280 million posts from public pages on Facebook using this crawling method. The collected posts include 35 billion likes and 5 billion comments from 700 million users. The data collection is the largest research dataset of social interactions on Facebook, enabling further and more accurate research in the area of social network analysis.

With the extracted data, it is possible to illustrate interactions between different users that do not necessarily have to be connected. Methods using the same data to identify and cluster different opinions in online communities have also been developed and evaluated. Furthermore, a proposed method is used and validated for finding appropriate seeds for information cascade analyses, and identification of influential users. Based upon the conducted research, it appears that the data mining approach, association rule learning, can be used successfully in identifying influential users with high accuracy. In addition, the same method can also be used for identifying seeds in an information cascade setting, with no significant difference than other network-based methods. Finally, privacy-related consequences of posting online is an important area for users to consider. Therefore, mitigating privacy risks contributes to a secure environment and methods to protect user privacy are presented.
to Lea and Emma.
This thesis consists of in total nine publication, of which five have been submitted, peer reviewed and published in conference proceedings. Two of the publications are peer reviewed book chapters, and one is published in a scientific journal. The thesis also consists of one publication that is submitted to a scientific journal and is currently (Nov. 2017) in peer review. The publications have been written together with other colleagues from Blekinge Institute of Technology, University of California Davis and Wroclaw University of Science and Technology. The thesis material has appeared in the following publications (in chronological order):


Publication (I) deals with privacy issues identified by the authors, in which the thesis author is the main driver. Publications (III) and (IV) are related as they form part of the motivation for the data collection process discussed in publication (V) and (IX). For the publications (II), (III), and (IV) the thesis author contribute with the dataset, experiment design and the development of the SINCERE search engine described in Chapter 6. The thesis author were also highly involved in the writing of publication (II). Publication (V) is an enabling study for most of the work in this thesis, with the thesis author as the main driver and contributor of the material. For the publications (VI), (VII), (VIII) and (IX), the thesis author was the main driver, conducting and developing experiments and tools. The thesis author is also the principal driver of the writing, in these studies, together with the senior co-authors.
Related papers

The following publications are related and written by the author but are not included in this thesis (in reversed chronological order):


First and foremost, I like to thank my family, my loving wife Johanna and my two lovely daughters Lea and Emma. I’m sorry for all the time I’ve been neglecting you in favor for my job, thank you for being there for me. Also, thank you dad for your support and inspiration.

Secondly, I like to thank Henric and Felix. I admit Henric, you were right; the trip I took in 2010 was the trip that changed my life. As this trip was the initial contact with Felix and when the journey to peruse a PhD actually started. Thank you, Felix, for giving me the opportunity to work with you and your team at UC Davis in 2012.

Thirdly I like to thank Piotr, without your insights and inspiration into Network Science and Complex Networks in particular I am not sure I would have managed to finish my thesis. Also, thank you Martin for your always valuable comments and ideas of how to make good research.

Finally, I like to acknowledge the whole department of Computer Science and Engineering at BTH, you are good colleagues and friends.
## Contents

Abstract ................................................. i  
Preface ................................................... v  
Acknowledgements ................................. ix  

1 Introduction ................................. 1  
1.1 Background ........................................ 2  
1.2 Central Concepts ............................... 3  
1.3 Thesis Outline and Structure .............. 9  

2 Approach ........................................... 11  
2.1 Related Work ...................................... 11  
2.2 Aim & Scope ....................................... 14  
2.3 Research Questions ............................ 14  
2.4 Methodology ...................................... 16  
2.5 Legal and Privacy concerns ................. 18  

3 Results ............................................. 19  
3.1 Contributions .................................... 19  
3.2 Discussion ....................................... 22  
3.3 Conclusion ....................................... 24  
3.4 Future Work ..................................... 25  

Bibliography ........................................ 27  

4 Privacy Threats Related to User Profiling in Online Social Networks 37  

_Fredrik Erlandsson, Martin Boldt, Henric Johnson_

xi
7 Mining User Deliberation and Bias in Online Newsgroups: A Dynamic View 95
Teng Wang, Fredrik Erlandsson, S. Felix Wu
7.1 Introduction .................................................. 96
7.2 Related Work ................................................. 98
7.3 Identification of Deliberation and Bias ..................... 100
7.4 Predict User Deliberation and Bias .......................... 110
7.5 Conclusion and Discussion .................................. 114
7.6 Acknowledgments ............................................ 115
7.7 References .................................................. 116

8 Crawling Online Social Networks 121
Fredrik Erlandsson, Roozbeh Nia, Martin Boldt, Henric Johnson, S. Felix Wu
8.1 Introduction .................................................. 121
8.2 Related Work ................................................. 123
8.3 Requirements and challenges ............................... 125
8.4 A platform to make interactions accessible ............... 128
8.5 Prioritization of the crawling queue ....................... 132
8.6 Evaluation .................................................. 133
8.7 Conclusion .................................................. 136
8.8 Future work .................................................. 136
8.9 References .................................................. 137

9 Predicting User Participation in Social Media 141
Fredrik Erlandsson, Anton Borg, Henric Johnson, and Piotr Bródka
9.1 Introduction .................................................. 141
9.2 Related work ................................................. 142
9.3 Data model .................................................. 143
9.4 Association Rules ............................................ 144
9.5 Methodology ................................................ 146
Introduction

Online Social Media (OSM) such as Facebook, Twitter, and Instagram, are attracting increasingly more interest from Internet users. With the possibilities of being connected and interacting with each other anytime and anywhere, OSM influence peoples’ daily routines and everyday behaviors. For instance, 48% of US adults between 18 and 34 are checking Facebook the first thing they do when they wake up [1]. In addition, Facebook alone increased its number of users with 13% between 2014 and 2016, and in May 2017 there were 1.8 billion active users [1]. In total, the Internet has 3.7 billion users, i.e., 47% of the users of the Internet are active Facebook users.

Apart from changing the way people interact and communicate, OSM also provides novel means of news aggregation. Today it is possible to stay in touch with the latest news from the world by following certain people and newsgroups within OSM, i.e., the need of watching the news or reading the newspaper to keep up with the world is emerging to just checking the OSM feed. Moreover, with the growing use of OSM the democratic powers have also changed. By using OSM like Twitter and Facebook you do not have to be a reporter on a newspaper or television to form opinions and reach a critical mass. With OSM everyone has the means of publishing thoughts and opinions as a citizen journalist. There are many examples of this; the most widespread and discussed is the role social media played in the Arab Spring [2–4]. The ability for everyone to post information and form opinions are also causing issues with news validity i.e. “Fake News”. Means and methods are needed to validity the information from users’ while not “blocking” users ability for free speach.

Gathering data and the corresponding user interactions from OSM is becoming more and more interesting for researchers and businesses. The
means of observing human behavior via OSM have been called the *social media lens* by Zafarani et al. [5].

This thesis focuses on three aspects of data from OSM. Firstly we show how user generated data can be efficiently collected from the OSM site Facebook. Secondly we address privacy issues that might exist for users of OSM sites. Finally we show how the collected data (from OSM) can be used in various research settings. Including complex network setting of identifying influential users and seeds for information cascade.

1.1 Background

The computer era enables communication in new ways. The early ways of social communications using computers included bulletin board systems (BBS), USENET, America Online and CompuServe. The ancestor to the Internet as we know it, ARPANET, which is considered the first packet switched computer network was designed to enable easy civilian and military communication, mostly in the form of email. In the early stages of the Internet, social communication was mainly organized as chat rooms or simple web forums. It was first in the late 1990’s that social media sites emerged into the same form as today, enabling users to maintain a profile and create a community (much like today’s “friend list”). One of the first OSM sites was SixDegrees.com, 1997 to 2001 that was followed by Friendster and Myspace [6]. In Sweden, a very large social network site was LunarStorm, active between 2001 and 2007. Early pages of social networking such as Friendster, Myspace and the Swedish page Lunarstorm have all more or less retired and one of their successors today is Facebook [6]. Social Media and in particular OSM provide means to enable users to connect with friends and share information, i.e., a digital way to mimic the real world communication. This is often made in form of a web page, but is also supplemented with a smartphone application. The information shared were intentionally only to be available for the closed group of the users’ friends (or network of friends).

Another interesting aspect with data from OSM is the fact that it is humans that produce the data, in contrast to synthetic data. Using this data enables research areas that were hard to realize just a few years ago, e.g., big-scale user interaction analysis [7, 8] and the creation of
Social Interaction Networks (SIN) graphs [9]. A SIN graph shows the interactions between users in various communities and can for instance represent interactions of all users on one newsgroup or relating to a specific topic. This allows, for instance, researchers to develop novel applications related to social sciences.

With the high number of users in OSM, awareness of privacy related issues is of importance. However, users of OSM tend to be naive in what information to reveal (on OSM), as related to what users tend to reveal on other places (online) [10–14]. As a consequence, the awareness of the privacy related threats needs to be addressed in a clear and structured way to aid users and raise the awareness.

In December 2009 Google introduced “Personal search for everyone”, enabling custom ranking by using 57 signals [15]. This method of personal ranking is nothing unique for Google. Today most search engines and even OSM are ranking the content based on personal information. For instance, Facebook have algorithms to personalize your news feed and prioritizing posts that you most likely will interact on. This prioritization of content poses new problems, as the ranking algorithms tend to just show content in the direction of the user’s sympathies and interest. It is, therefore, hard to get a diverse picture and the opposing point of view, essential for democracy. Pariser calls this The Filter Bubble and addresses it in detail in the book with the same title [16]. We argue that it is possible to address the issues of The Filter Bubble and to reintroduce diversity in the online world by letting the users manually configure the ranking method [17].

1.2 Central Concepts

This section aims at explaining the concepts addressed in the included publication in this thesis.

Crawling is used to describe the systematic data collection process from OSM. It is important to address means of identifying data to be collected as interesting in the collection process. Efficiency is a term used in this thesis and with efficiency we address ways of using as low amount of resources as possible. We argue that software is efficient if it uses significantly lower resources than other similar software.
1. Introduction

1.2.1 Network Science

Graph theory and Network Science [18] is tightly connected, and the later can be seen as an extension to graph theory. Where Network Science deals with what can be seen as complex networks. The terminology in Network Science and graph theory is similar with the difference that a graph is called a network in Network Science. Also, vertex is called node, and edge is called link in Network Science. The basics of both graph and network science builds on a model where entities (vertex/nodes) are connected in some way to each other (this connection is called edge/link). Figure 1.1 depicts an example of a network. The figure shows the seven stars in the asterism Big Dipper (Karlavagnen in Swedish). The stars are shown as nodes and the imaginary connections are shown as links.

![Image of the network of the Big Dipper](image.png)

Figure 1.1: Example network of the asterism Big Dipper (Karlavagnen in Swedish).

In Network Science there are a few concepts that are important to understand. Firstly, there is neighbors; which denotes nodes directly connected to one particular node. For example, Dubhe have the following neighbors: Megrez, and Merak. Secondly, there is the degree centrality; which denotes the number of neighbors a node has. For instance, Megrez has a degree of 3 as it has three neighbors, (thus Megrez also has 3 connecting links). Thirdly, there is the closeness centrality, which denotes how close a node is to all other nodes in the network. In our example network is the closeness centrality of Megrez equal to \( \frac{1+1+2+2+3}{7} = 1.43 \), i.e., Megrez is directly connected to three of the nodes and have a path length (number
of links) of 2 to two of the nodes and a path length of 3 to one node in the network. Fourthly, while degree centrality and closeness centrality are used to describe properties of the nodes, is degree distribution used to describe the network. Degree distribution is often presented as a histogram showing how many nodes with each of the available degree centrality that exist in the network. In our example, the nodes have the following degree centrality 1, 2, 2, 3, 2, 2, 2. Hence we have the following degree distribution: (1 : 1), (2 : 5), & (3 : 1). Another metric used to describe the network is density. The density describes the ratio between the number of existing links and number of possible links. Density is often also called connectivity as it describes how tightly connected the nodes are. The density of a network is calculated by dividing the number of links by the maximum number of links \( \frac{n \times (n-1)}{2} \) where \( n \) is the number of nodes. Hence, a complete network where all nodes are connected to all other nodes in the network has a density of 1. In our example, shown in Figure 1.1, the density is \( \frac{7}{(7 \times (7-1))/2} = 0.33 \).

### 1.2.2 Epidemic Modeling

Epidemic modeling is a subfield in Network Science that analyses how diseases spread over the network [18–20]. Several spreading models exist within the concept of epidemic modeling. The most common is the Susceptible-Infected-Recovered (SIR) model where entities are capable of getting Infected (the entity is said to be Susceptible) at a certain probability and once Infected (or sick) there is a probability that the entity gets Recovered [18]. In addition, with the simpler Susceptible-Infected (SI) model the infected entity can’t recover. Information cascade is a form of epidemic modeling using the SI-model, which can be thought of like this: once you receive the information from someone or are infected by someone you cannot recover.

An interesting field within epidemic modeling is seed selection [21]. In which an investigation is made to determine which nodes that are best to select in the initial stage, in order to maximize spread. There are many different approaches for seed selection but it’s shown that using simple network metrics like degree centrality for ranking nodes gives a good initial pool of seeds.
1. Introduction

1.2.3 Social Media and Social Networks

One of the currently biggest OSM sites is Facebook with 1.8 billion users [1]. When users communicate with each other it is called that they are interacting. In the context of OSM, social interactions are often in a simple form, i.e., a user can often just click a button in order to interact with another user. This simple interaction can be used to indicate that a user is interested in a post or an ability to share user’s text with another community.

In this thesis we are addressing OSM with a focus on Facebook. In the scope of Facebook there are some terminology that needs explanation and clarification. Each user on Facebook has a number of friends. Users create a mutual agreement of relationship. When a user writes something on its own profile, it is called that the user posts on its wall. A user can also follow a newsgroup. On Facebook these groups are called pages, and when one follows a page it is said that the user likes that page. It is also possible for a user to post on its friend’s wall or on a page’s wall. However, some pages have restricted their page, in which only page owners and selected users are allowed to post. The main page of Facebook is called the news feed or sometimes simply just the feed. This feed contains a subset of posts from the users’ friends. It also contains a limited subset of posts from the pages the user likes. For each visible post, each user have the ability to either react, comment, write a small comment or share the post with the users’ friends. Reactions exists in the following forms: like, love, haha, wow, sad and angry. It is also possible to react on comments, which we are calling a comment-like, as a reaction on a comment previously were limited to just a like.

It is possible to create Online Social Networks from OSM. We have mainly created so called Social Interaction Networks (SIN) [9] for analyzing the collected social media data. The graph a SIN represent differs from the more traditional ego graphs [22], normally seen when considering OSM, by not using the user as the center of the graph but instead create a bipartite graph using all the interactions from one (or more) pages with users and posts as nodes and the interactions among users and posts as links. We have chosen to project this bipartite graph into a user graph, considering the interactions among users in our most recent publications [23–26]. The projected network is created as a graph $G = < \mathcal{N}, \mathcal{L} >$, with a set of nodes $\mathcal{N} = \{ n_1, \ldots, n_n \}$ to represent users and a set of links.
\[ \mathcal{L} = \{ < n_i, n_j > : n_i, n_j \in \mathcal{N} \land i \neq j \} \]

representing relationship between the user \( i \) and user \( j \). The social network of users to users is projected from the bipartite network of users and posts. Where a link \( < n_i, n_j > \) is present if both of the users \( i \) and \( j \) have interacted on the same post. An example of how this type of graph looks and how it is constructed is shown in Figure 1.2.

![Social Interactions](image)

(a) **Social Interactions**

![Ego network for corresponding posts](image)

(b) **Ego network for corresponding posts**

![Bipartite Social Network](image)

(c) **Bipartite Social Network**

![Projected Social Network](image)

(d) **Projected Social Network**

Figure 1.2: Example of interactions extracted from posts. Fig (a) shows four different posts with number of likes ('thumbs-up' icon), number of comments ('speech bubble' icon), and the age of the posts ('watch' icon). Fig (b) shows the bipartite ego networks of interactions between the six users \( (U_{1-6}) \) and the eight posts \( (P_{1-8}) \), where red links denote likes on posts and green links denote comments on posts. The users are the same on all posts. Fig (c) shows the aggregated networks of user’s interactions towards posts. Red links denote likes on posts and green links denote comments on posts. Fig (d) shows the projected social network created from the ego networks in Fig (b).
1. Introduction

1.2.4 Crawling Online Social Media Data

OSM pose interesting big data challenges regarding storage, management and analysis of users’ online activities. In January 2014, Facebook stated that they are storing data of the magnitude of exabytes ($10^{18}$ bytes), and this number is steadily growing with roughly 9 million messages sent every hour [1]. Nevertheless, the handling of storage and processing of this data is not the only challenge. There is a need to develop methods evaluating the meaning and semantic/informational value of the content. This challenge further entails studies on the efficiency of handling the informative content and the relations and interactions between the users.

In order to perform experiments we need data. The first and most straightforward way to find data is to use synthetic and generated data. This is often the case for physics and math studies just concerning the methods and algorithms, where it is sufficient with generated data according to some known method (with known properties) including [27–31]. Synthetic data allows full control of variables, and properties, and limits. Also, synthetic data is typically noise free, and without outliers. Unfortunately are these models often quite far from reality and are best suited for validating and reproducing results.

The second approach is to use real data, e.g. data from surveys with the obvious drawback of the sparseness of such data, often used in sociology. Surveying many users is hard due to the fact that they are very costly. In surveys users may also be bias in their answers [32], typically unintentionally due to the fact that users believe they act in a certain way that differs to the actual way. Users might also intentionally introduce bias. It is also possible to generate user interactions based on data of users [32]. Another way to get real data is to crawl data. With Facebook being the biggest OSM [1] it makes sense to collect and crawl data from Facebook. There are however issues with this data too, you never get the whole data. There are also online repositories for online social networks already collected by other researchers.

Although the major OSM providers offer publicly available APIs to access their data, challenges still exist to collect data systematic and efficient. For instance, data from a post on Facebook can require a high number of requests to get full coverage of all interactions. In terms of data collection it is important to know what you lose from various sampling techniques,
so one can adopt to that. This is one of the major contributions in this thesis; A systematic approach for sampling and crawling data from OSM.

1.3 Thesis Outline and Structure

Chapter 2 presents the Related Work, the Aim & Scope together with the Research Questions and the Methodology. In Chapter 3, the contributions are presented and the results are discussed and concluded in Section 3.2 and 3.3 respectively. Finally, the proposed Future Work is presented in Section 3.4 and the publications are then presented in Chapters 4–8.
This chapter presents the Related Work for the publications in this thesis. Followed by Aim & Scope, Research Questions, Research Methodology, and a short section regarding Legal and Privacy concerns.

2.1 Related Work

Analysis of user interactions on OSM has been a topic for several years. Garton et al. [33] identified the connection of people via computer networks as social networks in 1997. The area of various types of OSM are comprehensively described in [34]. Interesting studies include the studies by Grabowicz et al. [35] where the authors apply and evaluate social theories on OSM. Also the studies by Ferrara et al. [36] are interesting as it maps topology models on various social networks.

Many studies exist that either directly or indirectly cover the challenge of crawling various OSM. The studies conducted by Mislove et al. [37] are, the largest OSM crawling study available. From four popular OSM; Flickr, Youtube, LiveJournal and Orkut, 11.3 M users and 328 M links were collected. Their analysis confirms known properties of OSM, such as a power-law degree distribution, a densely connected core, strongly correlated in-degree and out-degree graphs and short average path length. Moreover, indirect studies of OSM crawling are presented in the studies by Wilson et al. [7] and Crnovrsanin et al. [8], where the authors transverse user profiles from Facebook. They collected roughly 70% of user profiles from various regional networks at high speed (averaging 10 MB/s) with quite limited resources. However, this study was conducted in spring 2008 and since then Facebook redesigned their site and it is no longer possible to crawl user profiles. More recently, a study by Buccafurri et al. [38]
discussed different methods to transverse the social network in a crawling perspective. Still, the restriction on crawling users profiles is not an issue in this work, since we gather data from public groups only. As such, our work has substantial data to crawl and our challenge differs from Buccafurri et al. [38].

There are several studies on social media and social networks where most of the data is from Twitter. This data is, however, typically collected using Twitter’s free garden hose API with a risk of being unbalanced and an unrepresentative sample of the complete data. Studies that address quality of social media data include [37, 39], where the former addresses how social media data from online recommendation systems can be evaluated. Sampling studies of social networks are quite common, including [40, 41] that uses the original graph sampling study by Leskovec et al. [42] as a baseline. Wang et al. presents an interesting study [43] on how to efficiently sample a social network with a limited budget. The study uses metrics of the graph to make informed decisions on how to transverse it. On the topic of graph and social media crawling Zafarani et al. [5] presents ways to evaluate and understand the data generated in social media.

According to our literature review there is a lack of studies that address the challenge of collecting data from Facebook (and other social media sites) after Facebook started protecting user profiles. Most studies are simply using online data repositories and does not address the issue of how to collect data directly from the social media sites.

Ever since the start of OSM the issues with users privacy have been considered [44, 45]. However, this is limited to means of the users’ privacy of the content posted and privacy settings within the OSM. Another problem is that a large extent of OSM users does not reflect upon how their interaction within OSM affect their privacy [10, 11], which could be a threat to their privacy [46]. As a natural consequence these users do not bother to investigate the content of the OSM policy documents.

Studies to classify data include Linguistic Inquiry and Word Count (LIWC) [47], which is a transparent text analysis program that counts words in psychologically meaningful categories. With LIWC it is possible to show attentional focus, emotionality, social relationships, thinking styles, and individual differences from just a small sample of text. Diversity introduced by Bhattacharyya et al. [17] can also be used to classify data.
The diversity factor is based on the relationship distance between two users. Interesting studies also include the study to classify and analyze network typologies by Michalski et al. [48]. Which also gives prediction measures to model evolution patterns of a social network.

Online social networks are a popular research area in the domain of contemporary network science [18]. The main focus in social network research is on link prediction [49] and social connection prediction [50]. Different teams around the world also work on: (i) personality prediction for micro blog users [51], (ii) churn prediction and its influence on the network [52, 53], (iii) community evolution prediction [54, 55], (iv) using social media to predict real-world outcomes [56], (v) predicting friendship intensity [57, 58], (vi) affiliation recommendations [59, 60], and (vii) sentiment analysis and opinion mining [61].

Since the emergence of Network Science [18], one of the most interesting research questions was: How the influence and information spread through the network of social interactions and how to maximize it [21]? There are many approaches to maximize the final coverage of the spreading and one of them is selecting proper set of initial seeds which will initialize the process. This set should consist of nodes with the highest combined potential to reach as big portion (in terms of no. of members) of network as possible. Those node are often called Influential users and play an important role in information propagation on online social networks as they have the highest impact on other users in the network.

Research into detecting influential users on OSM indicates that, while a large amount of followers seem to be present among influential users, predictions of which particular user will be influential is unreliable [62]. Depending on the social network, how to define influence differs, e.g., influence on Twitter might be defined by retweets or mentions, while, on Digg, votes generated are used to measure influence [63–65]. While some initial research has been done using clustering algorithms to identify top users, based on influence features, e.g., likes and replies, evaluation is lacking [66]. Similarly, linear regression has been used to identify influential (categorical) users based on influence features [65].

Private information that is withheld can express an individual’s aim uphold their privacy. Privacy is a way to limit the dissemination of an individual’s data and thereby express them selectively. The boundaries of
users’ privacy vary with the individual’s background and culture. Often
the privacy is a way to protect private information, sensitive to the user.
There are a few different privacy threats including identity theft [12],
surveillance [14], and online victimization which is further explained in [13].

2.2 Aim & Scope

This thesis aims to investigate how to efficiently collect user content and
interactions from online social media sites, and how to use the collected
data. Currently, methods to access a complete, as all the interactions
Corresponding to a specific post, dataset of interactions in OSM is lacking.
Interactions and produced data need to be collected in a structured and
efficient way. The nature of social media interactions follows a constantly
growing pattern that requires selection mechanisms to find interesting data.

2.3 Research Questions

The main questions we explore in this thesis is: How can user generated
content and interactions be efficiently collected from online social media
sites and for what purposes is the data valuable? While investigating
this question other challenges have risen. First, users’ privacy must be
considered. Second, if available resources are not sufficient for full retrieval
it is of importance to perform prioritization, i.e., only crawl data that are
of use to the current application. The main research question has been
approached using the following five sub-research questions covered in this
thesis:

RQ 1/ How can data from Facebook be collected with regards to depth, i.e.,
covering all interactions in a given domain, e.g. page?

It is of interest to collect data from OSM, for analysis purpose. Most
OSM sites of today have an API providing the ability to build tools to
access information from the site. However, these APIs often provides
just a sparse interface to the data and requires additional effort to
connect the data and make it useful. We are interested in how a tool
extracting data from, e.g., Facebook’s public pages must be designed
to access data with aspect of covering all interactions.
2.3. Research Questions

RQ II / *How can sampling be used to improve the data-collection process with regards to maximizing interaction coverage with limited resources?*

In OSM like Facebook, Twitter, and LinkedIn new data is created all the time. All this information is probably not equally useful and by crawling a selection of the data we can maintain the essence of interesting interactions.

RQ III / *How can user content and interactions on the collected data from OSM be valuable?*

There exists a challenge in crawling and collection of interactions from OSM. But once that information have been collected there must exist valuable use of the data. What type of applications can the crawled interactions be used for.

RQ IV / *How can influential individuals be identified using data mining in OSM and can the identified users be used for seed selection in information cascade in multilayer-networks?*

It is of interest to find users and items that are influential. Influential in this domain means that the items have the ability to influence others, this could be in means of creating an opinion, or just engaging for further discussions and user participation.

In addition, it is also of interest to evaluate how good identified users are as seeds in an information cascade setting of a multi-layer network.

RQ V / *Which privacy threats exist in OSM and what measures can users take to protect their privacy?*

With the use of OSM comes the potential threat of user privacy as addressed in Section 2.1. Users of OSM are often publishing information concerning themselves or people in close relation to the users. Users share various types of information of different level of sensitivity; ranging from just sharing a general link or funny picture to information such as checking-in at places. It is of importance to identify potential threats and find ways to protect the privacy of the user by making it possible to “lock down” the information so only the intended recipient or recipients can access the information.
2. Approach

2.4 Methodology

The tools developed to address the problems in this thesis regarding data collection and organization are implemented and evaluated by prototyping. The developed crawler is built to be resilient to failures and adaptable to external issues. The developed crawler and the tools supporting it are acting as a foundation for further studies for the research group at Blekinge Institute of Technology and University of California Davis with the objective is to share the tools and data with other researchers.

2.4.1 Strategies of Inquiry

The studies in this thesis are conducted in both quantitative and qualitative form. Quantitative research is conducted with a focused description and with a conclusive research [67]. In quantitative research, only measurable data is observed. In contrast, qualitative research have a broad description and with exploratory results [67]. Qualitative research focuses mainly on verbal data rather than measurements. Gathered data is analyzed in an interpretative manner, impressionistic, subjective or even diagnostic way.

Further, as this work started with a broad question related to data collection, it could be argued that the work presented in this thesis is in the form of exploratory research. Applied research that require flexibility when approaching the problem is often referred to as exploratory research [68]. This is further supported by the fact that there is sparse prior research in the problem domain. Thus, making an exploratory approach feasible.

Case studies have been used in the studies presented in Chapters 6 and 7. A case study is a type of observational research where observations are made of a phenomenon without interfering [69]. The observations from a case study are conducted as an in-depth study of a particular situation. One problem with case studies is that it is not possible to fully answer a question, as it is not possible to know when all subjects are evaluated. Instead, a case study will give indications and allow further elaborations. On the other hand, one of the advantages with case studies is that researchers are allowed to take new directions based on the study. In addition, experiments are used in the studies presented in Chapters 8, 9, 10, 11 and 12.
2.4.2 Evaluation & Validity Threats

The results are evaluated using the statistical methods described in the section below. Unfortunately does neither ground truth nor labeled datasets exist for the presented publications. Therefore are the results evaluated and validated against results from other studies and state of the art. Hence, we are evaluating against what can be seen as consensus in data.

With exploratory research there is always a validity concern of the drawn conclusions, as the problem definition is allowed to change during the study [68]. Actions have been taken by both manual and automatic verification of the results, in order to avoid this validity threat. E.g., the crawled data has been evaluated both against available data on Facebook’s web-based front-end and against the data accessible via the API.

Further, as the studies conducted in this thesis are based on a self-developed crawling method, that may pose a validity threat as the results reflects the data collected by our own method. Actions have been taken to minimize this issue. For instance, the study in Chapter 12 is made on a randomly sampled dataset to minimize bias results. In addition, the gathered data have manually been verified to be accurate and complete.

There is also threat to generalizability, often referred as external validity [69]. The presented publications investigates behaviors in public groups on Facebook. As the presented results have not been validated on other data (except for the publication presented in Chapter 11), can we only assume that the results are generalizability for other data.

2.4.3 Statistical methods

For the quantitative parts of the research, statistical methods have been used, including statistical tests of similarity and correlation. The results and conclusions are presented and evaluated based on significance with two-tailed confidence interval. The datasets have been selected using random sampling of non-synthetic data.

To investigate whether any statistical significant difference exists between different datasets, the Friedman test is used [70, 71]. The Friedman test is a non-parametric test that evaluates different treatments over multiple datasets. A non-parametric test is chosen over a parametric as normality cannot be assumed over the different datasets. As the test only detects
whether a statistical significant difference exists, and not where the difference exists, a post-hoc test is necessary to determine where the difference is located. The Nemenyi test is used as a post-hoc test [70, 71] in the included publications. To estimate parameters influences, for instance in Chapter 12, is a ordinary least square regression test [72] used. In addition, is Cohen’s d [73] used to quantifying the difference between multiple samples. Moreover, in Chapter 7 we use Cox proportional hazard model [74] in survival analysis to explore the relationship between user lifetime and several explanatory variables on the influence of feedback comments. In Chapter 6, we use Jaccard similarity coefficient [75] to match subgroups. All reporting of results includes standard measurements such as the test statistic, p-value, mean/median and standard deviation.

2.5 Legal and Privacy concerns

As the data used in this thesis is based on users’ interactions, there is a concern regarding to which extent this data is ethical to use. As the data is collected from public domain (open groups) and all data is anonymized before analyses are there no legal and ethical aspects of the data.
The work presented in this thesis addresses means of getting a complete dataset of user interactions from Facebook and analyzing the collected data. During the last four years our crawler have been crawling data enabling research with a comprehensive dataset. Currently data produced by 700 million Facebook users have been collected. Covering 280 million posts with 5 billion comments, and 35 billion reactions. The analyses addresses both various descriptive statistics for the data as well as detection of influential users in networks created from interactions in OSM.

### 3.1 Contributions

This thesis presents the following five contributions. First, it address the challenge of collecting data from OSM a study of designing a crawler capable of covering all interactions in a given page is presented in this thesis. The work presented in Chapter 8 acts as a detailed framework for the novel crawling selection process presented in Chapter 12. The crawler is not just novel in the way it crawls posts to the full extent of all interactions, it is also efficient as it is built as a distributed system. This distributed system, with one main server and multiple active clients responsible for the interaction with Facebook, enables high crawling rate and support for additional clients whenever the system requires more capacity. This relates and addresses RQ I, and RQ II, in which we present how an efficient crawler can be implemented and evaluated. The presented findings on prioritization of posts in the crawling context shows that it is possible to reduce the crawling time by 48.5%, while still covering 99.5% of all interactions.

Second, in Chapter 4, a study of potential user privacy threats, within OSM is presented. The study discusses six major threats to the user’s privacy; OSM information leakage, friend-in-the-middle, trojan application,
3. Results

Figure 3.1: Interactions around the contents shared on several Facebook public pages in a period of three weeks. The depicted users have interacted with other users on at least four communities.

public information harvesting, social bot and friend-in-the-middle trojan application. We also conduct a proof-of-concept showing how public information harvesting can be used to create interaction profiles and how to profile users. This study is not only of description and demonstration purpose, we also show how users can protect themselves to the presented threats. This chapter addresses RQ V.

Third, we present a Social Interaction Network (SIN), a way of representing social interactions in OSM. With SIN it is possible to follow users activity among different groups and see how opinion moves. In addition, SIN also enables studies of social interactions and visualizations. Figure 3.1 illustrates such a visualization showing how interactions around posts and comments of several public pages on Facebook are related. In Figure 3.1, the relationship between various media pages and the first three weeks of
the occupy movement\textsuperscript{1} is shown. For illustrative purpose, users interacting on less than four different communities have been removed. This work is fully described in Chapter 5, which answers RQ III by showing the use of gathered social interactions. In Chapter 6 we present the use of user interactions for opinion classification and grouping. This work is conducted by looking at the corresponding like-graph of comments related to a post. Also, in Chapter 7 we propose a dynamic user-like graph model for recognizing user deliberation and bias automatically in online newsgroups. We evaluate our identification results with linguistic features and implement this model in our SINCERE system (described below) as a real-time service. The Chapters 5, 6 and 7 contributes and answers RQ III, as we both visualize social interactions and show means of using them.

In addition, a framework to make the crawled data available and search-able in the form of a webpage has been developed and is called Social Interactive Networking and Conversation Entropy Ranking Engine (SINCERE). Figure 3.2 shows a demonstration of the social search web page SINCERE; where the user is able to search text from the crawled posts. One of the goals of SINCERE is to diversify information and tackle \textit{The Filter Bubble} \textsuperscript{16}, allowing the user to manually control the search ranking. Currently SINCERE supports ranking by content, number of likes, number of shares and number of comments made on the post. It also supports two types of entropy ranking methods: user entropy and post entropy. Entropy in this context reflects on the level of information novelty and diversity. The comments corresponding to the search result are clustered in two columns based on the users’ opinions classification and grouping, presented in Chapter 6. In Figure 3.2 the comments from users identified as negative are to the left and from positive users are to the right.

Fourth, we address RQ IV by investigating methods to identify influential users using data mining and in particular association rule learning in Chapter 9 & 10. It is shown that the proposed method of using association rule learning for identifying influential users have an accuracy of 91\% ($sd = 12\%$).

Finally, in Chapter 11 we address RQ IV and show how the findings in Chapters 9 & 10, using machine learning and in particular association rule

\textsuperscript{1} The occupy movement is a protest against social and economic inequality.
3. Results

SINCERE
Social Interactive Networking and Conversation Entropy Ranking Engine

Heard some of the LAPD dropped off 3 crates of supplies: hygienic products, snacks, sunscreen, etc. for Occupiers! -M.K.

Figure 3.2: Snapshot of the webpage SINCERE, showing the search result for “Heard some of the LAPD”. The first post and its corresponding comments are visible. The comments shown are clustered in two opinion groups, where the left group are from negative users and the right group are from positive users.

Learning to identify influential users also can be used to identify information spreaders in multi-layer complex networks.

3.2 Discussion

Currently no methods exists to access the data corresponding to the complete interactions around posts\(^2\) from OSM sites. There are even indicators that the OSM providers themselves does not have easy access to this data and even if the data exists it is hard to extract it. For instance, Facebook have powerful tools to select information and advertisements for

\(^2\)complete interactions refers to all actions users have taken on a specific posts, including: reactions, comments, shares and reactions on comments
its users. However, methods for extraction of the complete interactions are not available, through the API.

The work presented in this thesis is limited to cover interactions around open pages on Facebook. Currently there is a gap as it is not possible to get interactions from a particular user in a specific time-span. This work bridges this gap and enables researchers access to social interaction data from publicly accessible pages.

As Facebook registers users’ actions; we are able to collect the users’ actions in public pages. This collected data is organized within the SINCERE framework and made publicly available at http://sincere.se. The way the data is structured and organized enables the research community to study patterns and behaviors of users. Do note that due to concern of individuals’ privacy no studies of single user behavior are conducted as described in Section 2.5. Our data is available, as shown in our web page SINCERE illustrated in Figure 3.2 with means of introducing diversity in the presented results, i.e., to mitigate The Filter Bubble [16]. It is also possible to create Social Interaction Networks as illustrated in Figure 3.1.

The findings of reduced crawling time by prioritization of highly interesting posts should also be investigated further. Prioritization has the advantage of putting stronger emphasis on information with higher interestingness, while disregarding the less interesting items. The likely disadvantage is that some of the disregarded items assessed as not interesting may in fact carry information of high interestingness.

Users within online social networks create a large amount of data in the form of interactions (e.g. comments and reactions). Not enough attention has been put on the analysis of how users influence each other and how to predict the behavior of users within Facebook groups. We have implemented and examined how users influence each other, by using association rule learning. Based on the results and analysis, we are able to determine to what extent users influence other users to participate and interact in new groups. The results show that influential users can be identified using association rule learning. That is, users on the left-hand-side, in a rule with high confidence and high lift, are influencing users on the right-hand-side to participate in the conversation. These results have been verified and compared with the traditional network analysis methods, PageRank Centrality and Degree Centrality. Showing that at best ~30% of the users
ranked using association rule learning overlap with the users ranked using traditional methods.

Information cascade is an interesting field to online social media. We have evaluated different seed selection methods to model how information spreads on public pages on Facebook. The public pages from Facebook were crawled completely (covering all posts, comments and reactions) using the crawler described in Chapter 8. The results show that association rule learning can be used for seed selection in an information cascade setting. Association rule learning perform equally well in selecting seeds as state-of-the-art methods (degree centrality and k-shell) in terms of information coverage. Surprisingly does VoteRank not perform as good as expected in terms of information coverage on our dataset.

3.3 Conclusion

This thesis investigates how data from OSM can be gathered and utilized. Different approaches to gather interactions from Facebook are investigated. First, a novel method of crawling built as a distributed system with high error tolerance is presented. Second, a study of how to improve the efficiency of data collecting process, with respect of gathering as many interactions as possible, with the available set of resources, is conducted. The crawled data is made available through a novel framework; both in the form of theoretical design guidelines and the openly available APIs of SINCERE. Enabling future research in the field of computer science but also in other research areas, e.g. network- and social sciences, where the vast number of interactions between different users and communities could be further studied.

The thesis further addresses different privacy threats that users of OSM are exposed to. One of these threats is previously undocumented, user profiling based on the activity in publicly open groups. It is proven that with limited resources it is possible to profile users within an OSM through the activity in open groups and then build a social interaction graph of their interactions. Any user within the OSM is vulnerable to this threat, if they interact with public pages, independent on their privacy settings. We suggest a number of different protection mechanisms against the threats identified.
Finally, the last chapters in this thesis addresses the value of data from social media. We show that interactions on social media can be modeled in to interaction networks. An interaction network is created from the bipartite network of users’ interactions on posts as a projection into a user to user network. This interaction network is shown to enable network based models to model and understand users’ behaviors on social media. The thesis address both methods for finding appropriate seeds for information cascade, and identification of influential users. Also, association rule learning has been evaluated as a method for identifying influential seeds. We show that it is possible to use association rule learning for identifying seeds with no significant difference than other novel methods.

3.4 Future Work

The developed crawler is currently capable of collecting data on Facebook. It would be interesting to enhance the crawler to cover other OSM as well. In addition, the current stage of the crawler requires manual input of pages. Therefore, work to extend the crawler to automatically discover pages to crawl would be interesting to pursue.

Finally, studies to further investigate the shared interactions in OSM are an interesting field with nearly unlimited opportunities for future studies. Interesting studies include, but are not limited to: a study of time distribution of interactions per communities and gender. A study to determine tendencies and trends of community route path, i.e., how users tend to move between different communities. Such a study could also aim to identify influential users, including the trend of user intensity and the top users with most activity. Based of the comprehensive dataset at hand, studies to validate social science and humanity research based on social interactions are also interesting to further investigate.
Bibliography


Privacy Threats Related to User Profiling in Online Social Networks

Fredrik Erlandsson, Martin Boldt, Henric Johnson

Abstract

The popularity of Online Social Networks (OSNs) has increased the visibility of users’ profiles and interactions performed between users. In this paper we structure different privacy threats related to OSNs and describe six different types of privacy threats. One of these threats, named public information harvesting, is previously not documented so we therefore present it in further detail by also presenting the results from a proof-of-concept implementation of that threat. The basis of the attack is gathering of user interactions from various open groups on Facebook which then is transformed into a social interaction graph. Since the data gathered from the OSN originates from open groups it could be executed by any third-party connected to the Internet independently of the users’ privacy settings. In addition to presenting the different privacy threats we also propose a range of different protection techniques.

4.1 Introduction

In the beginning of 2012 Facebook had about 800 million users and the company was valued to over 100 billion dollars which to large extent originate from advertisement and user profiling possibilities based on user interaction. Besides Facebook there are a number of different Online Social Networks (OSNs) that has reached a considerable user-base, e.g. Google+, Twitter and LinkedIn.

It is therefore important to address the privacy implications of how the published information within OSNs is handled. Information that is published by users within a limited group, or perhaps shared with a single
4. Privacy Threats Related to User Profiling in Online Social Networks

user is often of a nature that can cause significant inconvenience, or even harm to concerned users. As OSNs grow in size the methods and knowledge among its users about how to configure privacy settings is crucial. In this paper we list different privacy threats within OSNs together with potential protection mechanisms. In addition to this we also add a new privacy threat that originates from scraping publicly available information which is published in open groups within the OSN.

OSNs like Facebook, Google+ and Twitter all provide open interfaces (i.e. APIs) for third-party applications to interact with the OSN by accessing and publishing data. This is very convenient for the user as it opens up possibilities for value increasing applications to interact directly with the social network. Consequently, there are more than 500,000 third-party applications, such as online games that interact and coexist with Facebook [1].

What people does not reflect upon is the fact that most of these applications have the abilities to interact with the OSN on behalf of the user, which also includes the possibility to gather information that the user posted as private correspondence. Add to this that users on OSNs share information that could be harmful for the user itself, or even the user's friends. As an effect Trojan applications that use deceptive and covert behavior can gather such sensitive information from users. However, Trojan applications can also retrieve information among a user's friends including their posts, which threaten the privacy of the OSN users.

4.2 Privacy Threats

In this section we will present six different types of privacy threats illustrated in Fig. 4.1. All of these threats result in user information leakage from the OSN to third parties. These privacy threats exist because social information about OSN users has a value, and can be refined into revenues within the context of targeted advertisements etc.

4.2.1 OSN Information Leakage

The first type of privacy threat, illustrated in case (a) in Fig. 4.1, is based on that the owner of a OSN, e.g. Facebook or Google, continuously gather detailed information regarding users activities within the OSN. This is
probably the most obvious privacy threat and as such it is well known within research community and it is also the threat that OSN users first come to reflect upon [2, 3]. We therefore expect OSN users to understand that information they share within the OSN, e.g. user profile content, messages, and photos, can be mined, refined and sold by the owner of the OSN. Exactly how the OSN owner is allowed to use and benefit from this information is regulated within policy documents, e.g. the statement of rights and responsibilities [4] and the data use or privacy policy [5] for Facebook. A problem is that a large extent of OSN users don’t reflect upon how their interaction within OSNs affect their privacy, which could be a threat to their privacy [6]. As a natural consequence these users do not bother to investigate the content of the OSN policy documents.

There is also a risk that the OSN infrastructure get compromised, giving third parties unauthorized access to sensitive information [7].

4.2.2 Friend-in-the-Middle Threat

Case (b) in Fig. 4.1 shows a type of privacy threat where user information is leaked through a trusted friend within the OSN. Because of this threat the OSN infrastructure often provide users the possibility limit their posts and information spread to smaller group, which (if used correctly) could be used as one method for avoiding public scrutinization. Unfortunately a chain is not stronger than its weakest link, which goes for friendships within OSNs as well. A large portion of OSN users act irresponsible by more or less allowing anybody to establish a friendship, which not only affect the user but potentially also that particular user’s friends.

One must also consider the current state of social gaming, where users require a certain number of friends in order to achieve certain tasks (level up) [8]. This tend to cloud users’ judgements regarding whom they are accepting as friends, as they instead focus on the primary task ahead, i.e. leveling up.

4.2.3 Trojan Application

The third type of privacy threat is associated with Trojan applications leaking information about its OSN users to third parties, see Fig. 4.1 case (c) [9]. The user is deceived to install a Trojan application which claims
4. Privacy Threats Related to User Profiling in Online Social Networks

![Diagram of privacy threats in OSNs](image)

Figure 4.1: Six different privacy threats within OSNs, where (a) is leakage from the OSN infrastructure to a third-party, (b) a friend-in-the-middle threat, (c) a Trojan application, (d) is public information harvesting, (e) is a socialbot, and finally (f) that represent a friend-in-the-middle Trojan application.

To provide some desired functionality, but also hides unwanted and shady behavior, and as a result leak valuable information.

4.2.4 Public Information Harvesting

Case (d) in Fig. 4.1 illustrates a new type of threat that we present in this paper, and as such it is previously unknown within both academia and among OSN users. The basis of the threat is that third parties collect user information published in open groups within OSNs like Facebook. Such open groups exist in the boundary between the OSN and the publicly available Internet. Since the information is gathered from open OSN groups there is no need for using covert or deceiving methods when collecting the information. It is simply a matter of scraping the information available on these web pages, which can be done by anyone connected to the Internet. Using the harvested information it is possible for third parties such as profit-driven companies or national security agencies to create social interaction graphs, which details how users interact among a certain topic, e.g. the Occupy Wall street movement. This privacy threat is described further in Section 4.3.
4.2.5 Socialbot

Recently automated software programs, called *socialbots*, have been seen influencing OSN users [10]. These socialbots are designed to control OSN accounts, by autonomously performing basic tasks such as posting messages and sending friend requests. Socialbots are not applications within the OSN itself, but rather software programs that impersonate the human beings behind user accounts by imitating human behavior towards the OSN, and as such the socialbots fool both the OSN infrastructure itself and the users populating it. Socialbots with these features have been seen infiltrating private and trusted areas shared by Friend relationships in Facebook, and as a consequence harvesting sensitive data from the concerned user accounts.

The threat from socialbots increase since many users are irresponsible when accepting new friend requests from unknown users. In a practical demonstration a socialbot were accepted as friend by OSN users at a rate of 19.3% out of 4493 requested users during the initialization phase and by 59.1% during the socialbot’s propagation phase [11]. Given this high acceptance rate regarding unknown users’ friend requests it is questionable what the effect of privacy settings that limit information access to friends, or friends-of-friends within a OSN really have in practice. If a user’s friend is routinely accepting friend-requests from unknown sources, this friend is a privacy threat, even though this might be unintentional, to both himself and his friends. With respect to our privacy we have therefore come to a situation where we no longer can fully trust the integrity of our friends within OSNs.

4.2.6 Friend-in-the-Middle Trojan Application

This type of threat is indirectly affecting a user when one of the user’s friends add a deceptive Trojan application. The effects on the user and the user’s friends privacy is similar to Trojan application threat described previously. As such, a user’s privacy is dependent not only on his/her own ability and judgement, but also on his/her friends competences, or even weakest friend in this regard.
4. Privacy Threats Related to User Profiling in Online Social Networks

Figure 4.2: (a) and (b) show the interactions done through comments and likes on posts shared on various “Occupy WS” groups. (a) shows interactions before a pepper spray incident at UC Davis, while (b) shows interactions a few days after the incident. Different colors represent different groups; Occupy UC Davis - magenta, Occupy Wallstreet - lilac, Occupy Los Angeles - light blue and Occupy Sacramento - light green.

4.3 Proof-of-Concept

The threat we describe as public information harvesting is based on that users within Facebook can interact in open groups that are publicly available from the Internet. User interaction within these groups is in the form of “Likes”\(^1\) on the group itself, comments within the group, or “Likes” on other users comments. By systematically gathering this public information it is possible to create interaction profiles identifying and profiling users based on the interactions made, i.e., through social interaction graphs as shown in Fig. 4.2.

4.3.1 Gathering of Information from Open Groups

Facebook provide different methods for third-party application interaction, for instance using the Graph API [12]. The use of this API is straightfor-

\(^1\)“Like” is a term found in Facebook where an user can show that they agree or in other way would like to show that they share the same thought as the message, this is called +1 in Google+.
ward, in a few hours we built an application acting as a data extraction tool that gathered information as an authenticated user on Facebook. Then we created a dummy-user without any interactions or affiliates to begin with. Next our newly created dummy-user accepted our application with just basic permissions. It was then through this dummy-user’s application we gathered data from various open groups on Facebook. However, it is important to stress that the content of open groups are freely available on the Internet so there is no requirement of using a dummy-user to extract this information, we only used it due to convenience reasons.

The information gathered have traditionally been seen by research community as simple post and user information. We have however seen that the information gathered follows such a structured form that different users’ interactions can be combined and form a social interaction graph. Any third-party can gather this user information independent on the user’s privacy settings without their knowledge.

4.3.2 Creating a Social Interaction Graph

From the information gathered in the previous step we created a social interaction graph shown (Fig. 4.2). The figure shows the interactions between different networks before (Fig. 4.2 (a)) and after (Fig. 4.2 (b)) the Pepper-spray incident that happened in Davis, CA. This pepper-spray incident resulted in not only more intense interactions, but also that users involved in their representative community started to interact with other “Occupy” groups.

When looking at the created social interaction graph we can conclude that even if a user have strict privacy settings the user’s actions are hard to hide. We were able to gather not just the name of the users, but also the profile ID making it possible to find out more information about the human behind the user account. The users in today’s OSNs must understand that no matter how strict they are trying to protect their user profile with policies, they are still at risk of being profiled based on their behaviors in various groups.

4.3.3 Privacy Implications

Since public information harvesting can be carried out by basically anyone it definitely pose a threat to user privacy. One such example is countries
where the regime is interested in targeting and monitoring citizens engaged in various issues that are uncomfortable for the regime. For users living in countries that respect human rights the threat might come from corporations and advertisers to larger extent.

4.4 Protection Mechanisms

In this section we suggest different protection mechanisms against the threats described in section 4.2. This list of protection mechanisms is with no means complete. Using encryption for instance it would be possible to address several of these threats if the OSNs could act and facilitate a public-key infrastructure (PKI), but due to the space-limitation we exclude that in this paper.

4.4.1 Information Leakage from the OSN

Since it is impossible to reach absolute security in any system it is important to inform the OSN users in an adequate manner regarding how their information is handled. Here public discussions to raise user awareness is an important component. It should also be possible to benefit from existing techniques to increase transparency of the OSN policies towards the users, e.g. “Privacy Simplified” [13] that help summarize privacy policies using standardized icons. In addition to improving user awareness it is of paramount importance that the OSN infrastructure is properly secured, and that a continuos security process is established.

4.4.2 Trojan Applications

To improve the protection against Trojan applications, case (c) and (f), we suggest the use of an application certification and reputation program, which just recently has been announced by Facebook under the name “App Center” [14]. We suggest that a more privacy-driven application certification program is added to this initiative, where not only the overall application quality is evaluated, but also the privacy-implications of the data gathered. Combined with a privacy policy of what data the application will retrieve and how the application will handle this information would make a valuable addition. It is also preferable that interested users should
have the possibility to see an audit trail of the interactions the application has carried out on behalf of the user [15].

4.4.3 Questionable Friends

The most important issue to focus on is the lack of user awareness about the problem shown in case (b) and (e), which could be addressed through end-user education. Instructing the users about socialbots, Trojan applications, and the implications of the “Friends-of-Friends” privacy setting. Users should also be instructed to keep their friend-list up to date as far as possible. Also by using various algorithms like the one presented by Fire et al. in [16] the number of questionable friends can be limited.

4.4.4 Social Interaction Profiling

To protect against social interaction profiling, case (d) in Fig. 4.1, we suggest the use of pseudonyms or virtual profiles [17]. However, by hiding the real identity of the end-user, for instance using anonymity techniques, will also remove one fundamental value of the OSN, i.e., that the OSN transcends the real world since each user account (more or less) corresponds to a human being. By using pseudonyms it is possible for a user to interact under separate pseudonyms in different open groups, which renders it impossible to make connections between different groups at least.

4.5 Conclusion

In this paper we present different privacy threats in OSNs. One of these threats is previously undocumented and we therefore describe this threat in more detail together with the results from our own proof-of-concept implementation, which includes the resulting social interaction graph that could be used for user profiling. The proof-of-concept shows that with limited resources it is possible to profile users within an OSN through open groups and then build a social interaction graph of their interactions. Any user within the OSN is vulnerable to this threat, independent on their privacy settings. Finally we suggest a number of different protection mechanisms against the threats identified.
4. Privacy Threats Related to User Profiling in Online Social Networks

4.6 References


4.6. References


SIN: A Platform to Make Interactions in Social Networks Accessible

Roozbeh Nia, Fredrik Erlandsson, Prantik Bhattacharyya, Mohammad Rezaur Rahman, Henric Johnson, S. Felix Wu

Abstract

Online Social Networks (OSNs) are popular platforms for interaction, communication and collaboration between friends. In this paper we develop and present a new platform to make interactions in OSNs accessible. Most of today’s social networks, including Facebook, Twitter, and Google+ provide support for third party applications to use their social network graph and content. Such applications are strongly dependent on the set of software tools and libraries provided by the OSNs for their own development and growth. For example, third party companies like CNN provide recommendation materials based on user interactions and user’s relationship graph. One of the limitations with this graph (or APIs) is the segregation from the shared content. We believe, and present in this paper, that the content shared and the actions taken on the content, creates a Social Interaction Network (SIN). As such, we extend Facebook’s current API in order to allow applications to retrieve a weighted graph instead of Facebook’s unweighted graph. Finally, we evaluate the proposed platform based on completeness and speed of the crawled results from selected community pages. We also give a few example uses of our API on how it can be used by third party applications.

5.1 Introduction

Facebook has over 950 million users and still growing. There are over 2.7 billion likes and comments posted on Facebook on a daily basis as of February 1st 2012 [1]. The fundamental block of the Facebook platform is the social graph. A social graph can be defined as set of nodes and edges,
where each node represents a user and each edge represents a connection between two users. Moreover, along with the growth in social graph, Facebook has introduced technologies for users to share multiple levels of information. Users share personal information related to their name, contact details, photo, current location, hometown, interests, activities among other examples. Users also share non-personal information in the form of content from the traditional Web. ‘Share’, ‘Like’ and ‘Recommend’ buttons typically help users share this set of content. Interactions on Facebook pages also creates an additional set of content. Facebook allows users to interact with each other through many different means through the content shared on its platform. For instance, users can like, comment on, or re-share a content that is posted by another user. Users are not limited to interact with their immediate friends, but they can interact with anyone on Facebook through Facebook Pages and/or Groups.

Most online social networks including but not limited to Facebook, Twitter, and Google+ provide APIs for third party applications to request parts of the social graph. The traditional API provided by Facebook is able to capture the static information regarding the social graph as described above, but is limited in regards to providing the social interactions on its platform. For instance, many “occupying movement” pages have been created recently on Facebook, which brings up the question of how would one capture the social interactions of one page and combine or compare the results with the social interactions of another related page. For example, users of Occupying San Francisco seem to be interacting more around the idea of “Tea Party,” while the users of Occupying Chicago had more interactions on the issue of large corporations. Often times, we do not need to know who in particular has had interactions, but we are interested to know what the society as whole is interested in; therefore, we can anonymize the networks to preserve users privacy without loss of critical information.

This particular limitation arises in Facebook APIs due to the fact that the social graph is completely segregated from the content shared on these networks except for the ownership of the content. For example, an application can access the content shared by the logged in user or the immediate friends of the user if they have the required permissions, which gives a weak social relationship between the content and people. We believe that the content shared and the actions taken on the content, whether it is to like, comment, or re-share the content, creates a Social Interactions
5.1. Introduction

Figure 5.1: shows interactions around the contents shared on several Facebook public pages in the third week of occupying movement. The users shown have interacted with other users on at least four communities. (a) Shows news agencies such as ABC News and MSNBC on the left side of the graph and the occupying movement communities on the right side of the graph. (b) A closer look at the left portion of the graph shows that MSNBC has a much stronger tie to the occupying movements than ABC News. This could result in a higher influence from MSNBC on the occupying movements compared to ABC News.

Network (SIN). The graphs generated from SIN connects people through actions and thus interactions with other users instead of the traditional friendship connections. We believe that the social interaction networks can be very useful and may represent a closer social network to the real life human interactions. The SINs could be used to solve many of the existing problems in today’s world such as the Social Search Engine, Friend Finder, and/or Related Shopping Items.

In this paper, we ask the question on how we can design a Social Content based API to support the interactions between social network users and the contents shared. We introduce a new set of API calls in addition to the current Graph API supported by Facebook, which allows third party applications to create Social Interactions Networks based on a given context. Our API is comparable to Facebook’s Graph API, making it easy for further developers to easily adopt the new API. We also address the scalability issues of effectively capturing social interaction information from Facebook where the number of interactions are many and happen
Figure 5.2: shows for every two occupying movement pages how many users have interacted on both pages within the same time frame. The data above shows the interactions that had taken place during one week. For instance, the data shows that around 900 people interacted with other users both on “Occupy Los Angeles” page and “Occupy Chicago” page.

very quickly. We evaluate our API based on completeness of the results returned and the speed of our platform.

We also use our platform development efforts to analyze the influence of social interactions of a particular community has on other communities/pages on Facebook. Figure 5.1 shows the influence of news agencies on social interactions on some of the occupying movement communities on Facebook. Furthermore, figure 5.2 shows the influence of occupying movement pages on each other by showing for every two pages how many users have interacted on both pages within the same time frame.

The rest of the paper is organized as follows. In Section 5.2, we talk about the related work in the area of social interactions network. We then describe the idea of social interactions network in section 5.3. We give a few example applications for our API in section 5.4. Section 5.5 describes the details of our proposed API. In section 5.6 we discuss the security, privacy, and implementation challenges. We evaluate our API in sections 5.7. Finally, section 5.8 talks about our future plans.
5.2 Related Work

Researchers have begun to look at the real-world social interactions instead of the social networks of friendships or followers provided by OSNs. One of the original papers to study the emerging social network phenomena focused on the Club Nexus website of Stanford University [2]. Ever since, there’s been work done on CyWorld, MySpace, Orkut [3], YouTube, Flickr, LiveJournal, and Orkut [4]. Yet another study focused on profiling social network evolution on Flickr and Yahoo! 360 [5]. Finally, a recent measurement study analyzed the growth of Flickr social network using a three month crawl data [6]. These studies confirm that online social networks obey power-law scaling characteristics [7] and exhibit high clustering coefficients, firmly establishing them as small-world networks [8].

Recent studies analyzed the online communication patterns among the users in a large IM trace [9], and in an online social network [10]. The IM study also reported relatively higher value of average path length for the graph formed from user interactions. However, the IM interaction graph is more resilient to node removal than the interactions graph in Facebook, as Christo Wilson, et al indicated in their study [11], where they introduce the interaction graph as a more accurate representation of meaningful peer connectivity on social networks. They believe analysis of interaction graphs derived from their Facebook data reveal different characteristics than the corresponding social graph. They conclude with experiments to evaluate effects of interaction graphs on two well known social applications. The performance of RE [12] improves with the use of interaction graphs, as the streamlined link structure helps control spam proliferation. In the case of Sybilguard [13], the system becomes less able to effectively classify nodes once its assumptions about graph structure are violated. Researchers have shown that the social interaction networks represents a strong representation of active developers in OSS projects [14]. They further show that social interaction networks are very stable in presence of noise or lack of enough information and still have a very strong correlation with the active developers network [15].
5.3 Social Interactions Network

Next, we will discuss how we leverage the Facebook’s platform to design one architecture that provides the social interactions networks. Most Social Networks today, such as Facebook, Twitter, and Google+, provide APIs for third party applications to build applications on top of their platform. Looking at the data of many different Facebook pages on how the users interact with each other, we have found that the social network graph that arises from these interactions differs a lot in type and structure based on the type of interactions we are looking at. The social graph provided by Facebook currently does not provide the enormous amount of information we can gain from the social interactions networks formed around Facebook communities. We believe that the content shared and the context of items shared on Facebook groups and pages plays a huge role in the formation of these social networks. Although, it is possible to recreate these networks from Facebook’s current API, one has to make many different requests; since, Facebook only returns a limited portion of the data with each request, and do a lot of analysis and computations on the data retrieved in order to accomplish this task; therefore, not only recreating these social interactions networks from Facebook will require a lot of work, but a naive implementation may results in an incomplete results set due to instabilities, or a slow application due to delays in a sequential and non-parallel implementations.

In Social Interactions Network (SIN), which is an extension of FAITH [16], we provide a set of API calls in addition to Facebook Graph API calls to allow third party applications to retrieve the Social Networks formed around the contents shared on Facebook groups and pages efficiently and easy. From now on in this paper we will call these networks Social Interactions Networks. Our API uses the same ideology and interface as Facebook’s Graph API, which makes it very easy for third party applications to adopt our API.

We believe that each community (i.e. page, group, or a user’s profile) on Facebook gives a context around which people will interact with each other. Looking at the community structure of social interactions network we believe that the context plays a huge roll in how people interact. For example, the social interaction network formed on Jay Leno’s page is very different from the social interaction network formed in the Citi Bank page
on Facebook. Facebook gives the option to page admins to allow or disallow fans to post on the page’s wall. For instance, the Citi Bank page does not allow its fans to post content on their wall, so users can only like or comment what has already been posted by the page admins. Although, there is some interactions between users by liking comments that were posted by fans of the page, the average path length on the SIN formed around the contents shared on this page is one. Table 5.1 shows how the social interactions networks formed around the contents shared on different public pages on Facebook differ in number of members, the way users interact with each other, whether it is through likes or comments, the amount of interactions, and other network properties such as the overall average path length and clustering coefficient of the networks. Furthermore, the data shows us that the interactions on the same page can differ a lot in different time periods or around different events. Figure 5.3 shows the social interactions network around the contents shared on UC Davis’s Facebook page. Figure 5.3 (a) shows the interactions before the pepper spray incident [17] at UC Davis, while Figure 5.3 (b) shows the interactions immediately after the incident. Given the community id (i.e. the context) our API will retrieve the social interaction network that is formed around the contents shared in that context.

Table 5.1: Shows the different social interactions networks formed around the contents shared on different public pages on Facebook.

<table>
<thead>
<tr>
<th>Community</th>
<th>Posts</th>
<th>Comments</th>
<th>Likes on Posts</th>
<th>Likes on Comments</th>
<th>Fans</th>
<th>Avg Path Length</th>
<th>Clustering Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Against SOPA</td>
<td>5690</td>
<td>8564</td>
<td>3864</td>
<td>15529</td>
<td>117000</td>
<td>3.3</td>
<td>0.172</td>
</tr>
<tr>
<td>Chase Community Giving</td>
<td>196</td>
<td>27231</td>
<td>4410</td>
<td>40481</td>
<td>3300000</td>
<td>2.3</td>
<td>0.331</td>
</tr>
<tr>
<td>Citi Bank</td>
<td>204</td>
<td>2801</td>
<td>4257</td>
<td>1307</td>
<td>300000</td>
<td>1</td>
<td>0.362</td>
</tr>
<tr>
<td>Jay Leno</td>
<td>41153</td>
<td>42236</td>
<td>50789</td>
<td>43042</td>
<td>423000</td>
<td>2.4</td>
<td>0.271</td>
</tr>
<tr>
<td>Chase Slates</td>
<td>105</td>
<td>229</td>
<td>1705</td>
<td>127</td>
<td>58000</td>
<td>4.4</td>
<td>0.218</td>
</tr>
</tbody>
</table>

5.4 Applications

**Friend Suggestion:** One of the biggest challenges that most popular social networks face is their friend suggestion feature [11], called People you may know in Facebook.” Although, Facebook does a great job of showing the people that we might know who are not among our Facebook friends,
they do a very poor job of finding people that we do not know, but may be valuable friends. There are over 900 million users of Facebook, and based on the social interactions that form around the contents shared, Facebook or third party applications should be able to suggest people who have similar tastes, ideologies, and/or believes to each other to be friends. We believe that the social interactions network is able to identify these people based on how they have been interacting with each other through the content shared in a given context on Facebook. Our API provides data to create this functionality quite easy. Since, we know the context that users are interested in (i.e. the Community) and based on the corresponding social interaction networks used we can find people who share similar interactions on the community shared content.

**Better News Feed**: People spend hours and hours on Facebook every day. However, they are only bound to see the posts shared by their immediate friends and the pages they have liked. Through our API, it is possible to see what kinds of posts the user has been interacting with and find similar posts based on the SIN formed around it that the user has not interacted with. This will create a more dynamic newsfeed rather
than the current one where users see the same posts over and over again throughout the day. Again, we know the context that the user is interested in, and using the corresponding social interactions network we can identify which posts the user would be interested in but has not interacted with yet. Therefore, the user will only see posts that he/she has not seen before and the content is related to what he/she likes. The social interactions network constantly changes based on user’s behavior. Therefore, we can even show relative content to users current mood depending on how they have been interacting with the content shared recently. Figure 5.4 shows how SINs change over time.

**Social Search:** Social Search [18] is one of the hottest areas in the market and companies like Google, Facebook, and Microsoft are spending billions of dollars in the race of building the best social search experience. We believe that the SINs formed around the content shared on these page and groups give better results when combined with a search engine than the friendship networks currently used. While a group of users have very similar and close interactions around the content shared on Facebook, we can use this information when a person from this group queries something. We know the group’s interests and that will help us serve the user with better social search results. Since there is a cap on how many friends users can have on Facebook, the social search will be limited to the number of direct friends. In addition to the limited social network, there are no guarantees that users immediate friends will share the same taste, thought process, or needs. In our approach we link users with many interactions on related content to provide better search results. Based on the query we can identify the context and use the matching social interactions network to find related content.

### 5.5 SIN API

We have adopted the same interface as Facebook’s Graph API, which makes it easy for third party applications to use. We introduce the following API calls to enable third party applications to interact with the social networks formed around the content shared on Facebook.
5. SIN: A Platform to Make Interactions in Social Networks Accessible

Figure 5.4: Shows the interactions around 49 different Facebook public pages. The users have interacted with other users on at least two different communities. (a) Shows the interactions that have taken place during a one week period exactly a week after the occupation movements started. (b) Shows the social interactions that have taken place during a one week period three weeks after the occupation movements started.

Definitions

A **Community** on Facebook can be one of the following: Page, Group, or User. Usually, every community has an owner or an admin who keeps the community active. Each community usually defines a context around which people share content. Then the users interact with each other through the content that is shared on a given community by liking, leaving a comment, or re-sharing the content.

A **Post** is anything that is shared on Facebook. It could be a simple text message, a link to a third party web site, an image, or a video. There are many pages and groups with millions of members. It is amazing that posts in popular pages and groups get tens of thousands of likes and comments and hundreds or thousands of shares. The SINs that form around these posts are very large and have been neglected for the most part.

Our code is done in PHP. Developers will need to use our SDK instead of the one Facebook provides and once an application creates an instance of the Facebook class and assign it to the $facebook object as they would with
Figure 5.5: Shows directed and weighted graph of a network of social interactions formed around a single post on a public Facebook Page. Clusters in the graph are identified by colors. The network consists of 1097 nodes and 2028 edges. There were a total of 25 likes on the post itself, 888 comments on the post and 1252 likes on the comments.

a normal Facebook application using their Graph API, they can simply use our added api functionalities by calling the following methods:

\$facebook\rightarrow\text{api(‘/faith/{Post-ID}’, \$limit)} - This call returns the Social Network that is created by the interactions of users around a single post. We first retrieve general information about the post itself and then we iterate through the likes, comments and shares related to the post. We create the SIN around that single post and return the results to the user. We give different weights to different actions that have been taken on the post. Shares have the highest weight on the link from the person who shared the content to the person who posted the content originally. Comments have lower weight than shares but higher than likes on the link created from the person who left a comment to the person who posted the original content. Finally, likes have the least weight on the link from the person who liked the content to the person who originally posted the
5. SIN: A Platform to Make Interactions in Social Networks Accessible

Figure 5.6: shows the interactions done through comments on the same post. There were a total of 406 users interacting on this given post at the time the data was generated. The closer a node to the center of images means that the user has had more interactions on the post than the users that are further away from the center. (a) shows everyone who has left a comment on the post. (b) shows the people who have left more than three comments on the post. Applying this filter reduced the number of remaining users to 176.

$\text{facebook} \rightarrow \text{api}('/faith/{Post-ID}/comments', [\$limit])$ - The Social Graph returned by this call contains only links created by comments around the given post. Basically, a link in this graph simply means that a user has left a comment on the content. This Graph is a star shaped graph. The weight of the links depends on how many comments each user has left on the given post. The higher the number of comments the stronger the link from the user to the middle of the star, which represents the originator of the post. Figure 5.6 shows the interactions done through comments on the same post as above. Fig 5.6(a) shows everyone who has left a comment on the post. Fig 5.6(b) shows only users that have left more than three comments.

$\text{facebook} \rightarrow \text{api}('/faith/{Post-ID}/shares', [\$limit])$- The Social Graph returned by this call contains only links created by re-shares of the original content. Each link represents a re-share of the content between the person who has shared the content to the person who has originated the content. The Social Graph returned by this API is also star shaped.
Figure 5.7: shows a social interactions network of “likes” around a single post. Since, each user can only like a post once, all edges have the same weight. There were a total of 25 likes on this post at the time we crawled. Clusters are separated by the colors in the graph.

Facebook allows users to re-share posts on different places, such as their own wall, their friends wall, or a page or a group’s wall if they have the permission. Therefore, the weight of each link depends on how many times a user has shared the content.

\$facebook\rightarrow api('/faith/{Post-ID}/likes', [$limit])\$ - This call is similar to the previous call, except it only returns the graph created by likes action. In other words the links in the Social Graph returned represent a like from one person to another. The Graph returned has a star structure where the originator of the content is in the middle and all other users who have liked the content are connected to the person in the middle only. Since, each user can only have one like on the content, the weights of all the links are the same in this graph. Figure 5.7 shows the interactions networked returned by this API call for a public post made on the “Against Stop Online Piracy Act SOPA” page.

\$facebook\rightarrow api('/faith/{Community-ID}', [$limit])\$ - This call will return the whole Social Network of all interactions around all the posts in the given community. For each individual post we make separate calls to receive all the comments, likes, and shares of that particular post. Much like the separate calls described above. Fig: 5.8 shows the entire network formed around the community page “Against Stop Online Piracy Act SOPA”.

5. SIN: A Platform to Make Interactions in Social Networks Accessible

Figure 5.8: shows the complete social interactions network of the public Facebook page, “Against Stop Online Piracy Act SOPA”, over all the contents shared on the page. At generation time there were 5690 posts, 3864 likes directly on the posts, 8564 comments to the posts and 15529 likes on the posted comments. Clusters are separated by colors. For better visibility all nodes with a degree of lower than six have been removed from the graph; hence; the above graph contains 16% of the total nodes of the original graph and 36% of the total edges of the original graph.

$\texttt{facebook} \rightarrow \texttt{api('/faith/\{Community-ID\}/comments', [\$limit])}$
- This call is similar to \texttt{api('/faith/Community-ID')}, except we return the network that is created based on comments. In other words, We only return the portion of the network, where links are created by user comments and we do not take likes and shares into account. We have to send separate API calls to Facebook for each post in order to retrieve the comments, but it is still relatively faster than retrieving the whole graph since we do not need to send additional API calls for likes and shares.

$\texttt{facebook} \rightarrow \texttt{api('/faith/\{Community-ID\}/shares', [\$limit])}$
- This call is similar to \texttt{api('/faith/Community-ID')}, but here we only consider the sharing of the Community-ID posts when we create the graph. From what we have seen in our datasets this network is considerably smaller than the likes network, which suggest that on Facebook it is more likely that people like a post than re-share it.
$\text{facebook} \rightarrow \text{api}('/faith/\{\text{Community-ID}\}/likes', [\text{$\text{limit}$}])$ - This call is similar to $\text{api}('/faith/\{\text{Community-ID}\}$). The difference is that we only look at the Social Community based on users likes. This requires fewer requests to the Facebook servers, since we do not need to retrieve the comments and shares any more; therefore, it is significantly faster than getting the whole graph. On the downside, we believe that the graph returned by this call is relatively weaker than the graph returned by the previous call; since, likes have the least weight among actions a user can take on a post.

The optional $\text{$\text{limit}$}$ variable limits the number of items to be returned. Many times it is sufficient to receive a subset of the graph and are just interested in the latest interaction of users on a given context or content. Using the $\text{$\text{limit}$}$ variable the third party applications have the ability to retrieve as much data as they need and not more. For the community API calls, the $\text{$\text{limit}$}$ variable simply limits the number of posts returned and for each post we still retrieve the complete interaction data. For the post API calls, the $\text{$\text{limit}$}$ variable simply limits the number of interactions taken place on the given post. The default value for $\text{$\text{limit}$}$ is the same as Facebook, 25.

All these API calls return a response in JSON, which contains the weighted graph. We calculate the weight of the links based on the type of interaction (i.e. whether it’s a like, comment, or a share) and the number of interactions between users. The graph returned is a directed Graph as opposed to Facebook’s Social Network which is an undirected graph. We also include the timestamp on when each of these links were created, which allows us to recreate the whole social interaction network graph through time.

5.6 Security Issues and Implementation Challenges

Issues about security and privacy of user data is a cause of major concern in online social network API development as discussed in [19]. In our current implementation, we consider the related issues and our API only fetches public data. As part of our continued research efforts, we are currently looking at methods to anonymize the SIN returned in order to protect the privacy of users who have interacted in a given community. The major cost to select an algorithm that can successfully anonymize the data is based
on the algorithm’s effectiveness on preserving the original graph properties during the anonymization step [20–22]. A more detailed explanation of our future solutions is out of the scope of this paper.

Since, Facebook does not allow applications or platforms to store any of its data, we would need to get all the information we need through the API calls on the go. For API calls that try to get the structure of the whole community this requires a lot of calls depending on the amount of interactions on the page. One way for us to make things faster would be to use more threads and make our Facebook API calls in parallel with more nodes and save as much time as possible. We first make an initial call to get all the posts shared by a community and then in order to get the details of each post (i.e. likes, comments, likes of comments, and shares) we would make requests in parallel [23]. Also the amount of data retrieved with each Facebook API request is very limited due to their “Paging” mechanism [24]. For example, by default each api request to get posts of a Community only returns 25 results. In order to get the next 25 results one would need to make another API request to Facebook’s servers. For comments, likes, and shares, each call only returns 50 items and in order to get all of the items one would need to make many requests depending on how many interactions that have taken place on a given post.

As of today, there is a bug in Facebook’s API for retrieving information about re-shares of a post. We currently are not able to provide this data because of this bug [25]. We believe that once Facebook fixes this bug, our API should be able to retrieve the re-sharing data correctly.

5.7 Evaluation

We have taken many steps in order to deal with software failure while generating data from Facebook. There are timeout errors, when using a browser, the browser might time out, or PHP execution time may exceed the server configuration. These timeouts could be increased from the default value, but API errors due to making too many API requests too quickly, or any other server errors on Facebook’s side is harder to handle. At any time, we keep track of where we are in the process of data generating, so in case of a software failure we can simply continue fetching from where we left off instead of restarting from the beginning.
Table 5.2: This table shows the details of how many posts were successfully crawled before a software failure occurred. After each failure, the new round automatically starts again. Over a 20 hour long period we were able to crawl the data regarding interactions that had happened around 6479 posts shared on Against Stop Online Piracy Act SOPA fan page.

<table>
<thead>
<tr>
<th>Round #</th>
<th>Duration (secs)</th>
<th>Posts crawled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round 1</td>
<td>3307</td>
<td>393</td>
</tr>
<tr>
<td>Round 2</td>
<td>1410</td>
<td>188</td>
</tr>
<tr>
<td>Round 3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Round 4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Round 5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Round 6</td>
<td>12380</td>
<td>994</td>
</tr>
<tr>
<td>Round 7</td>
<td>7086</td>
<td>885</td>
</tr>
<tr>
<td>Round 8</td>
<td>5324</td>
<td>615</td>
</tr>
<tr>
<td>Round 9</td>
<td>7866</td>
<td>642</td>
</tr>
<tr>
<td>Round 10</td>
<td>9229</td>
<td>910</td>
</tr>
<tr>
<td>Round 11</td>
<td>10859</td>
<td>615</td>
</tr>
<tr>
<td>Round 12</td>
<td>8753</td>
<td>840</td>
</tr>
<tr>
<td>Round 13</td>
<td>5046</td>
<td>397</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>71267</strong></td>
<td><strong>6479</strong></td>
</tr>
</tbody>
</table>

Table 5.3: Shows how much time we saved during the second phase of our crawler by using a parallel approach with 10 threads.

<table>
<thead>
<tr>
<th>Community</th>
<th>Run Type</th>
<th>Average Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU Community (3041)</td>
<td>Sequential</td>
<td>12'460</td>
</tr>
<tr>
<td>EU Community (3041)</td>
<td>Parallel (10 threads)</td>
<td>1'166</td>
</tr>
<tr>
<td>Milwaukee Bucks (5400)</td>
<td>Sequential</td>
<td>79'897</td>
</tr>
<tr>
<td>Milwaukee Bucks (5400)</td>
<td>Parallel (10 threads)</td>
<td>5'189</td>
</tr>
<tr>
<td>New York Knicks (66020)</td>
<td>Sequential</td>
<td>976'864</td>
</tr>
<tr>
<td>New York Knicks (66020)</td>
<td>Parallel (10 threads)</td>
<td>65'563</td>
</tr>
<tr>
<td>Jay Leno (41152)</td>
<td>Sequential</td>
<td>179'636</td>
</tr>
<tr>
<td>Jay Leno (41152)</td>
<td>Parallel (10 threads)</td>
<td>16'320</td>
</tr>
</tbody>
</table>

For example, processing the community page of Against Stop Online Piracy Act SOPA, which is a public page, with over 117 thousand members crashed 12 times over a 20 hour period. Table 5.2 shows the number of pages posts fetched during each run and how long the code ran before failure. We fetched likes, comments, shares, and likes of comments of over 6000 posts of this community over a 20 hour long time period, which suggests that on average each post on this page takes on average 11 seconds to be fully fetched.

We use two phases for the API calls to generate the SIN of the whole community. The first phase is a sequential phase, where we keep making API calls to Facebook in order to get the full list of posts shared on the given community. This phase gives us an idea about the interactions around each post (i.e. likes count and comments count are given). In the second phase, we try to divide and balance the posts among different machines as
5. SIN: A Platform to Make Interactions in Social Networks Accessible

much as we can based on the amount of interactions done on posts and crawl the posts in parallel. Table 5.3 shows how much time we have saved during the second phase of crawling using 4 and 16 machines on Emulab instead of running the phase sequentially.

5.8 Future Work

We plan to find more ways to create social interactions networks. Other than liking, commenting, and sharing posts users interact in many other ways on OSNs. Users can also send content to each other through a message on Facebook. Currently, we do not crawl these interactions which might give valuable results. Other interesting examples are the relations on how users interact through third party applications built on top of Facebook’s platform. Obviously, we cannot crawl this data using Facebook’s API, but more traditional ways of crawling, such as parsing the html of the applications, might be used in order to extract this information. Another feature is the tagging done in Facebook. Everyone who shares a posts on Facebook or leaves a comment can tag their immediate friends in the post. This is another indication of interaction between users that we would like to consider in future versions of SIN.

We are planning on using our API to create applications that leverage the results. We talked about some of the ideas for applications in previous sections. Social Search Engine, Friend Suggestion, and a Dynamic News Feed are among the projects that we are planning to build using our API; furthermore, we would like to enhance some of our previous projects, such as the TrustWiki [26] application under FAITH, that relied on the traditional friendship networks by using the SIN networks. We believe that using the social interactions network will provide much more accurate information than the social networks provided by the current API.

5.9 Acknowledgements

This paper is based upon work supported in parts by the National Science Foundation under Grant No. CNS-1152320, IIP-1161015, and CNS-0832202, Army Research Office under the Multi-University Research Initiative (MURI) grant W911NF-07-1-0318.
5.10 References


5. SIN: A Platform to Make Interactions in Social Networks Accessible


5.10. References


The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Teng Wang, Keith C. Wang, Fredrik Erlandsson, S. Felix Wu, Robert Faris

Abstract

With the popularity of social media in recent years, it has been a critical topic for social network designer to understand the factors that influence user continued participation in online newsgroups. Our study examined how feedback with different opinions are associated with participants’ lifetime in online newsgroups. Firstly, we proposed a new method of classifying different opinions among user interaction contents. Generally, we leveraged user behavior information in online newsgroup to estimate their opinions and evaluated our classification results based on linguistic features. In addition, we also implemented this opinion classification method into our SINCERE system as a real-time service. Based on this opinion classification tool, we used survival analysis to examine how others’ feedback with different opinions influence user continued participation. In our experiment, we analyzed more than 88,770 interactions in official Occupy LA Facebook page. Our final result showed that not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation in online newsgroup. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent. This finding forms the basis to understand better how to improve online service in social media.
6. The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

6.1 Introduction

Social media has influenced people’s lifestyle from many aspects. It not only changes the way people collect news and information, but even reforms the way people communicate with each other. One of the most popular uses of social media is to support online newsgroups [1]. They allow people to seek latest news and exchange opinions on a wide variety of topics from entertainment, education to religions and politics. Despite the popularity of online newsgroups, it is very difficult to maintain them for a long time. Member participation in online newsgroups is often sparse and uneven [2]. In this paper, we examine the factors that influence user continued participation in online newsgroups. To be specific, our work focuses on the effect of feedback with different opinions.

In 2008, S.L. Johnson [3] gave a detailed definition of online groups from the views of group membership and interaction. Besides the criteria mentioned by S.L. Johnson, online newsgroups, as a special case of online groups also has its own characteristics: members in online newsgroups are less of social component in real life and most of them are strangers with few off-line communication. It is just these characteristics that make online newsgroups ideal resources for researchers to examine user influence. Firstly, because most of the members in newsgroups are strangers in real life, they would be more open to share their opinions online while people in private friendship group may have more concerns. Additionally, in online newsgroup, because most of people’s interaction happens in online environment, the offline influence will have very little effect on the analysis result. In all, user interaction data in online newsgroup can give us more comprehensive information of their mutual influence pattern.

In general, there are mainly two challenges to examine the effect of feedback with different opinions on user continued participation. Firstly of all, we need to find out an effective and efficient method to classify user comments into different opinions. In this paper, instead of only focusing on corpus itself, we leverage user behavior information and build user-like graph to do opinion classification. Besides that, we also use linguistic analysis tool to evaluate the classification results and develop this method into a real-time service on our SINCERE system\(^1\). The second challenge of this topic is to recognize the influence of different feedback among those

\(^1\)http://sincere.se/
factors on user continued participation. In this paper, we perform a large scale study of user interaction on Official Occupy LA Facebook page with totally 20,569 unique users, 56,937 comments, 31,833 posts and 66,758 likes. Based on this dataset, we use Cox proportional hazards model [4] in survival analysis to explore the relationship between user lifetime in online newsgroup and several explanatory variables. Our final result showed that the content of feedback in online newsgroup are significantly related to user continued participation: Not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent.

The rest of the paper is organized as follows. In section 6.2, we discuss related work on factors in user continued participation. Then we describe our opinion classification method and result evaluation in section 6.3. Based on this method, we design an experiment to analyze the influence of different feedback on user continued participation in section 6.4. Discussion and conclusions are talked about in section 6.5 and we also mention the limitations and future work in section 6.6.

6.2 Related Work

Moira et al. [5] grouped the theories of user online participation into three high level categories: social learning, distribution and feedback. In this paper, we focus our work on examining the influence of feedback on user continued participation. In previous work, theories of reciprocity [6] and reinforcement both proved that feedback from other users should predict long-term participation. As a careful analysis, the influence of feedback can come from the following three parts: the role of people who give the feedback, the amount of feedback and the content of feedback.

As to the role of people who give feedback, Steven Johnson [7] showed that interaction with online group leadership is associated with higher participation continuance and participation intensity. During the interaction with leadership in online group, the participant can feel psychological safety. Steven indicated that members of online groups with higher levels of psychological safety report higher levels of continued participation intentions.
6. The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Furthermore, people also found positive relationship between the amount of received feedback and user continued participation. Previous work [2, 5] on newcomers in online newsgroups showed that newcomers will be more likely to post again if anyone responds to their initial post. Besides the work on newcomers in online groups, Y. Wang et al. [8] examined the factors that influence the continued participation of any member in online health support groups. Their results showed that the count of all comments in the threads in a week in which the user had posted is highly correlated (r=.67) with user’s continued participation in online health support group.

When it comes to the content of feedback, there are many different ideas. Some previous work focused on newcomers in online newsgroups [2, 9]. E.Joyce [2] indicated that length, tone, content and personal affirmation are not found to be significant predictors of long-term engagement for newcomers in online newsgroup. However, Y. Wang et al. [8] examined the influence of feedback on all the group members and found significant relationship between the content of feedback comments and user commitment in online health support groups. In this paper, our work also considers all the group members. But different from Y. Wang’s work which analyzed different types of social supports among user interaction contents, we focus on different feedback opinions in online newsgroups.

The influence of feedback on different opinions is widely discussed in democratic deliberation research on political communication area [10, 11] and this issue is still controversial now. Generally, there are two different perspectives in this area. Some people believed that expressions of disagreement may violate expected norms of politeness in social interactions [12]. Mutz [13] proposed that the negative effects of disagreement may make people avoid political discussion and deliberation. However, J. Stromer Galley et al. [10] indicated that expressions of disagreement do not generally harm participants’ future participation and an interaction of agreement and disagreement can even boost expected future participation in democratic deliberations. However, their work only focused on people discussion content on political topics and most of them used questionnaire or phone survey as their data collection method, which severely limits their experiment sample. This paper, to the best knowledge of us, is the first work to give a large scale analysis of the influence of feedback with different opinions on user continued participation in online newsgroup. We offer a new opinion classification algorithm to distinguish feedback comments into
different opinions automatically, which allows us to do analysis on a much larger dataset. Additionally, instead of only focusing on political topics where people’s stance is always sharp and in opposition to each other, our experiment broadens user interaction data to general discussion topics in online newsgroups.

6.3 Opinion Classification

In Natural Language Processing area, there are many work discussing opinion classification in online threaded discussion [14, 15]. In the latest research work, Rob Abbott et al. [16] identified disagreement in political blogs using only lexical features. However, their machine learning method need a complex training process and every training model is only valuable for its source corpus. Instead of only focusing on the corpus itself, we can also leverage user behavior information in online newsgroups to assist us on opinion classification. Some recent work showed that user behavior features can be used to capture contextual information present in textual features very accurately [17]. Taking the public newsgroup on Facebook website as an example, besides the text of user interaction under each post, the information of like behavior is also a valuable tool for us to recognize user’s opinion. In this paper, we leverage the user-like-graph under each post to classify user interaction content into different opinion based on its link structure.

6.3.1 Data Collection

Our dataset is crawled from public pages on Facebook. Those pages are open to the public and anyone with a Facebook account can post or comment to existing posts. To be specific, each public page is a tree-shaped structure in which multiple posts are organized in temporal ordering. Facebook users can share latest news and discuss their opinions on different topics in those pages, which makes them ideal examples of online newsgroups. Under each post, we can get all of the user interaction information, such as the content of comments, likes information and their time-stamps. In general, our method leverages user like information on each comment to do opinion classification, which can be applied to analyze user interaction information on any Facebook public page in real time.
6.3.2 User-like graph

As to each post in a Facebook public page, we define a user-like graph $G = (V, E)$, where $V$ is a set of nodes and $E$ is a set of edges among $V$. For simplicity, in this paper, we consider the user-like graph undirected graph. A node stands for a user who has liked a comment or whose comments were liked by others in this post. An edge $e$ stands for the connection between two users $u$ and $v$ in this user-like graph and its weight $w_{uv}$ equals to the number of likes they have clicked on each other’s comments in this post. Figure 6.1 shows two examples of user-like graph of the posts in official Occupy LA Facebook group.

![Figures 6.1](image)

(a) Post1  (b) Post2

Figure 6.1: Examples of User-Like Graph

6.3.3 Opinion Classification method

The general idea of our opinion classification method is as follows: As to each post, we firstly classify people into different groups based on its user-like graph. Then we collect the comments made by different groups of people as different opinions contents. Therefore, as the first step, we need to find a partition of the user-like graph such that edges between different groups have a very low weight (which means people in different clusters are holding different opinions from each other) and the edges within a group have high weight (which means that people within the same cluster are...
6.3. Opinion Classification

holding similar opinions with each other). Obviously, this is a classic graph partitioning problem.

In order to make our output clusters reasonably large groups of nodes, we use the concept of Ratiocut [18, 19] to formulate our objective function. In user-like graph \( G = (V, E) \), we denote a subset of vertices \( A \subset V \) and \( \overline{A} \) for the complement of \( A \). In addition, for two set \( A, B \subset V \), we defines

\[
W(A, B) = \sum_{i \in A, j \in B} w_{ij}
\]

(6.1)

For a given number \( k \) of subsets, we choose a partition \( A_1, \ldots, A_k \) which minimizes

\[
Partition(A_1, A_2, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|}
\]

(6.2)

In the objective function 6.2, the size of a subset \( A \) of a user-like graph is measured by its number of vertices, i.e. the amount of people in this subset. It will get a small value if the clusters \( A_i \) are not too small. Therefore, this objective function can make our output clusters balanced, measured by both the connections between each cluster and the number of their vertices. Unfortunately, introducing balancing conditions makes our partition problem become NP hard. In order to solve this objective function, we choose to use spectral clustering algorithm [19], which is the most popular algorithm to solve relaxed versions of Ratiocut problem.

Based on our discussion above, we can now get a near-optimal partition of any user-like graph by spectral clustering algorithm when the number of cluster \( k \) is fixed. Now we need to decide the optimal number of cluster \( k \) for a user-like graph. In this paper, for simplicity, we only choose the cluster number from 1 to 2, i.e. we only want to decide if the comments under the same post can be clustered as two different opinions or people are just holding the same opinion in this post. In 2004, Newman and Girvan [20] proposed a modularity function which can directly measure the quality of a particular clustering of nodes in a graph. Their function \( Q \) measures the fraction of the edges in the graph that connect nodes in the same group minus the expected value of the same quantity in a graph with the same partitioning result but random connections between the nodes. If the sum weight of edges in the same groups is the same with that got by random connection, we will get \( Q = 0 \). And if the partitioning result has a strong community structure, the value of \( Q \) will be very closed to 1 [20].
In our method, after getting the partitioning result by spectral clustering algorithm with \( k = 2 \), we will use modularity \( Q \) to check the clustering quality of our partitioning result and decide the number of clusters in this user-like graph from 1 to 2. Newman’s work showed that real-world unweighted networks with high community structure generally have \( Q \) values within a range from 0.3 to 0.7. Figure 6.2 shows four examples of our clustering results and their corresponding \( Q \) values. Each of these graphs represents the user-like structure of one of the posts in Occupy LA Facebook group. The two different colors (pink and blue) stands for the spectral clustering result when we fix \( k = 2 \). We can find that \( Q \) value can be a good measurement for deciding the number of clusters in the user-like graph: The clustering results with \( Q \geq 0.2 \) show strong community structure, where \( k \) remains to be 2, while results with \( Q < 0.2 \) converge at some special nodes, where \( k \) is determined as 1. In this paper, we set \( Q = 0.2 \) as an threshold. Therefore, if the value of modularity measurement \( Q \) is less than 0.2 when \( k = 2 \), we will consider the user-like graph as one cluster, or we accept its partitioning result and cluster the nodes into two different groups. As the last step, for each post, we collect the comments made by different clusters of people as different opinions contents.

### 6.3.4 Evaluation from Linguistic Features

In order to evaluate the effectiveness of our opinion classification method, we select three Facebook public pages as our dataset: OccupyLA\(^2\), Occupy Wall Street\(^3\) and Occupy Together\(^4\). Starting from Sept. 2011, the Occupy movement call for people to protest against social and economic inequality, which attracts a wide range of people all round the world. Online social networks, during this time, plays an important role by offering people an ideal platform to get latest news and share opinions. Among all the public newsgroups about occupy movement on Facebook website, the public pages of Occupy Wall Street and Occupy LA are the largest two groups. and OccupyTogether is a comprehensive public page where people share information and opinions on any occupy movement. Additionally, in order to examine user interaction content in each post, the number of comments in the post that we analyze should reach a certain amount. In this experiment,

\(^2\)http://www.facebook.com/occupyLA
\(^3\)http://www.facebook.com/OccupyWallSt
\(^4\)http://www.facebook.com/OccupyTogether
we select all the posts which contain more than 35 comments inside. Finally, we get 205,198 comments among 1929 posts in these three public pages from Sept. 2011 to Apr. 2012. Because our dataset is very large, it is impossible for us to get the ground truth of the opinion classification result for each post by employing workers to rate manually. In this paper, we evaluate our opinion classification method from the linguistic features of user comments, which is a totally different view from that with the structure of user-like graph.

We use the Linguistic Inquiry and Word Count (LIWC) tool [21] to study the linguistic characteristic of user comments. LIWC is a popular tool which calculates the frequency with which words in a text match each
of 68 categories representing linguistic dimensions, psychological constructs and personal concerns \cite{li}. Many previous work \cite{li, mo} have shown that the categories in LIWC are effective in determining linguistic differences on attentional focus and emotionality of the relationship. In this paper, we consider 39 categories in 13 areas, which is shown in Table 6.1. Among these 39 categories, 32 of them belong to psychological processes and 7 of them belong to personal concerns area. The analysis results in \cite{li} and \cite{mo} reveal that the LIWC scores on psychological process are very helpful on identifying people’s emotional attitudes, i.e. agreement and disagreement opinions, in their online comments. Besides that, \cite{ni} also showed the LIWC scores of categories on personal concern are also effective on identifying people’s different references in online groups.

During the evaluation process, our theoretical basis is: Comments on different opinions have different characteristics on their linguistic features \cite{mo}. After using our opinion classification method to analyze those 1929 posts, 1216 posts are considered with two different opinions inside. As to each of these 1216 posts, we denote \(G\) as the set of comments of one post and \(P, Q \subset G\) as the two groups of comments on different opinions. Suppose there are \(m\) comments in set \(P\) and \(n\) comments in set \(Q\). The LIWC analysis result of one comment is denoted as \(S_{ij}\), where \(i\) denotes the ID of the comment in set \(P\) or \(Q\) and \(j\) denotes the serial number of the 39 categories. In set \(P\), we define the LIWC scores in the \(i^{th}\) category as \(X_{pj} = [S_{1j}, S_{2j}, ..., S_{nj}]^T\) \((j \in [1, 39])\). Similarly, in set \(Q\), the LIWC scores in the \(i^{th}\) category is defined as \(X_{qj} = [S_{1j}, S_{2j}, ..., S_{mj}]^T\) \((j \in [1, 39])\). Therefore, the LIWC result of these two groups of comments can be denoted as follows:

\[
L(P) = [X_{p1}, X_{p2}, ..., X_{p39}] \quad (6.3)
\]

\[
L(Q) = [X_{q1}, X_{q2}, ..., X_{q39}] \quad (6.4)
\]

Then as to each of the 39 LIWC categories, we use Welch’s t-test \cite{wel} to test whether the means of the two population \(X_{pj}\) and \(X_{qj}\) are different from each other. For LIWC category \(j\), we denote the p-value of their t test as \(p_j\) \((j \in [1, 39])\). So we get the result of linguistic comparison as

\[
\text{Compare}(P,Q) = [p_1, p_2, ..., p_{39}] \quad (6.5)
\]

Because the posts in our dataset cover diversified topics, we cannot limit their different linguistic features show in only one particular item of those
6.3. Opinion Classification

Table 6.1: The 39 textual Categories in 13 areas used in our linguistic evaluation. Areas marked with $^1$ are psychological processes, and areas marked with $^2$ are personal concerns.

<table>
<thead>
<tr>
<th>Areas</th>
<th>Categories</th>
<th>Selected Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social $^1$</td>
<td>Family</td>
<td>mate, talk, they, child</td>
</tr>
<tr>
<td></td>
<td>Friends</td>
<td>husband, aunt</td>
</tr>
<tr>
<td></td>
<td>Humans</td>
<td>adult, baby, boy</td>
</tr>
<tr>
<td>Affective $^1$</td>
<td>Pos. Emotion</td>
<td>happy, cried, abandon</td>
</tr>
<tr>
<td></td>
<td>Neg. Emotion</td>
<td>love, nice, sweet</td>
</tr>
<tr>
<td></td>
<td>Anxiety</td>
<td>hurt, ugly, nasty</td>
</tr>
<tr>
<td></td>
<td>Anger</td>
<td>hate, kill, annoyed</td>
</tr>
<tr>
<td></td>
<td>Sadness</td>
<td>crying, grief, sad</td>
</tr>
<tr>
<td>Cognitive $^1$</td>
<td>Insight</td>
<td>cause, know, ought</td>
</tr>
<tr>
<td></td>
<td>Causation</td>
<td>think, know, consider</td>
</tr>
<tr>
<td></td>
<td>Discrepancy</td>
<td>because, effect, hence</td>
</tr>
<tr>
<td></td>
<td>Tentative</td>
<td>should, would, could</td>
</tr>
<tr>
<td></td>
<td>Certainty</td>
<td>always, never</td>
</tr>
<tr>
<td></td>
<td>Inhibition</td>
<td>block, constrain, stop</td>
</tr>
<tr>
<td></td>
<td>Inclusive</td>
<td>and, with, include</td>
</tr>
<tr>
<td></td>
<td>Exclusive</td>
<td>but, without, exclude</td>
</tr>
<tr>
<td>Perceptual $^1$</td>
<td>See</td>
<td>heard, feeling</td>
</tr>
<tr>
<td></td>
<td>Hear</td>
<td>View, saw, seen</td>
</tr>
<tr>
<td></td>
<td>Feel</td>
<td>Listen, hearing, touch</td>
</tr>
<tr>
<td>Biological $^1$</td>
<td>Body</td>
<td>eat, blood, pain</td>
</tr>
<tr>
<td></td>
<td>Health</td>
<td>cheek, hands, spit</td>
</tr>
<tr>
<td></td>
<td>Sexual</td>
<td>clinic, flu, pill</td>
</tr>
<tr>
<td></td>
<td>Ingestion</td>
<td>horny, love, incest</td>
</tr>
<tr>
<td>Relativity $^1$</td>
<td>Motion</td>
<td>area, bend, exit, stop</td>
</tr>
<tr>
<td></td>
<td>Space</td>
<td>arrive, car, go</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>down, in, thin, season</td>
</tr>
<tr>
<td>Work $^2$</td>
<td></td>
<td>job, majors, xerox</td>
</tr>
<tr>
<td>Achievement $^2$</td>
<td></td>
<td>earn, hero, win</td>
</tr>
<tr>
<td>Leisure $^2$</td>
<td></td>
<td>cook, chat, movie</td>
</tr>
<tr>
<td>Home $^2$</td>
<td></td>
<td>kitchen, family</td>
</tr>
<tr>
<td>Money $^2$</td>
<td></td>
<td>audit, cash, owe</td>
</tr>
<tr>
<td>Religion $^2$</td>
<td></td>
<td>altar, church, mosque</td>
</tr>
<tr>
<td>Death $^2$</td>
<td></td>
<td>bury, coffin, kill</td>
</tr>
</tbody>
</table>
6. The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Table 6.2: Evaluation Results from Linguistic Features.

<table>
<thead>
<tr>
<th>Group</th>
<th>Total</th>
<th>Graph</th>
<th>Linguistic</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupy LA</td>
<td>380</td>
<td>264</td>
<td>200</td>
<td>75.8%</td>
</tr>
<tr>
<td>Occupy Together</td>
<td>451</td>
<td>285</td>
<td>230</td>
<td>80.7%</td>
</tr>
<tr>
<td>Occupy Wall Str</td>
<td>1098</td>
<td>667</td>
<td>529</td>
<td>79.3%</td>
</tr>
</tbody>
</table>

39 categories. Therefore, as the final step, if any of these 39 p-values is less than the predetermined significance level \( \alpha (= 0.05) \), we will conclude the two groups of comments got by our opinion classification method reveal different characteristics on their linguistic features, which indicates that our opinion classification result is acceptable for this post. Table 6.2 shows our evaluation results of all the 1216 posts in the three Facebook public pages. The item Total denotes the amount of posts with more than 35 comments in each online newsgroup. The item Graph denotes the amount of posts which are recognized as two opinion groups inside by our opinion classification method. The item Linguistic denotes the amount of posts within Graph which reveal different linguistic features. From the result we can see that 959 of those 1216 posts reveal different linguistic features, which achieves an accuracy of 78.9%.

6.3.5 System Development

Besides theoretical evaluation, we also implemented this opinion classification method into our SINCERE system (Social Interactive Networking and Conversation Entropy Ranking Engine) as a real-time service. SINCERE system is a diversified search engine based on user social informatics. Its database offers all the interactions (such as likes, shares, comments, timestamps) of 1391 Facebook public pages.

Figure 6.3 shows a screen shot of our opinion classification service on SINCERE. This system can automatically classify the comments of this post into different opinions (one or two opinions) and show the result in a pull-down menu. If there are only one opinion group recognized, the system will list all of the comments in one list. Otherwise, it will show the different groups of comments on two parallel columns, which is the same style with the example in Figure 6.3. From the content of those classified comments in our example, we can find that the opinion classification result is very effective: Under the post where people are discussing the recent
6.4 The Influence of Feedback with Different Opinions on User Continued Participation

In section 6.3, we introduced a new opinion classification method in online newsgroup and evaluated its effectiveness from linguistic features. In this section, we use this method as a tool to analyze the influence of feedback with different opinions on user continued participation in online newsgroup. The dataset we use in this experiment is crawled from the official Occupy LA Facebook public page. We collected all the posts, comments and like information on this page from Sep. 2011 to Apr. 2012. During this period of time, there were in total of 20,569 users who posted 56,937 comments belonging to 31,833 posts. Additionally, there are also 66,758 likes among all of these comments. In order to analyze the influence of feedback on
online newsgroup participants, the user we examine should have enough amount of activity record in this group. Therefore, in this experiment, we are only interested in those users who have made more than 20 comments in Occupy LA public page, which includes 622 users in total.

We use Cox proportional hazard (PH) model [24] in survival analysis [4] to explore the relationship between user lifetime in this newsgroup and several explanatory variables on the influence of feedback comments. Survival analysis is a main method to examine and model the time it takes for some special events to occur [24]. In our experiment, the specified event is defined as the end of the user’s active lifetime on this page. In previous work, this technique has been widely used in medical science, sociology and engineering [9]. As the most widely used method of survival analysis, Cox regression can provide estimated coefficients for each covariate and allow the assessment of the impact of multiple covariates in the same model.

6.4.1 Experiment Design

6.4.1.1 Dependent Variable

- **Lifetime**: Because we are interested people’s active lifetime on online newsgroups, user’s first and last comment time may not be ideal indicators for the actual user lifetime in online newsgroup. [25] gave a definition on the lifetime of IRC channels based on their level of activity. In our experiment, we include user comment frequency into the definition of user lifetime. In our dataset, the total time duration is 220 days. Firstly, we divide this period of time into 22 time-blocks with the same 10-day interval. Then as to each participant in this newsgroup, we considered him/her to start his/her activity when he/she makes more than 3 comments in two consecutive time-blocks. And we consider him/her to left this newsgroup at the day after when he/she gives no comments in its following two consecutive time-blocks. Figure 6.4 shows the distribution of user lifetime of those selected 622 participants in Occupy Los Angeles public page. Their online lifetime ranges from 202 days to 0 days. In addition, because people whose last comment is found within the last time-block may still be participating in this Occupy LA group, we treat them as right censored in the survival analysis.
6.4. The Influence of Feedback with Different Opinions on User Continued Participation

Figure 6.4: Distribution of User Lifetime in Occupy Los Angeles Group.

6.4.1.2 Control Variables

- **OriginalPostWriter**: Among all the 31,833 posts in Occupy Los Angeles public page, most of them are written by the official maintainers of this public homepage named *OccupyLA*. However, there are also many posts written by normal users. Considering people who write original posts on this page may have different participating enthusiasm than people who just take part in discussions started by others, for each of those 622 members we are interested in, we define a control variable OriginalPostWriter as the percentage of his/her original posts among all of his comments in Occupy Los Angeles public page.

- **NonDiscussionPostsInvolved (NonDiscussion)**: Among all the posts written by the official maintainer of Occupy Los Angeles group, we also split them into two parts: one is the posts with more than 35 comments inside and another is those with less than 35 comments. The posts in the first part are considered as discussion posts. As to the posts in the second part, we use them to define a control variable NonDiscussionPostsInvolve. It is the number of an individual’s comments in non-discussion posts divided by the amount of his/her comments in all posts.

- **ReceivedCommentsPerActivity (ComPerAct)**: In each discussion post in Occupy LA group, there is not exactly reply during user interaction: people just make comments one after another along the
timeline of each post. So we define the following comments within three hours after the individual makes a comment as his/her received replies. We assume this people read all the following comments within three hours after his/her comment and regard them as feedback for his/her activity. Based on this definition, we calculated the average number of received replies during the three hours after the individual makes a comment.

6.4.1.3 Independent Variables

- **SameOpinionsPercentage(SamePercent):** This variable measures the percentage of the replies in the same opinion with this person among all of the replies this person received. To be specific, during the three hours after this person makes a comment, all the received comments can be classified into three groups: comments with the same opinion with him/her in this post (these comments are made by people who are in the same opinion group in this post); comment with different opinion with him/her in this post (these comments are made by people who are in different opinion group in this post); comments with unclear opinion (these comments are made by people who neither clicked like on other’s comments nor are liked by others).

- **DifferentOpinionsPercentage(DifferentPercent):** This variable measures the percentage of the replies in different opinion with this person among all of the replies this person received.

Table 6.3 shows the descriptive statistics of all these variables. Based on the definitions above, we standardize all the control and independent variables with a mean of zero and standard deviation of one and use Cox regression model to analyze the relationship between the user lifetime in Occupy LA group and these variables.

6.4.2 Experiment Result

Results of Cox regression model are shown in Table 6.4. The exponential coefficient indicates the direction of the effect of variables: when \( \exp(\text{coef}) \) is smaller than 1, it represents a positive relationship between the variable and the lifetime. For example, because the \( \exp(\text{coef}) \) of OriginalPostWriter
6.4. The Influence of Feedback with Different Opinions on User Continued Participation

Table 6.3: Descriptive Statistics for Variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalPostWriter</td>
<td>0</td>
<td>1</td>
<td>0.320</td>
<td>0.318</td>
</tr>
<tr>
<td>NonDiscussion</td>
<td>0</td>
<td>1</td>
<td>0.415</td>
<td>0.237</td>
</tr>
<tr>
<td>ComPerAct</td>
<td>0</td>
<td>57.625</td>
<td>5.679</td>
<td>6.900</td>
</tr>
<tr>
<td>SamePercent</td>
<td>0</td>
<td>1</td>
<td>0.375</td>
<td>0.283</td>
</tr>
<tr>
<td>DifferentPercent</td>
<td>0</td>
<td>0.8</td>
<td>0.110</td>
<td>0.133</td>
</tr>
</tbody>
</table>

Among the results of two control variables, the exp(coef) value (0.652) of OriginalPostWriter tells us that when all other variables are in average values, members who posted an average of one standard deviation (0.320) more original posts were 35\% more likely to remain in the group. Similarly, the exp(coef) value of NonDiscussionPostsInvolved indicates that members who involves one standard deviation more in Non-Discussion Posts were 28\% more likely to remain in the group. And those who received a standard deviation more feedback after his/her comment revealed to be 23\% more likely to remain in the group.

In addition, both of the independent variables, SamePercent and DifferentPercent show significant influence on user survival rate in social newsgroup. Members who received a standard deviation more feedback on the same opinion with him/her were 19\% more likely to remain in the group and those who received a standard deviation more feedback on different opinions also showed improved preference to remain in the group, with the rate of 13\%. In other words, both positive and negative feedbacks from others can motivate user longtime participation in social newsgroups, and

\[35\% = ((1 - 0.652) \times 100\%)]
positive feedbacks has a slightly more driving effect on it.

Furthermore, we also consider the interaction between independent variables and those with the control variable ComPerAct. Firstly, when controlling all of the control variables, the two independent variables SamePercent and DifferentPercent revealed super-linear positive influence on the user lifetime: not only each of them reveals positive influence on the lifetime, their interaction (SamePercent * DifferentPercent) shows significant positive relationship with user longtime participation as well. In other words, compared with people received an average number of positive feedback and negative feedback, members who received a standard deviation more feedback with both positive and negative feedback revealed to be 46% more likely to remain in the group, which is much more higher than the result of their linear combination 29%. What is more, we also find that the interaction between ComPerAct and SamePercent/DifferentPercent shows negative influence on the user lifetime. Taking the interaction ComPerAct * DifferentPercent as an example, it indicates that the common influence of ComPerAct and DifferentPercent 23% is not as great as that from their linear combination 33%. This may be the result of duplication influence of them on user lifetime.

### 6.5 Discussion and Conclusions

In this paper, we built user-like graph to classify different opinions of user interaction content in online newsgroup. And then we use Cox regression model to evaluate the influence of feedback with different opinions on user lifetime in official Occupy LA Facebook group. From the results shown above on different variables, we can get many important conclusions which can help designers of social network system to get a deeper understanding of user behavior and improve their online service.

Among the three control variables, firstly, the results show that members who started more original posts revealed a longer lifetime in this newsgroup. Therefore, the designers of Online social network can motivate

---

\[^6\] 46\% = (1-0.813*0.875*0.761)*100\%
\[^7\] 29\% = (1-0.813*0.875)*100\%
\[^8\] 23\% = (1-0.77*0.875*1.140)*100\%
\[^9\] 33\% = (1-0.77*0.875)*100\%
users longtime participation by offering more opportunities for normal users to post their own news and become a discussion starter. Secondly, we find those who have more comments in Non-discussion posts preferred to stay in this newsgroup longer. Our explanation for this result is that although Non-discussion posts do not have enough comments to host a discussion environment, the information it offered is also very important. D. Fisher et al. [26] indicates that the topic of forum is one of the factors which we can use to predict user engagement. Therefore, this result tells website designers that despite the importance of discussion posts with many comments and people involved, they should not neglect the information offered by Non-discussion posts. Last but not the least, the result of control variable ComPerAct shows that the more feedback one individual receives after his/her comment, the more likely he/she will remain in this newsgroup.

When controlling all the control variables, our final result indicates that not only the number of replies, but also their content has a significant influence on user’s commitment to online newsgroup. To be specific, our conclusion is that not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent. Based on our conclusion, we think that although feedback with different opinions may result in a unpleasant interaction, they help to form a comprehensive and healthy discussion environment. In psychological area, De Dreu et. al. [11] indicates that conflicts in group discussion can increase creativity and divergent thinking. Therefore, we believe that when people are involved in a discussion with different perspectives, their understanding of certain topics may be improved, which will increase people’s evaluation on this online newsgroup and motivate their future participation. This result tells website designers that they can also motivate user continued participation by digging into user interaction contents. For example, website designers can highlight or send notices to a user when feedback from different opinions show up in his/her involved discussion posts.
6. Limitations and Future Work

One of the limitations in this paper comes from our opinion classification method. Firstly, we modeled the user-like graph as an undirected graph which makes no discrimination between the writer of one comment and the likers of that comments. However, the degree of their preference on certain opinions may different from each other. Future work can model it as a directed graph so as to get a more accurate model on user opinions. Secondly, in this paper, we only choose the number of opinion groups in one post between 1 to 2 and classify the person in user-like graph into either one of the opinion groups. However, user interaction in online newsgroup is a much more complex scenario: Different from those content in debate forum [16] where people’s opinions form apparently two parties, the discussion in online newsgroup may consist of many different opinions, each of which starts from a different view and is not necessarily opposed to others. Therefore, another future work can consider more than two opinion-groups among user interaction content and assign them various degrees as feedback comments instead of simply agreement and disagreement.

Furthermore, when we analyze the influence of feedback on user continued participation, we average the effect of the feedback comments during the whole user’s lifetime in online newsgroup. However, the influence of feedback may change during user participation in one group and different factors of feedback, such as the role of speaker, the amount of feedback and the content of feedback, may have various dynamic influence pattern during user participation. Future work can analyze how those different factors are changing their influence during user lifetime in online newsgroup.

6.7 Acknowledgment

We want to thank Mohammad Rezaur Rahman and Roozbeh Nia for their help on data collection. We are also very grateful to Ran Cao in Statistics Department of Michigan State University for her insightful comments on the survival analysis part. This paper is based upon work supported in parts by the National Science Foundation under Grant No. CNS- 1152320, IIP-1161015, and CNS-0832202, Army Research Office under the Multi-University Research Initiative (MURI) grant W911NF-07-1-0318.
6.8 References


6. The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups


[23] B. L. Welch. “The generalization of student’s’ problem when several different population variances are involved”. In: Biometrika 34.1/2 (1947), pp. 28–35.


Mining User Deliberation and Bias in Online Newsgroups: A Dynamic View

Teng Wang, Fredrik Erlandsson, S. Felix Wu

Abstract

Social media is changing many different aspects of our lives. By participating in online discussions, people exchange opinions on various topics, shape their stances, and gradually form their own characteristics. In this paper, we propose a framework for identifying online user characteristics and understanding the formation of user deliberation and bias in online newsgroups.

In the first section of the paper, we propose a dynamic user-like graph model for recognizing user deliberation and bias automatically in online newsgroups. In addition, we evaluate our identification results with linguistic features and implement this model in our SINCERE system as a real-time service. In the second section, after applying this model to two large online newsgroups, we analyze the influence of early discussion context on the formation of user characteristics. Our conclusion is that user deliberation and bias are a product of situations, not simply dispositions: confronting disagreement in unfamiliar circumstances promotes more consideration of different opinions, while recurring conflict in familiar circumstances evokes close-minded behavior and bias. Based on this observation, we also build a supervised learning model to predict user deliberation and bias at an early online life-stage. Our results show that having only the first three months of users’ interaction data generates an F1 accuracy level of around 70% in predicting user deliberation and bias in online newsgroups. This work has practical significance for people who design and maintain online newsgroups. It yields new insights into opinion diffusion and has wide potential applications in politics, education, and online social media.
7. Mining User Deliberation and Bias in Online Newsgroups: A Dynamic View

7.1 Introduction

As one of the most popular uses of social media, online newsgroups have generated torrents of opinion-based data on a wide variety of topics. During online discussion, people exchange ideas, influence group interactional norms [1], and change opinion dynamics [2, 3] over time. On the other hand, users are also influenced by the content of others. In online newsgroups, people shape their stances and build their own characteristics during interactions [4, 5].

In general, people usually show the following two characteristics in online environments: deliberation and bias. Deliberation is a process of thoughtfully weighing options, a concept mainly studied in political science and sociology. In political science, deliberation is a practice in citizenship that allows people with differing political goals to listen to alternative viewpoints and seek common ground [6]. In sociology, deliberative skill involves the ability to listen actively without being tempted to respond to statements in a rash or disruptive manner [7]. All of these explanations emphasize open-mindedness and matter-of-factness during interaction. In contrast, bias is an inclination to hold on to a partial perspective and refuse to even consider the possible merits of alternative points of view [8]. Bias can be found in almost any scenario where opinions are expressed. In politics, for example, biased views are considered partisan, a label placed on people who practice selective attention toward the views of their political party and show closed-mindedness about alternatives [6]. Understanding the formation of deliberation and bias has profound implications for many application areas:

- **Education**: Deliberation is regarded as a cognitively oriented collaborative skill [9]. Many previous studies in learning science have focused on developing educational software in a collaborative environment to support development in self-regulated learning skills and reflective reasoning skills [10, 11]. With the formation model of deliberation and bias, teachers can adjust their methods to effectively teach these skills at an early stage.

- **Politics**: All campaigns confront three distinct populations: supporters, opponents, and spectators. To get elected, candidates not only need to activate the enthusiasm of their supporters and guard against
the attacks of their opponents, but also require new supporters. In light of this, spectators become very important targets. By gaining insights into the formation of deliberation and bias, campaigners are able to not only detect open-minded people more accurately, but also target those people at an earlier stage to make their ads campaign more effective.

- **Social media:** Maintaining active discussion groups for a long time is the most important job for website developers. Previous research [12, 13] on group formation indicates that a healthy group should consist of different roles of participants at different stages. By detecting and predicting deliberation and bias in online groups effectively, social website designers can add more functions to promote group activity, such as inviting users with differing characteristics into certain discussion groups to balance member composition.

However, it is not an easy task to study deliberation and bias. Both are composite characteristics that are deeply involved in the evolutionary process of user opinions. In general, there are two main challenges. The first is to figure out an effective and efficient method to detect user deliberation and bias from online activities. The second is to identify the factors that influence the formation of user deliberation and bias. In this paper, we consider the definition of the deliberation and bias from the perspective of user dynamic behavior. In summary, our main contributions are listed as follows:

- We design a dynamic model of user-like graph and use the evolutionary path of user opinions to identify user characteristics. In order to analyze the big dataset in online social networks, we develop an unsupervised learning algorithm to automatically process user behavior data without any model training step. In addition, we leverage linguistic analysis tools to evaluate our identification results and implement the model in our SINCERE\(^1\) system as a real-time service. Considering the diversity and huge amount of our dataset, it is impossible for us to obtain enough ground truth by employing human workers to rate user interaction records manually. The linguistic evaluation step, which is discussed in Section 3, offers an

\(^1\)http://sincere.se
alternative way of validating behavior analysis results from a different perspective.

- We analyze the influence of different interaction contexts on user characteristics in large online datasets. We perform a large-scale study of user interaction on two online newsgroups: the official Occupy Wall Street Facebook page\(^2\) and Occupy Together Facebook page\(^3\). These two newsgroups include a total of 101,553 unique users, 311,302 comments, 175,088 posts, and 1,914,718 likes. Our conclusion is that user deliberation and bias are a product of situations, not simply feedback content or dispositions. To be specific, confronting disagreement in unfamiliar circumstances promotes more consideration of different opinions, while recurring conflict in familiar circumstances evokes close-minded behavior and bias. Furthermore, we also develop a framework to predict user deliberation and bias from their behavior information at an early stage. Our results show that having only the first three months of users’ interaction data generates an F1 accuracy level of around 70%.

The rest of the paper is organized as follows. In Section 2, we talk about related work on the identification and formation of user deliberation and bias in online newsgroups. Then, in Section 3, we discuss our main contribution of identifying user online characteristics by user-like graph and its diffusion model. Based on this method, in Section 4, we analyze the influence of context on user characteristic formation in online newsgroups. Conclusions are in Section 5.

### 7.2 Related Work

In this section, we review previous research on user deliberation and bias in online newsgroups.

**Identification of Deliberation and Bias.** In the past few decades, deliberation and bias have been widely discussed in political science and sociology. Among previous work, most researchers design their experiments using questionnaires or volunteer surveys [6, 14]. However, these methods

\(^2\)http://www.facebook.com/OccupyWallSt
\(^3\)http://www.facebook.com/OccupyTogether
7.2. Related Work

severely limit the scale of experimental datasets. In order to deal with the huge amount of data in online social networks, it is necessary to use an automated method to identify user characteristics from their online activities. In recent years, researchers in Natural Language Processing have proposed linguistic models to identify user deliberation and bias in online discussion forums: Xiaoxi et al. [9] build the $L_1$ Regularized Logistic Regression model to identify social deliberative behavior using lexical, discourse, and gender demographic features. Zhao et al. [15] study confirmation bias on controversial topics and identify biased user groups through the use of social context analysis. Tae et al. [16] propose a model using Amazon Mechanical Turk judgements and show that lexical indicators strongly associate with bias. All of these machine learning methods require a complex training process, and every model is only valuable for its source corpus. In online newsgroups, textual features of user-generated content are highly dynamic, and it is very difficult to find ground-truth or labeled data to help us do model training. Therefore, in this paper, we leverage user behavior features to capture contextual information. The most similar work to our method is the transfer learning framework proposed by Pedro et al. [5]. They exploit user endorsement information to do real-time sentiment analysis, but their static model only deals with biased opinions with high degrees of opinion polarization. In online newsgroups, users often have many different perspectives that fall outside of the binary of agree and disagree. This paper, to the best of our knowledge, is the first work to propose a dynamic user behavior model that identifies both user deliberation and bias at the same time.

**Formation of Deliberation and Bias.** In general, the formation of deliberation and bias is influenced by user dispositions and the content of interaction. Disposition is the natural tendency of each individual to take a certain position in any field [17]. Previous research [18] shows that people often use their prior opinions to make an evaluation. Instead of using all available information, people usually reply based on their heuristics or cognitive shortcuts [19]. As to the content of interaction, framing is the most common effect. It refers to alternative conceptualizations of an issue or event. A number of studies over the past quarter-century show that framing effects can substantially shape opinions and stance [20, 21]. Furthermore, in recent years, researchers have shown that the time effect of interaction content also influences the formation of different opinions
and stances. In this paper, we claim that information processing is also context specific: confronting disagreement in unfamiliar circumstances promotes more consideration of different opinions, while recurring conflict in familiar circumstances evokes close-minded behavior and bias. Our result matches the conclusion of the civic engagement research by Michael et al [6] in political science. However, their work emphasizes emotional effects on citizenship under different circumstances, and their experiment is limited to political content with only 215 participants. Our work gives a large-scale analysis of the influence of context on user deliberation and bias. Additionally, we propose a prediction model from behavior information taken at an early stage.

7.3 Identification of Deliberation and Bias

The interactions of individuals in online newsgroups are temporal and dynamic in essence. Examining their evolutionary pattern and network structures provides deep insights into user activities online. Dynamic graph models have been proposed to do outliers detection [22, 23] and topic extraction [24]. In this chapter, we focus on identifying user characteristics in online newsgroups.

7.3.1 Data Collection

Our data is crawled from public pages on Facebook [25]. These pages are open to the public, and anyone can leave comments on existing posts. We collect all the user interaction information from these public newsgroups, including the content of comments, user-like information, and their timestamps. User-like information includes all the clickstream data when people press the Like button on others’ Facebook comments. In this paper, we select two Facebook public pages as our dataset: Occupy Wall Street and Occupy Together. These two news pages are excellent datasets for studying user deliberation and bias: Both of them have been active since the beginning of the Occupy movement in September 2011. Occupy Wall Street is the largest newsgroup in the Occupy movement, and Occupy Together is a comprehensive public page where people share information and opinions on any Occupy movements. Moreover, in both of these two public newsgroups, users share their opinions on various topics that include not only political events, but also local daily news. Unlike online debate
forums, where user opinions are extremely polarized, our dataset includes user comments from many different perspectives, which is an ideal resource for analyzing user characteristics. In summary, we collect a total of 311,302 comments on 175,088 posts from September 2011 to July 2012. Statistics of the two datasets are given in Table 7.1.

Table 7.1: Statistics of the Occupy Together and the Occupy Wall Street Newsgroups.

<table>
<thead>
<tr>
<th></th>
<th>OccupyTogether</th>
<th>OccupyWallSt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (Days)</td>
<td>299</td>
<td>247</td>
</tr>
<tr>
<td>Posts</td>
<td>79,931</td>
<td>95,157</td>
</tr>
<tr>
<td>Comments</td>
<td>74,835</td>
<td>236,457</td>
</tr>
<tr>
<td>Likes</td>
<td>509,665</td>
<td>1,405,053</td>
</tr>
</tbody>
</table>

7.3.2 Dynamic User-like Graph Model

The general idea of our dynamic user-like graph model is as follows: first, we use a graph partition algorithm to classify user opinions into different subgroups at each snapshot of the user-like graph. Then, we connect those opinion subgroups together to build opinion diffusion paths. Finally, we identify deliberation and bias by analyzing user’s stability and transition among the detected paths.

**User-like Graph.** For all the comments data in one Facebook public newsgroup, we divide them into a sequence of consecutive static graph snapshots \( \{G_1, G_2, ..., G_n\} \). At each time stamp \( t_k \), the graph \( G_{t_k} \) aggregates all the user-like information \([26]\) within the interval \([t_{k-1}, t_k]\). In the graph \( G_{t_k} = (V_{t_k}, E_{t_k}) \), nodes \( V_{t_k} \) stand for the users who liked a comment or whose comments were liked by others during the time slot \([t_{k-1}, t_k]\). Edges \( E_{t_k} \) stand for the like connections between those users. Figure 7.1 shows an example of user-like graph. In our experiment, considering the dynamic features of online newsgroups, we set the interval time \( \Delta T = [t_{k-1}, t_k] \) as one hour for each time slot. In summary, we build an undirected weighted user-like graph for each hour since the beginning of data collection.

**Opinion Classification.** In previous work \([26]\), Teng et al. showed that the user-like graph is a powerful tool to do opinion classification: people who have strong like-connections with each other show similar opinions in online newsgroups. However, their method can only classify user comments into one or two opinions. In this paper, we use the fast greedy modularity
optimization algorithm [27] to classify user opinions into more than two subgroups.

In 2004, Newman and Girvan proposed a modularity function to measure the quality of a particular clustering of nodes in a graph [28]. Vertices are divided into communities such that vertex $v$ belongs to community $c_v$ and vertex $w$ belongs to community $c_w$. We use $A_{vw}$ to record the connection within the network: $A_{vw} = 1$ if vertices $v$ and $w$ are connected. Otherwise, $A_{vw} = 0$. Based on the work by Clauset and Newman [27], the modularity $Q$ can be written as

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{d_v d_w}{2m}] \delta(c_v, c_w)$$  \hspace{1cm} (7.1)$$

where $d_v = \sum_w A_{vw}$ is the degree of a vertex $v$ and $m = \frac{1}{2} \sum_{vw} A_{vw}$ is the number of edges in the graph. At each step of the algorithm operation, we go through the amalgamations of each pair of communities in the user-like graph and perform the amalgamation which can improve modularity $Q$ the most. Although this fast greedy algorithm can only give us a locally optimal partition of the user-like graph, it is a very efficient algorithm that runs in essentially linear time on some real-world networks [27]. Considering
7.3. Identification of Deliberation and Bias

Figure 7.2: Examples of Opinion Diffusion Paths Detected in the Occupy Together Newsgroup.

the huge amount of user interaction data and the requirement of real-time analysis, the fast greedy algorithm is the most suitable algorithm for our framework.

**Opinion Diffusion Paths.** Using the modularity optimization algorithm, we classify users into many different opinion subgroups at each static graph snapshot. In order to build opinion diffusion paths for the whole newsgroup, we need to connect the subgroups with the same opinion together over different time stamps. At this step, we regard subgroups detected at adjacent time stamps as communities sharing the same opinion if the number of common members is above a certain threshold. Suppose we find a set of \( l \) opinion subgroups \( C_{t_k} = \{C_{t_k1}, C_{t_k2}, ..., C_{t_kl}\} \) at time stamp \( t_k \) in graph \( G_{t_k} \) and their predecessors \( C_{t_{k-1}} = \{C_{t_{k-1}1}, C_{t_{k-1}2}, ..., C_{t_{k-1}l'}\} \) at time stamp \( t_{k-1} \) in graph \( G_{t_{k-1}} \). To match adjacent subgroups together between \( C_{t_k} \) and its predecessors \( C_{t_{k-1}} \), the most widely-adopted method is to use Jaccard coefficient \([29]\). Given a current subgroup \( C_{tk,a} \) and a predecessor \( C_{tk-1,i} \), the Jaccard coefficient between the pair is calculated as:

\[
Jaccard(C_{tk,a}, C_{tk-1,i}) = \frac{|C_{tk,a} \cap C_{tk-1,i}|}{|C_{tk,a} \cup C_{tk-1,i}|} \tag{7.2}
\]

However, this classic definition can only be used for identifying state transition of two communities of similar size. During the evolution of opinion subgroups, we still need to consider other dynamic events where
community size changes significantly, such as forming, dissolving, expanding, contracting, splitting and merging. Although the subgroups involved in those events should be regarded as well-connected in our framework, the Jaccard coefficient may often report them to be low similarity communities. In order to deal with this problem, researchers [30, 31] have developed many definitions to handle these events separately. In our framework, we propose another definition to evaluate community similarity in all possible events:

$$sim(C_{t_ka}, C_{t_k-1i}) = \frac{|C_{t_ka} \cap C_{t_k-1i}|}{\min(|C_{t_ka}|, |C_{t_k-1i}|)}$$ (7.3)

If the similarity value exceeds the threshold $\theta \in [0, 1]$, this pair of communities will be matched. With this definition, we can effectively evaluate community similarity for all the state transition events with only one threshold parameter: When the size of two adjacent communities does not vary much, this definition keeps the property of the Jaccard coefficient; When dynamic evolution events occur where the community size changes significantly, this definition can match the communities together because of the high proportion of common nodes in the smaller community. Many previous works [32, 33] show that a reasonable similarity threshold in synthetic dynamic networks is between 0.3 and 0.4. In order to give a general model of user behavior in online newsgroups, we set the threshold $\theta$ to be 0.35, which can provide a reasonable compromise between community matching accuracy and identifying the optimal number of diffusion paths.

Furthermore, we also add the concept of gap interval [33] into our model when building the opinion diffusion paths. For a subgroup $C_{t_ka}$, we not only consider the subgroups $C_{t_k-1}$ detected at the prior time stamp $t_{k-1}$, but also include all the subgroups $\{C_{t_k-1}, C_{t_k-2}, C_{t_k-P}\}$ detected within the last $P$ steps into consideration. In online newsgroups, user attention and behavior are highly dynamic. Allowing a reasonable number of gap intervals helps us to detect more opinion diffusion paths. Considering the periodicity of online user behavior, we set gap interval as one day in our model, i.e. we allow possible connections between detected subgroups within 24 hours.

After connecting those opinion subgroups at different timestamps together, we get a new graph showing the evolution of user opinions along the whole timeline. In this graph, each node represents a detected opinion

104
7.3. Identification of Deliberation and Bias

subgroup. As the last step, we find all the connected components in this graph and define them as the opinion diffusion paths in this newsgroup. Figure 7.2 shows some examples of the opinion diffusion paths detected in Occupy Together newsgroup from Oct. 15, 2011 to Oct. 16, 2011. The number inside each node represents the size of the detected opinion subgroup, i.e. the number of people in that opinion subgroup. Each connected component in this graph represents a detected opinion diffusion path, which are shown in different colors.

Identifying User Characteristics. In our dynamic user-like graph model, a subgroup detected at one snapshot represents a group of users who share the same opinion during that time slot. And an opinion diffusion path aggregates all the users who share that specific opinion within the whole newsgroup. As we discussed in the introduction section, there are many different definitions of deliberation and bias in various domains. In this paper, we consider their definitions from the perspective of people’s dynamic behavior. To be specific, our hypothesis is: If a person appears in many different opinion diffusion paths, he may show deliberation in this newsgroup. On the other hand, if a very active user is found in only one opinion diffusion path, he may show bias towards that specific opinion in the newsgroup. To be specific, if a user appears in three or more different opinion diffusion paths, we will consider him to be a deliberative member. If a very active user who has made more than 30 comments in this newsgroup only shows up in one opinion diffusion path, we will consider him to be a biased member.

For the Occupy Wall Street newsgroup and the Occupy Together newsgroup, we run our dynamic graph model to identify user deliberation and bias. To get enough interaction content for each target user, we only collect information from people who have made more than 10 comments in the newsgroup. We find that, in the Occupy Together Facebook newsgroup, there are 787 people who made more than 10 comments. Based on our model, 201 of them are identified to be deliberative members and 26 of them are flagged as biased members. In the Occupy Wall Street newsgroup, we find a total of 2928 people with more than 10 comments. Among them, 916 people are found to be deliberative and 151 people are biased. Figure 7.3 shows some example comments of users who are identified to be deliberative or biased members in our dataset.
7. Mining User Deliberation and Bias in Online Newsgroups: A Dynamic View

![Example Comments of Deliberative Users and Biased Users](image)

7.3.3 Evaluation with Linguistic Features

Considering the huge size of our dataset, it is impossible for us to obtain the ground truth of online user characteristics by employing human workers to rate user interaction records manually. In this paper, we evaluate the effectiveness of our dynamic user-like graph model by comparing the linguistic features of user comments, which is a totally different view from the dynamic structure of a user-like graph.

We use the Linguistic Inquiry and Word Count (LIWC) tool [34] to study the linguistic features of user comments. LIWC is a popular natural language processing tool that calculates the matching frequency of words within each of 68 categories including linguistic dimensions and psychological aspects. The LIWC features we use in this evaluation process are based on previous work on Natural Language Processing. In [9, 35], researchers claim that deliberative behavior contains different lexical characteristics in the following five LIWC features: total word counts (WC), number of dictionary words (Dic), number of big words (Sixltr), words per sentence (WPS), and cognitive processes words (cogmech). In [16], Tae et al. use Amazon Mechanical Turk judgments to study biased sentences on American political blogs. Their result shows that the following LIWC features are indicators of bias: negative emotion (negemo), positive emotion (posemo), causation (cause), and anger (anger). In addition, they also include a list of 11 kill verbs\(^4\) as indicators of bias based on the study of Green and Resnik [36]. Based on their results, we use five of these LIWC features (WC, Dic, Sixltr, WPS, cogmech) to evaluate user deliberation and six features (negemo, posemo, cause, anger, anx, kill verbs) to evaluate user deliberation.

---

\(^4\)Those verbs are: kill, slaughter, assassinate, shoot, poison, strangle, smother, choke, drown, suffocate, and starve.
7.3. Identification of Deliberation and Bias

(b) Information Highlighting on SINCERE System.

Figure 7.4: Screenshots of SINCERE System.

bias. Table 7.2 shows some selected LIWC features we use in the evaluation process.

Table 7.2: Selected Linguistic Categories in LIWC.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Abbrev</th>
<th>Selected Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive processes</td>
<td>cogmech</td>
<td>cause, know, ought</td>
</tr>
<tr>
<td>Positive emotion</td>
<td>posemo</td>
<td>Love, nice, sweet</td>
</tr>
<tr>
<td>Negative emotion</td>
<td>negemo</td>
<td>Hurt, ugly, nasty</td>
</tr>
<tr>
<td>Anxiety</td>
<td>anx</td>
<td>Worried, fearful</td>
</tr>
<tr>
<td>Anger</td>
<td>anger</td>
<td>Hate, annoyed</td>
</tr>
</tbody>
</table>

Table 7.3: Evaluation of User Deliberation Identification.

<table>
<thead>
<tr>
<th>Newsgroup</th>
<th>Population</th>
<th>Graph</th>
<th>Linguistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccupyTogether</td>
<td>787</td>
<td>201</td>
<td>135 (67.2%)</td>
</tr>
<tr>
<td>OccupyWallSt</td>
<td>2928</td>
<td>916</td>
<td>662 (72.3%)</td>
</tr>
</tbody>
</table>

Table 7.4: Evaluation of User Bias Identification.

<table>
<thead>
<tr>
<th>Newsgroup</th>
<th>Population</th>
<th>Graph</th>
<th>Linguistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccupyTogether</td>
<td>787</td>
<td>26</td>
<td>15 (57.7%)</td>
</tr>
<tr>
<td>OccupyWallSt</td>
<td>2928</td>
<td>151</td>
<td>83 (55.0%)</td>
</tr>
</tbody>
</table>

For each of the detected deliberative users and biased users, we want to check if their linguistic features are significantly different ($p \leq 0.05$) from the average level of the whole population. This is a classic Z-test problem. Taking the evaluation of deliberative users as an example, we
use WC, Dic, Sixltr, WPS and cogmech as linguistic features. For each of these five LIWC features, we calculate the mean and standard deviation of the comments made by the whole population in the target newsgroup. Denote their mean values as \([\mu_01, \mu_02, \ldots, \mu_05]\) and their standard deviation as \([\sigma_01, \sigma_02, \ldots, \sigma_05]\). Then for each of the detected deliberative users, we denote the mean of his comments as \([\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_5]\) and the number of his comments as \(n\). Furthermore, we calculate the Z-score which represents the distance from the sample mean to the population mean in units of the standard error:

\[
Z_k = \frac{\sqrt{n}(\bar{\mu}_k - \mu_{0k})}{\sigma_{0k}} \quad k = [1, 2, \ldots, 5]
\]

Now we have a list of Z-scores \([Z_1, Z_2, \ldots, Z_5]\) for each of the detected deliberative users. As the last step, if the absolute value of any of these five Z-scores is larger than the predetermined significance threshold (\(|Z| \geq 1.96\)), we conclude that the comments made by this user show significant difference \((p \leq 0.05)\) on deliberative features in this newsgroup.

Table 7.3 and 7.4 show our evaluation results for user characteristics in the Occupy Together newsgroup and the Occupy Wall Street newsgroup. The item Population denotes the number of users who made more than 10 comments in the target newsgroup. The item Graph denotes the number of users who are detected to be deliberative/biased based on our dynamic graph model. The item Linguistic denotes the number and the percentage of the detected users who also show significant difference in our linguistic evaluation. From the results, we find that our dynamic graph model does a good job identifying user characteristics in the two online newsgroups. Furthermore, we notice that biased user identification does not perform as well as deliberative user identification. A possible reason is that biased users show different behavior patterns in various discussion topics. In topics where discussion is always intense, e.g., elections and religious issues, biased users are very likely to leave a large amount of comments. But in topics where discussion is less active, biased users may leave fewer comments in one opinion path. By defining a constant threshold for the number of comments in biased user identification, we may lose some accuracy. In summary, around 71.4% of identified deliberative users and 55.4% of identified biased users show linguistic features significantly different from the average level of the whole population. Additionally, because our dynamic graph model
Figures 7.5: The Relationship Between Different Discussion Circumstances and User Characteristics at Early Life-stages.

...does not require any training process or manually labeled ground truth, it is an efficient algorithm for user characteristics analysis in online social networks.

7.3.4 System Development

In order to evaluate the effectiveness of our dynamic graph model on social media, we also implement this algorithm in our SINCERE system as a real-time service. SINCERE stands for Social Interactive Networking and Conversation Entropy Ranking Engine. It is a diversified search engine based on user social informatics, which stores all the user interaction data of more than 1,800 Facebook public pages.

Figure 7.4(a) shows a screenshot of our user characteristics identification algorithm in SINCERE. By choosing a Facebook newsgroup from the drop-down menu, people can get real-time analysis results of the list of representative deliberative and biased users in that newsgroup. Both of their comments and Facebook links are also shown on that page. This function can help commercial institutions target people for political or entertainment advertisements.

Furthermore, we also use our algorithm to improve information presentation under each post of an online newsgroup. In Figure 7.4(b), we show comments made by deliberative (Blue) and biased (Green) users highlighted in different colors. This function is very useful for online users to obtain quick knowledge of different opinions and stances under a post.
Table 7.5: User Deliberation and Bias Prediction Results.

<table>
<thead>
<tr>
<th>Model</th>
<th>Deliberation Precision</th>
<th>Deliberation Recall</th>
<th>Deliberation F1</th>
<th>Bias Precision</th>
<th>Bias Recall</th>
<th>Bias F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Logistic Classifiers</td>
<td>65.2</td>
<td>67.2</td>
<td>66.2</td>
<td>56.4</td>
<td>71.0</td>
<td>62.9</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>63.4</td>
<td>85.6</td>
<td>72.8</td>
<td>60.0</td>
<td>82.3</td>
<td>69.4</td>
</tr>
</tbody>
</table>

7.4 Predict User Deliberation and Bias

The dynamic user-like graph model gives us a powerful tool to analyze the online characteristics of users. However, we must first collect all of the user interaction information before we can identify characteristics. Now we proceed to ask a deeper question: how can we predict deliberation and bias from user activities at early stages of interaction? In this section, we answer this question in two steps. First, we analyze the influence of different discussion circumstances at early stages on the formation of user online characteristics. Then, based on the analysis, we propose a supervised learning model to predict user deliberation and bias using content gathered at the early stages of user interaction.

7.4.1 The Influence of Early Discussion Context on the Formation of User Characteristics

Our hypothesis is as follows: if people confront disagreements in unfamiliar circumstances at an early stage, they may tend to show deliberation in later interactions. On the other hand, if users receive disapproval under familiar discussion circumstances, they are likely to become biased later. To verify this hypothesis using our online social network dataset, we first need to define some sociology concepts quantitatively:

- **User online lifetime.** We use the timestamps of feedback comments to define a user’s online lifetime. To be specific, we consider the timestamp of the user’s first received feedback comment as the starting point of his experience in the target newsgroup, and we consider him to have left the newsgroup after he has received the last feedback comment. In addition, we define the life-stage of a user as the percentage of time he has spent, out of his total lifetime, in the newsgroup. Obviously, a life-stage of 0% represents birth, which is the moment the user joins the discussion in the newsgroup, and a
life-stage of 100% represents death, which is the moment the user leaves the newsgroup.

- **Different feedback opinions.** In Facebook public newsgroups, there is no exact reply function during user interactions: people just make comments one after another in the time-line under each post. Therefore, we define the comments given within three hours after an individual makes a comment as his feedback replies. To classify these feedback comments into different opinions, we follow the same method used in our dynamic user-like graph model: we use the fast greedy modularity optimization algorithm to split the hourly graph snapshot into different opinion subgroups. Each user may get three kinds of feedback: the same opinion, different opinions, and opinion undetermined.

- **Stranger and Acquaintance.** We classify the people who give feedback into three types: stranger, acquaintance, and undetermined. We define a stranger as a person who has never given feedback to the target user before and an acquaintance as a person who has given at least three feedback comments to the target user before the current timestamp. Obviously, based on our definition, when a user first joins a newsgroup, every member there is a stranger to him. As the user starts to be involved in the discussion of the newsgroup, he will accumulate more and more acquaintances. One common concern about this definition is pre-existing friendships between users in online social networks. However, the dataset we use in this experiment is crawled from online newsgroups where group interaction is very different from that of private friendship groups. Members in online newsgroups have little social connection in real life since most interaction happens in an online environment [26]. Therefore, pre-existing friendships are a small concern for our definition of stranger and acquaintance.

In online environments, we assume that the discussion circumstances are mainly determined by the people who give feedback comments. Therefore, we define two context parameters to represent the extent of familiarity of the circumstances:
• **UnfamiliarPerAct** measures the percentage of feedback comments of different opinions received from strangers out of all the feedback comments this person received during the current life-stage.

• **FamiliarPerAct** measures the percentage of feedback comments of different opinions received from acquaintances out of all the feedback comments this person received during the current life-stage.

According to the identification results of our dynamic user-like graph model, we have 1117 deliberative users, 177 biased users and 2421 users with uncertain characteristics. Based on the definitions above, we plot the average context parameters for each category of users at different life-stages. Figure 7.5(a) shows the relationship between UnfamiliarPerAct and user characteristics at different life-stages. Figure 7.5(b) shows the relationship between FamiliarPerAct and user characteristics at different life-stages. Because we are only interested in the relationship at an early user life-stage, we only consider life stages before 30%. Comparing the two curves in each of the two figures, we find that the result supports our hypothesis pretty well: The users who are identified as deliberative experienced more conflicts in unfamiliar circumstances in their early stages and the users who are identified as biased experienced more conflicts in familiar circumstances at their early stages. Note that at the very first life-stage (5%) in both Figure 7.5(a) and Figure 7.5(b), the correlative relationship of the two curves does not fit our hypothesis well. The reason is that when a user joins a newsgroup, almost everybody is a stranger to him, which makes the effect of familiar circumstances quite weak. Considering this slow-start factor, we need to dismiss the effect of discussion circumstances at the very first stage in our following prediction model.

### 7.4.2 Supervised Prediction Model of User Deliberation and Bias

In this section, we leverage early discussion context factors to predict user deliberation and bias in online social networks.

**Features used for learning.** We define the following three features in our prediction model:

- **CommentsPerAct**: It measures the average number of received feedback replies after a user makes a comment.
7.4. Predict User Deliberation and Bias

• *EarlyUnfamiliarPerAct*: This feature is similar to the definition of *UnfamiliarPerAct*. It measures the percentage of feedback comments in different opinions given by strangers out of all the feedback comments this person received during the *early stage*. Considering the *slow-start* factor we mentioned in the previous section, we define the *early stage* as the time duration from the second week to the third month of the user’s lifetime in this newsgroup.

• *EarlyFamiliarPerAct*: This measures the percentage of feedback comments in different opinions given by acquaintances out of all the feedback comments this person received during the *early stage*.

**Experimental setup.** We use two supervised learning models to design our experiments: the binary logistic classifier (BLR) and the support vector machine (SVM). The ground-truth data in these models comes from the characteristics identification results of our dynamic graph model. Although binary logistic classifiers can only use linear predictor functions to build learning models, we can get explicit results about the regression coefficients in logistic classifiers, which can help us evaluate the effectiveness of each feature. In each of our experiments, we split 60% of the data for training and 40% for testing. Because our data is seriously imbalanced, especially for the sample of biased users, we downsample both experiments for deliberative and biased users.

**Experimental results.** AUC (Area Under the Curve) of the predicted user deliberation is 0.72, and AUC of the predicted user bias is 0.64. Table 7.5 also summarizes our prediction results for user deliberation and bias. We find that our framework does a good job of predicting both user deliberation and bias. Both supervised learning techniques, Binary Logistic Classifier (BLC) and Support Vector Machine (SVM), perform well. The SVM model gives an additional 6% absolute (10% relative) improvement in F1 scores for user deliberation and user bias.

To get a deeper insight into the effect of the discussion context features on our prediction results, we also examine the coefficients of the learned Binary Logistic Classifiers in Table 7.6. We find that the feature *EarlyUnfamiliarPerAct* shows statistical significance ($p < 0.05$) in the experiments of both user deliberation and bias. Furthermore, the sign of its coefficient in both experiments fit our conclusion in the previous section perfectly: the more conflicting comments people receive in unfamiliar circumstances
at their early stage, the more likely he will become deliberative in the future; meanwhile, the conflicting experiences people have in unfamiliar circumstances at their early stage will also reduce the probability of him becoming a biased user in the newsgroup. However, we did not see a clear effect of EarlyFamiliarPerAct on the BLC model. There may be many possible reasons: First, in contrast to the classification of strangers, everybody has their own understanding of what makes familiar discussion circumstances. Some people may regard a person who has talked with him more than two times online as an acquaintance, while others may require ten times. By simply defining the acquaintance threshold as three, we may fail to evaluate some users’ discussion experiences. Second, the feeling of familiar discussion circumstances may decay as time goes on. Instead of keeping a static table of acquaintances for each user, it may be better to set a sliding window of acquaintances, so that we can recognize familiar circumstances dynamically.

7.5 Conclusion and Discussion

In this work, we propose a framework for identifying characteristics of online users and understanding the formation of user deliberation and bias in online newsgroups. First, we propose a dynamic user-like graph model for recognizing user deliberation and bias automatically. Then, after applying this model to two large online newsgroups, we analyze the influence of early discussion context on the formation of user characteristics. Our conclusion is that user deliberation and bias are a product of situations, not simply their dispositions: Confronting disagreement in unfamiliar circumstances promotes more consideration of different opinions, while recurring conflict in familiar circumstances evoke close-minded behavior and bias. Furthermore, based on this observation, we also build a supervised learning model to predict user deliberation and bias using information from the early life-stages of newsgroup participation. Our results show that having only the
first three months of users’ interaction data generates an F1 accuracy level of around 70% in predicting user deliberation and bias in online newsgroups.

In politics, the influence of contextual factors on people’s behavior is widely discussed. Researchers find that people pursue two different kinds of citizenship, deliberative and partisan, under different circumstances and they explain that this is a result of emotion [37, 38]. Michael et al. [6] claims that it is emotion that conveys information about the environment and guides the kind of citizen behavior people pursue. When people are in novel circumstances, they may feel anxiety, making them more likely to engage in deliberative mechanisms in order to handle uncertainty. When people find themselves involved in conflicts under familiar circumstances, they may feel aversion, which pushes them to rely on previously learned solutions and become close-minded to alternatives. Although emotions of anxiety and aversion can explain the formation of deliberation and bias more directly, they can hardly be applied to large-scale data analysis in online social networks, because it is very challenging to differentiate user anxiety and aversion accurately by natural language processing methods. By leveraging user behavior information, our framework shows good performance for both identification and prediction of user deliberation and bias.

For future work, we plan to incorporate the interaction of different newsgroups into our analysis framework. In this paper, we only consider the influence of user interaction within the same newsgroup. However, it is highly likely that online users participate in many different online newsgroups at the same time. Because of the dynamics of online user membership, an opinion that seems to disappear in one newsgroup may actually begin to dominate the discussion in another newsgroup. By incorporating cross-group interactions, we can get a clearer picture about opinion diffusion and user characteristic formation in social media.

7.6 Acknowledgments

We would like to thank Shijia Che, Haifeng Zhao and Ran Cao for their insightful comments on this paper. This work is partially supported by the Cyber Security Research Alliance of United State Army Research Laboratory.
7.7 References


[9] X. Xu, T. Murray, B. P. Woolf, and D. A. Smith. “Mining social deliberation in online communication: If you were me and i were you”. In: EDM. 2013.


Crawling Online Social Networks

Fredrik Erlandsson, Roozbeh Nia, Martin Boldt, Henric John-son, S. Felix Wu

Abstract

Researchers put in tremendous amount of time and effort in order to crawl the information from online social networks. With the variety and the vast amount of information shared on online social networks today, different crawlers have been designed to capture several types of information. We have developed a novel crawler called SINCE. This crawler differs significantly from other existing crawlers in terms of efficiency and crawling depth. We are getting all interactions related to every single post. In addition, are we able to understand interaction dynamics, enabling support for making informed decisions on what content to re-crawl in order to get the most recent snapshot of interactions. Finally we evaluate our crawler against other existing crawlers in terms of completeness and efficiency. Over the last years we have crawled public communities on Facebook, resulting in over 500 million unique Facebook users, 50 million posts, 500 million comments and over 6 billion likes.

8.1 Introduction

Recently, online social networks, OSNs, have gained significant popularity and are among the most popular ways to use the Internet. There have been efforts to make the social informatics on, for instance, Facebook available for applications (e.g., [1, 2]). Additionally, researchers and developers have become more interested in using the social interaction networks, SINs [3], to further enhance and personalize their services [4]. OSNs are also redefining roles within the publishing industry, allowing publishers and authors to reach and engage with readers directly [5]. However, SINs are not directly available today through the current APIs provided by most
OSNs. Applications using SINs would therefore spend a lot of time trying to gather the data needed to create the SINs for their services. Therefore, our research problem is how we can design a crawler that makes social interactions in OSNs accessible. To the best of our knowledge it exists no crawler today with the capabilities of crawling all interactions in a timely manner. This also refers to the problem of how, when and what to crawl from OSNs in a structured way, which is the focus of this paper.

Researchers have studied and explored the information flow on online public communication boards. However, these communities are usually significantly smaller than the communities we find on Facebook. For instance, communication boards of Open Source Software are limited to a few thousand people out of which only a few hundred of them are active members [6, 7]. Such networks are considerably smaller than Facebook communities. The number of members of Facebook groups that we have crawled range from a few thousand to tens of millions of people.

The nature of OSNs and the amount of information available makes the problem of what to crawl interesting. To narrow down the scope of the proposed research, we are focusing on the interactions in OSNs. By doing this, we noticed a gap and segregation between the content and the social graph. There have been efforts to make social informatics on Facebook available for applications; enabling social computing applications can simply worry about the computation part and not the social informatics. Performance and incompleteness issues of existing crawlers were the main reason for us to start developing SINCE - SIN Crawler Engine, that serves as a bridge between the content and the social graph in the online world, by not only providing which users have interacted with each other but around exactly which content these interactions have occurred. For readability in this paper, a page refers to a single community and a post refers to anything shared on a page. SINCE is crawling open pages on Facebook and can make informed predictions on how social interactions take place. It leverages this to prioritize what content to crawl next or decide which contents need to be crawled again, re-crawled.

SINCE makes the social information around posts easily and thoroughly accessible, which is necessary in order to create SINs and build applications such as Friend Suggestion [8] which are dependent on the complete set of interactions between users.
8.2. Related Work

Although SINCE only crawls data from public pages and domains, as discussed in [9] we threat the crawled data with high respect to the users. For instance, we do not draw direct connections between the user id and the profile page of the user. This also means that we do not try to access information from users’ wall so we never access or make available any data that was published with configured privacy settings. The content of public pages are by definition open information and we have discussed this with Facebook representatives and they do not have any concern with our data.

The rest of the paper is organized as follows. We start with a comprehensive discussion of related work in Section 8.2 and from that we can validate the need and originality of our work. In Section 8.3 we discuss our requirements and various challenges our crawler have been facing, including resource allocation and capabilities to predict when posts needs re-crawling. In Section 8.4 we describe the design decisions taken while developing our crawler, including the first published solution on how to crawl shares using the Facebook API. We also present our created API that makes the data gathered by SINCE available for other researchers and developers. The methods of prioritizing the crawling queue is described in Section 8.5. We finish the paper (Section 8.6) with an evaluation and a comparison between SINCE and other crawlers. We also show that the location of the crawler is important together with statistics of measured crawling time for 2.5 million posts. Finally, the paper is concluded in Section 8.7 and future work in Section 5.8.

8.2 Related Work

Despite the huge number of social network publications, few have been dedicated to the data collection process. Chau et al. [10] briefly describe using a parallel crawler running breadth-first search, BFS, to crawl eBay profiles quickly. The study conducted by Mislove et al. [11] is, to the best of our knowledge, the first extensive OSN crawling study published. From four popular OSNs, Flickr, Youtube, LiveJournal, and Orkut, 11.3 M users and 328 M links are collected. Mislove et al. confirms known properties of OSNs, such as a power-law degree distribution, a densely connected core, strongly correlated in-degree and out-degree, and small average path length.
Gjoka et al. [12] are proposing two new unbiased strategies: Metropolis-Hasting random walk (MHRW) and a re-weighted random walk (RWRW). Where Catanese et al. [13] describes the detailed implementation of a social network crawler. It used the BFS and uniform sampling as the crawling strategies to run the crawler on Facebook, and then compared the two strategies.

Most studies are based on subgraphs, thus it is important to know how similar the sampled subgraphs and the original graphs are. Leskovec et al. [14] evaluate many sampling algorithms such as random node, random edge, and random jump. The datasets used by Leskovec et al. [14] are citation networks, autonomous systems, the arXiv affiliation network, and the network of trust on opinions.com, the largest of which consists of 75k nodes and 500k edges.

Ahn et al. [15] obtain the complete network of a large South Korean OSN site named CyWorld directly from its operators. They evaluate the snowball sampling method (which is in fact breadth-first search) on this 12 M node and 190 M edge graph. Their results indicate that a small portion (<1%) of the original network sampled in snowball fashion approximates some network properties well, such as degree distribution and degree correlation, while accurate estimation of clustering coefficient is hard even with 2% sampling.

Gjoka et al. [16] propose a sampling method to select nodes uniformly without knowledge of the entire network, and use this method on a large sample (1M nodes) of the Facebook graph. The link privacy problem raised by Korolova et al. [17] concerns how an attacker discovers the social graph. The goal of the attacker is to maximize the number of nodes/links it can discover given the number of users it bribes (crawls). Several attacks evaluated actually correspond to node selection algorithms for crawling, such as BFS and greedy attacks. The same problem is considered by Bonneau et al. [18] who took a survey of several approaches for obtaining large amounts of personal data from Facebook, including public listings, false profiles, profile compromise, phishing attacks, malicious applications and the Facebook Query Language. The research dataset in [11, 19] was mined through public listing in [18].
8.3 Requirements and challenges

Our crawler highly depends on Facebook’s API, and therefore, bugs in Facebook’s API will cause problems that we have no control over. Also, resource limitations have forced us to be picky about which communities to crawl. Given enough resources, our crawler can be modified to automatically crawl all public communities on Facebook and other OSNs given an initial set of seeds.

8.3.1 Requirements

SINCE, from a high level, takes the identifier of a Facebook community as input and outputs a stream of documents. In addition to capturing the response of API requests, our crawler has to satisfy the following requirements:

Coverage

It is important and desirable to be able to crawl each and every post thoroughly and completely. However, if resources do not allow this, it is more desirable to get all the data from a limited set of posts, depth, rather than less data from a larger set of posts, breadth. Example of an application that leverages breath crawling could be a Dynamic News Feed, where users are not only bound to see the posts shared by their immediate friends and pages they have liked but can quickly see emerging topics of interest for the user, which will create a more dynamic news feed.

Since we are dependent on depth, we are aiming on applications built for leveraging social interaction networks, such as Friend Suggestion [8]. Hence SINCE is implemented as a deep interaction crawler.

Real-time

The information and interactions on Facebook public communities is time-sensitive. When crawling a post as deeply as SINCE does, the crawling time is an important factor. Although many of the observed posts are quite small with a short crawling time, just a few seconds, we also have big posts with crawling time up to a few hours. Since the interactions on posts and the social interaction networks around these posts are constantly changing,
we need means to decide if a re-crawl is needed or not, which is described in Section 8.5.

Important questions that arise due to the nature of how the interactions around posts evolve are “Which posts do we have to re-crawl to get the most updated information?” and “When would be the best time to re-crawl these posts?”.

Scale

As of today there are over a billion users and millions of public communities on Facebook [20]. There are over 2.7 billion likes and comments posted on Facebook on a daily basis as of February 1st 2012 [21]. A crawler must have capabilities to scale to become more and more efficient as content grows.

Data Quality

The crawler should output reliable and uncorrupted data. Therefore, it needs to be able to detect failures in Facebook’s current API and be able to restart from exactly where it stops when a failure occurs.

8.3.2 Re-crawling

Unlike traditional blogs or websites where only the administrators are able to post updates to their websites, OSNs are constantly receiving new posts from hundreds of millions of users around the clock. Therefore, traditional web crawlers and their algorithms that identify re-crawling time would not satisfy our requirements. OSN users are more active with ongoing
interactions, collaborating and posting new content. Therefore, not only do we need to crawl everything efficiently in a given amount of time but we have to detect whether we would need to re-crawl what we have already crawled in order to get additional content. This issue most often arises for popular posts that are constantly receiving new interactions such as likes, comments, and shares from users; hence, crawling such posts are extremely expensive and it is crucial to make an informed decision whether and when we should issue a re-crawl.

The data we have gathered has been analyzed by Wang [22] suggests that most interactions on posts happen within three hours after the post was made. Table 8.1, shows the number of comments on popular posts divided between four different time intervals after the post was initially made. In (a) we see that more than 70% of first comments take place within the first 30 minutes after the post was initially created. In addition, if the post has more than 20 comments, then 98% of the posts have the first comment within the first 30 minutes, and merely 2% of the posts get the first comment after 30 minutes. (b) shows that if the post has more than 20 comments, only 2% of the posts will get the fifth comment over two hours from when the post was initially created. Another point we can take from this table is that 95% of posts that get their tenth comment later than three hours from when it was first initially shared, will get fewer than 15 comments total. (c) shows that 92% of posts that get their tenth comment later than three hours from when it was first initially shared, will get less than 20 comments total. (d) shows that 95% of posts that get there 20th comment later than three hours from when it was first initially shared, will get less than 40 comments total. This information is used and discussed further in Section 8.5, and we are using it in order to decide whether we will need to re-crawl a post that have already crawled and further, when would be the best time to re-crawl the post in order to get the complete view of each post.

Another approach which helps to decide whether we should re-crawl a page or post is by looking at how the SIN forms around a particular page or post and how the members of SIN have interacted with the posted items before. Psychology studies show that people tend to interact with posts that they personally agree with, and that most people would not initiate an opposing point of view [23]. Although, once an opposing point of view has been posted, the rest of the people are very likely to follow.
Based on the ideas described above, SINCE not only is able to efficiently
detect which posts need to be re-crawled, but also decides when would be
the best time to re-crawl such posts to capture the maximum amount of
interactions and reduce the probability of needing to re-crawl the post in
the future.

8.4 A platform to make interactions accessible

SINCE is able to crawl all public pages and groups on Facebook, given just
basic privileges like a normal Facebook user. Even private communities can
be crawled given that the user id of the crawler has access. Furthermore, it
is easy to modify our tool in order to extract information from other OSNs.

8.4.1 Design

SINCE is designed to perform crawling in two stages. *Stage one* uses the
Facebook’s unique identifier of a public community (page or a group) to
find the id of all posts, messages, photos, and links posted on the given
community by admins and members. This is a straightforward process that
has to consider Facebook’s pagination [24] of API requests as discussed
below. A stage one crawl will simply access the feed connection on the
community-id of the community we are interested in and continue to read
the next page until it reaches the last page. This will give us a complete
list of posts in a particular community.

*Stage two* is a bit more advanced. Since we are interested in all social
interactions for each post, we have to make the following requests for each
post gathered at stage one. We first gather the *post* itself, this post contains
basic information like author, type, the message, and in applicable cases,
links to the posted photo, link, video. In the first request we also get a
preview of the posted comments and who have liked the post but this is
not a complete view. In order to get all *likes* we iterate through the *like
handle*. To get all comments on a post we have to iterate through the
post’s *comment handle* and since each comment can have *likes*, we have
to iterate through the *like handle* for each comment as well. As discussed
in Section 8.4.2, Facebook does not provide a direct API to the *shared by
information*. However, we have found a work-around for this problem and
that will add an additional call to the graph. The methods described for
*stage two* crawling means that for posts with a lot of interactions we have
to make multiple requests to the graph. For instance, we have crawled posts with hundreds or thousands of comments each with a few likes, where we have to make a request for each comment to get its likes, resulting in crawling times of several hours for one single post.

**Pagination**

As mentioned before, Facebook has a limit on how many entities to be returned from calls to their graph-API [24]. This is by default set to 25 for Facebook’s internal calls. We have modified this, so all calls we make to Facebook requests 200 entities. The trade off is that, the higher this limit is configured, the more likely a failure might occur on Facebook’s servers; since, every request has a short time limit to be completed. For each failure we will need to re-crawl the number of posts equal to the limit that we have set. We have found 200 to be the ideal limit for our use cases with concern of the issues described above.

**Facebook restrictions**

Our crawler is built as a distributed system as discussed by Chau et al. [10]. This satisfies our demand of high crawling rate and works as a work around to the fact that Facebook only allows 600 requests per 600 seconds. We have one controller that is keeping track on current status and what data (in our case which page or post) to crawl next. The controller supports multiple agents. Figure 8.1 shows a basic sketch of how the controller and the crawling agents are connected. Each agent runs independently and can have its own Facebook application id and user id. In our current version we have seen that it is possible to reuse each application id for ten to fifteen agents (based on the physical location of the agent as discussed in Section 8.6.2). Running more agents with the same application id will hit Facebook’s 600/600 limits and force the agents’ with the same application id to wait up to 600 seconds before they can continue crawling.

**Efficiency measures**

As described before, our crawler is designed as a distributed system. The controller is keeping track of interesting pages and the corresponding posts with support for n-agents to do the actual crawling. At most we have had just over one hundred active agents. The controller holds and prioritizes a
8. **Crawling Online Social Networks**

![Diagram](image)

Figure 8.1: *Our distributed crawling mechanism.*

queue of which *pages* and *posts* to crawl, when we see that one *page* has many interesting interactions we can point the agents to crawl that page. As of today, we have up to a few hundred agents that are able to grab community and post ids from the controller and crawl the context based SINs. Given enough active crawling agents, our tool can easily adapt to crawl every public community on every OSN in a timely manner.

### 8.4.2 Crawling Shares

One of the shortcomings in the current Facebook API is the lack of ability to crawl shares. Share is a term used by Facebook to show posts that have been shared by a user other than the initial poster. A user can share a post to their own, their friends, or any community that they have access to. The problem to crawl shares has been reported as a bug [25], but to the best of our knowledge there are no solutions to this issue. In the documentation Facebook has provided for developers, the shares are not covered. Results from API calls like `http://graph.facebook.com/-<community-id>_<post-id>` only returns the number of shares. Opposed to similar calls for likes and comments where we get the full list of who has taken which action and when. It is interesting in terms of weighting different posts among each other to see how many shares, likes and comments each post have, where we consider the shares to have the highest impact of importance for a post. In our crawled dataset we have seen that a user is much more prone on doing a simple *like* or perhaps leaving a *comment* on a post than to re-sharing the post among its social graph. Not only is the number of shares important, but to use the crawled data to build ego-networks it is also interesting to see who have shared a post.
Our solution to this is using the fact that most of the items on Facebook have a globally unique id. When looking at the standard method for crawling a post we always combine the page-id with the post-id and separate them with an underscore. For instance, http://graph.facebook.com/123_456 where 123 is the page-id and 456 is the post-id for the post. By making a request to post-id directly and then adding the keyword sharedposts is it possible to see who have shared a particular story. In fact, to crawl the users who have shared a story the request have to look like this: http://graph.facebook.com/456/sharedposts. This request will return information of the users who have shared the post with time stamp and which audience they have shared it to.

8.4.3 Application Programming Interface

Together with SINCE we also have implemented a social aware kernel that is able to compute the social interactions, and make the produced data available through an API. Enabling developers and researchers to implement applications that can access these interactions and our crawled data. In addition, we compute and produce social interaction networks on the go around different content shared on OSNs. This functionality allows developers to only worry about how they like to leverage such social informatics to improve their services, instead of spending tremendous amount of time gathering the raw data and producing the networks themselves. Furthermore, we allow developers to directly access and even modify our database by adding new fields and creating new tables in order to be able to store their computed data so that they and other developers can benefit from their computations in the future. This functionality is not available through any API provided by OSNs as of today; therefore, developers and researchers are either not able to request social interactions or have to implement a crawler and compute this.

The functionalities provided by our API include retrieving more information with fewer requests compared to other APIs out there. In addition, we do an extensive amount of computing in order to create the SINs [3] based on the specified types of interactions in the request and providing the interaction networks in the returned response. Finally, the feature that separates our work from every other API is the extended functionality feature. We allow developers and applications to extend our object oriented
8. **Crawling Online Social Networks**

Table 8.2: *Penalty model used by the crawler to prioritize the queue.*

<table>
<thead>
<tr>
<th>Status</th>
<th>Probability (of new content)</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new (uncrawled)</td>
<td>high</td>
<td>5</td>
</tr>
<tr>
<td>$\Delta t &gt; interval$</td>
<td>medium</td>
<td>3</td>
</tr>
<tr>
<td>$\Delta t &lt; interval$</td>
<td>low</td>
<td>1</td>
</tr>
<tr>
<td>Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new (uncrawled)</td>
<td>high</td>
<td>4</td>
</tr>
<tr>
<td>$(\Delta t &lt; 30 \text{min}) &amp; (\text{comments} &gt; 20)$</td>
<td>medium</td>
<td>3</td>
</tr>
<tr>
<td>$(\Delta t &lt; 3 \text{h}) &amp; (\text{comments} &gt; 40)$</td>
<td>medium</td>
<td>2</td>
</tr>
<tr>
<td>$(\Delta t &lt; 3 \text{h}) &amp; (\text{comments} &lt; 40)$</td>
<td>low</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta t &gt; 2 \text{h}$</td>
<td>low</td>
<td>0.5</td>
</tr>
<tr>
<td>$\Delta t &gt; 2 \text{months}$</td>
<td>low</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\text{current time} - \text{update time} \Rightarrow \Delta t \text{ (time since last crawl)}
\]

\[
\frac{\text{last post time} - \text{first post time}}{\text{number of posts}} \Rightarrow \text{interval} \text{ (post intervals)}
\]

designed system to add their own functions/modules to our code and later call these functions on our server.

### 8.5 Prioritization of the crawling queue

As stated before, we have one controller keeping track of the current progress and status of the agents. In addition, the controller does not do any actual crawling but is managing a queue of posts and pages that needs crawling. This queue was initially built as a simple FIFO queue, first in first processed, with the addition of keeping track of failures and timeouts from the agents. If a post is sent to an agent and the controller does not receive a response after 4 hours, the controller considers this post to have failed during crawling and will move it to the top of the queue. Although this is a simple and quite efficient method to get the system up and running, it is neither very intelligent nor efficient in terms of getting the most interesting posts. Therefore our controller has been updated to make use of the findings discussed in Section 8.3.2 in order to prioritize the queue in a more intelligent way. The controller also uses the penalty model shown in Table 8.2.

As seen in Table 8.1, if a post is crawled less than 30 minutes from the creation and it has more than 20 comments we can expect this post to expand more. Therefore, the crawler keeps track of this post and re-crawls it again. Also, the crawler knows the last time a *stage one* crawl was performed of the communities (groups and pages). This information is used to know when to expect new content and issue a re-crawl of the community.
Posts older than a few months are not prone to have many new interactions, based on the findings in Table 8.1, so these posts typically does not need to be re-crawled.

8.6 Evaluation

Many of traditional metrics used for crawlers such as speed, bandwidth utilization and staleness are not usable when crawling interactions on Facebook. The reason is that the major issues are related to restrictions introduced by Facebook to limit application’s ability to extract too much data as fast as described in Section 8.4.1. However as discussed in the same Section we have taken measures to get an efficient crawler by using multiple application ids and user ids to maneuver around these restrictions.

As seen in Table 8.3, our crawler is the only one with full coverage, looking at all interactions around a specific post. It is also the one with the largest dataset as our dataset is much broader and covers all interactions on crawled content. Although our crawler limits in automatization of crawling, the current implementation where we are adding interesting pages manually have given us a few advantages over automatic crawling strategies; We can decide if a page seems to be of interest and if the interactions on the page can be of value for other research applications. To our knowledge, there exists no other crawler with the same capabilities and with full coverage of pages and posts.

8.6.1 Crawling time

Our tool is considerably faster and at the same time it does a more thorough job than other crawling tools crawling OSNs using HTML parsers. This is due to the following reasons. Firstly, we use Facebook’s API and get all the content in JSON format meaning that we do not have to worry about parsing text or HTML, which by itself could be a complicated process. Secondly, other crawlers that rely on HTML pages will miss a lot of information since Facebook only makes a limited part of what has been shared available through HTML, i.e. what they provide within their API contains a lot more data that is simply not visible to the user. Finally, upon a failure, HTML crawlers will need to restart the process from the very beginning, while our tool will pick up from where it last successfully crawled the OSN.
| Author | Open Source Coverage | Dataset(s) | Crawling (Algorithm) | Performance | Parallelism | Crawling
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Erlandsson et al.</td>
<td>yes</td>
<td>full</td>
<td>-</td>
<td>yes</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mislove et al.</td>
<td>partial (just users)</td>
<td>yes</td>
<td>Flickr, LiveJournal, Orkut, YouTube</td>
<td>yes</td>
<td>Manual</td>
<td>yes</td>
</tr>
<tr>
<td>Gjoka et al.</td>
<td>no</td>
<td>partial</td>
<td>Twitter</td>
<td>yes</td>
<td>MHRW</td>
<td>yes</td>
</tr>
<tr>
<td>Catanese et al.</td>
<td>no</td>
<td>partial</td>
<td>Facebook</td>
<td>yes</td>
<td>BFS</td>
<td>yes</td>
</tr>
<tr>
<td>Haralabopoulos &amp; Anagnostopoulos</td>
<td>partial</td>
<td>yes</td>
<td>Facebook</td>
<td>yes</td>
<td>BFS</td>
<td>yes</td>
</tr>
<tr>
<td>S. Ye, J. Lang &amp; F. Wu</td>
<td>no</td>
<td>partial</td>
<td>Flickr, LiveJournal, Orkut, YouTube</td>
<td>yes</td>
<td>Greedy, Lottery, BFS</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 8.3: Comparison of different crawlers and crawling strategies.
8.6. Evaluation

(a) Crawling time distribution for 2.5 million posts
(b) Crawling time per location for 2.5 million posts

Figure 8.2: (a) shows the crawling time distribution over 2.5 million posts. As we can see the average time is quite short but there is a spread over 10000 seconds, or roughly 3 hours, for posts with a lot of interactions. (b) shows the distribution of crawling time based on the agents’ location.

We have had agents spread over the world, Figure 8.2a shows the distribution of crawling time over 2.5 million posts. The figure shows that the median crawling time is 1.86 seconds, but there are also posts that require a lot longer crawling time. We have recorded crawling times of over 10000 seconds for posts that have hundreds of thousands different types of interactions. The box plot on the top illustrates the median, and the distribution of the crawling time.

8.6.2 Crawling location

Our agents are spread over the world (currently Sweden, Taiwan and the USA). What we have observed here is that the closer to Facebook’s data center in California the agent runs, the faster response it will get. While this could be considered to be an obvious fact; the closer you are to the server you are communicating with the faster the response will be. However we still think it is interesting to show that it have a significant impact on crawling rate based on where the crawler runs. The network latency must be considered when building large-scale data mining tools.

Figure 8.2b shows the distribution of crawling time based on the agents’ location using the same dataset as for Figure 8.2a. It confirms our assump-
8. Crawling Online Social Networks

...tion that the closer the crawling agents are to the Facebook’s data centers the shorter crawling time can be achieved. The long tail is created since some of the crawled post can be very large, compared to the distribution of our crawled posts. The Amazon plot is utilizing Amazon’s EC2 cloud service and its shape differs because it only ran for a few days, where the rest of our agents ran for a few weeks, and during that time the posts served were considerably larger than the rest of distribution of our crawled posts.

8.7 Conclusion

We have shown means for building an extensive tool to crawl public communities in OSNs. Our distributed crawler satisfies the requirements described in Section 8.3.1 and is capable of retrieving a complete set of non-corrupted data, including all the content shared and all the user interactions around them, from public pages on Facebook. We have given a description of how to design a mining tool for OSNs that can be used in social interactions.

Our findings show that to get a full view of social interactions on networks like Facebook it requires a considerable amount of resources. To increase performance of our crawler we have built a distributed system with one controller, responsible for keeping track of the current status, and multiple agents, making the actual crawling.

Our crawler is capable of handling failures and errors that might happen on the OSN’s API level. In addition we have taken steps to overcome the 600 API requests per 600 seconds that Facebook has by letting our crawling agents use different application keys.

8.8 Future work

There is a lot of information being shared on different OSNs, but due to resource limitations we can not capture everything that is happening on Facebook. It is fair to say that Facebook is known to be the largest OSN with the most number of active users. It would be interesting to modify our crawler to capture the interactions on other OSNs, such as Google+, Twitter, and Youtube. The social graph introduced by these services differ from each other and it would be interesting to study and compare the effect
of that on the communications and interactions that take place on different OSNs.

Our prioritization scheme discussed in Section 8.5 poses the risk of having “blind” interactions, interactions that occur very late in the post’s lifetime. Therefore we would like to further investigate what percentage of the interactions we potentially are missing by not re-crawling old posts.

Another limitation that we have due to the limited hardware access is that we cannot capture everything that is happening on OSNs, we have to pick the most interesting groups to us and prioritize which communities to crawl. Given enough hardware, our crawler can easily be modified to automatically detect new communities or pages based on their interactions with the communities that we start crawling from (i.e. our initial seed).

The prioritization discussed in Section 8.5 address the issue on when to crawl a post. It would be interesting to use meta-data gathered in stage one for checking if a post could be considered to not be interesting and thereby ignoring it. Here the traditional data mining term of interestingness should be evaluated on social media. A study to map interestingness to social media data in order to prioritize data would be a good start to evaluate data and prioritize if some posts can be ignored.

8.9 References


8. CRAWLING ONLINE SOCIAL NETWORKS


Predicting User Participation in Social Media

Fredrik Erlandsson, Anton Borg, Henric Johnson, and Piotr Bródka

Abstract

Online social networking services like Facebook provides a popular way for users to participate in different communication groups and discuss relevant topics with each other. While users tend to have an impact on each other, it is important to better understand and analyze users behavior in specific online groups. For social networking sites it is of interest to know if a topic will be interesting for users or not. Therefore, this study examines the prediction of user participation in online social networks discussions, in which we argue that it is possible to predict user participation in a public group using common machine learning techniques. We are predicting user participation based on association rules built with respect to user activeness of current posts. In total, we have crawled and extracted 2,443 active users interacting on 610 posts with over 14,117 comments on Facebook. The results show that the proposed approach has a high level of accuracy and the systematic study clearly depicts the possibility to predict user participation in social networking sites.

9.1 Introduction

Online social networks are a large part of our society. Just Facebook alone attracts 1.3 billion users\(^1\) with 640 million minutes spent each month. Facebook had a total revenue of $12,466 M in 2014\(^1\). Consequently, discovering trending topics or influential users is of interest for many researchers, e.g. for marketing [1]. Several studies have tried to identify user influence,

\(^1\)http://www.statisticbrain.com/facebook-statistics/
however most have used page rank [2] or centrality [3, 4] based approaches to identify influential users.

In this article we argue that users, on Facebook groups, are following each other and that it is possible to detect influential users. E.g. if user A, B, C and D share common interests, the chance is that if A, B, and C already have commented on a topic, D also will comment on it. Therefore, this paper relates to how users perform actions (e.g. comments or likes) on posts in Facebook pages. In addition, we use association learning to discover relationships between variables, or in our case users, in the dataset [5]. Given a list of posts from a specific domain we extract users actions such as comments and likes. Using association rule learning on the data, we argue that it is possible to predict if a particular user will or will not participate on a post discussion based on the other users activity.

For evaluation, a systematic study is conducted, which include building association rules that can be used to predict if a specific user will be active in a particular post. The prediction is done based on the activeness of users within current posts. Moreover, the scope of the paper is limited to user interactions on a subset of Facebook users on posts with a similar topic.

The paper is organized as follows: In Sect. 9.2 related work is discussed. Section 9.3 and Sect. 9.5 presents the data and the methodology. Association rule learning and the evaluation metrics are discussed in Sect. 9.4. Finally, the results are presented in Sect. 9.6 and discussed in Sect. 9.7.

9.2 Related work

Online social networks and social media analysis are one of the hottest areas of research in modern network science. Like in many different areas, scientists struggle to predict the future of online social network. The main focus in social network area is on link prediction [6] but different teams around the world work also on: (i) popularity prediction in social media based on comment mining [7], (ii) personality prediction for micro blog Users [8], (iii) churn prediction and its influence on the network [9, 10], (iv) community evolution prediction [11, 12], (v) using social media to predict real-world outcomes [13], (vi) predicting information cascade on social media [14], (vii) users features prediction using relational learning [15, 16], (viii) predicting patterns of diffusion processes in social network [17], (ix)
predicting friendship intensity [18, 19], (x) affiliation recommendations[20, 21], and many others.

Association rule mining has been previously used in social network and social media analysis. In [22], the authors explores the association rule between a course and gender in the Facebook 100 university dataset. This was performed to discover the influence of gender in studying a specific course. Yu et al. [23] introduces the scheme for association rule mining of personal hobbies in social networks, while Schmitz et al. [24] tackle the problem of mining association rules in folksonomies and try to find out how association rule mining can be applied to analyze and structure folksonomies.

However, while online social network analysis is popular, there is according to our review a lack of research on using association rules for predicting user participation in online social media discussions.

9.3 Data model

The data used in this study has been obtained from the crawler described by Erlandsson et al. [25]. This crawler gathers complete posts from Facebook. In this context, the term complete stands for posts that contains all likes and comments. In addition, if a post is crawled, the dataset contains all likes, comments and interacting users up to the crawling time. Our current dataset, captured from public pages and groups on Facebook, consists of over 56 million posts, 560 million comments and 7.3 billion likes made by 820 million Facebook users. The crawled data is parsed and available from a SQL database, structured as described in [26], making all fields needed for our task available. In this study, we assume that the investigated posts will not get any new comments. We simplify the dynamics of social media by saying that the posts we are investigated are “dead” when the data was collected, in which the term of dead posts refers to posts that no longer attracts attention or new comments or likes.

We are limiting this study to only investigate a subset of groups available by the crawler. From these groups we exclude posts with less than 20 comments as these posts are considered to be of too low value and do not hold enough information.
9.3.1 Data selection

To perform prediction of user interactions, we have selected the page OccupyTogether. This page was selected based on the following properties: it is active, it has a high number of users (~300k), it has a reasonable high number of active users (~30,000 users with more than one comment) and it is political with a bias user group (most of the users are positive to the Occupy movement). From this page, only users that have made more than five comments are investigated. This ensures that the selected users are or have been fairly active in the community. The resulting dataset consists of 2,443 users interacting on 610 posts totaling in 14,117 comments.

9.4 Association Rules

As stated in Sect. 9.1, we are predicting user participation based on previous interactions with other users on common posts. We argue that if user A participates in all posts where B is participating, there is a high chance of A participating in a new post where B is already active. The method of matching items in different transactions is called association rule mining. We apply association rule mining to the domain of social media where we model the data as follows. Items correspond to users on Facebook and transactions correspond to posts. An user is considered to be active and part of the transaction, as an item, if the user comments on a post.

To build association rules from our dataset, we evaluated several implementations. Agrawal et al. [27] presented the Aprori algorithm, which was proven to be an efficient method for association rule learning. This algorithm is however proven to have efficiency issues in large datasets [28] and the identified implementation for Python is very slow (considering our dataset it was not possible to get a result within reasonable time). Hence, other algorithms were tested, and in particular the Eclat algorithm [29]. The Eclat algorithm quickly discards items with low frequency by considering a minimum of associations as input parameters.

From the selected dataset, described in Sect. 9.3.1, we firstly count the frequency of all posts where A and B are active respectively. Secondly, we count all posts where $A \cup B$, both participates. This gives us two measures, length (the number of participating users) and frequency (the sum of all posts where they are participating). These two steps can be summarized
as, building frequent item-sets ($I$). Finally, all possible rules from the computed $I$s are generated. In this step we also compute the evaluation metrics described below.

### 9.4.1 Evaluation metrics

To understand the learned association rules, there exist a few metrics. First, we have *Support*, where we compute the frequency of a given item-set, $I$, and divide it with the total number of transactions (posts) in $D$. Or, the number occurrence of $\{A, B\}$ in our dataset, $D$ divided by length of $D$. As shown in (9.1).

$$ \text{support}(\{A, B\}) = \frac{\{A, B\}}{|D|} \tag{9.1} $$

Secondly, we have *Confidence*, which is an indicator saying that $\{A, B\} \Rightarrow C$ in the set of transactions in $D$ is the proportions of transactions that contain $\{A, B\}$ also will contain $C$ as illustrated in (9.2). Say that $\{A, B, C\}$ participates in 4 common posts and $\{A, B\}$ participates in 8 posts in total. This leads to $4/8 = 0.5$ i.e., the confidence that $C$ will participate on a post where $A$ and $B$ already are active is $50\%$.

$$ \text{confidence}(\{A, B\} \Rightarrow C) = \frac{\text{support}(\{A, B, C\})}{\text{support}(C)} \tag{9.2} $$

Thirdly, we have *lift*, a ratio of the interdependence of the observed values. As we see from (9.3), if lift is 1, it implies that the rule and the items are independent of each other. However, if the lift is $>1$, the lift indicates the degree of dependency of our item-sets.

$$ \text{lift}(\{A, B\} \Rightarrow C) = \frac{\text{support}(\{A, B, C\})}{\text{support}(\{A, B\}) \times \text{support}(\{C\})} \tag{9.3} $$

Finally, we have *conviction*, as the ratio of the expected support that $\{A, B\}$ occurs without $C$ as shown in (9.4). Notable, conviction is infinite (due to division with zero) when the confidence is 1.

$$ \text{conviction}(\{A, B\} \Rightarrow C) = \frac{1 - \text{support}(\{A, B\})}{1 - \text{confidence}(\{A, B\} \Rightarrow C)} \tag{9.4} $$
The described measures enable understanding of the learned rules in $\mathcal{D}$, where higher number of all four measures indicate that the learned rule has relevance for prediction.

### 9.5 Methodology

The final dataset used in the experiment consists of 2,443 users interacting on 610 posts and writing 14,117 comments. The selected users are or have been fairly active in the community, which reflect how we build the association rules.

The algorithm used for the association rule mining is the Eclat algorithm. The Eclat algorithm learns about all the frequent item-sets in our data. By using Eclat, it is possible to define a lower bound threshold and in our dataset a good trade-off between resolution and speed is 4, where lower frequency is ignored. The used implementation of Eclat is modified to sort the item-sets by participants so only $\{A, B, C\}$ is considered. Other combinations e.g., $\{B, C, A\}$ and $\{C, A, B\}$ are consolidated in the item-set $\{A, B, C\}$. Association rules supporting the hypothesis of user participation based on other users activities were computed from the calculated frequency item-sets. The results are measured using the evaluation metrics presented in Sect. 9.4.1.

### 9.6 Results

The resulting frequent item-sets are depicted in Figure 9.1. This figure illustrates the length of elements (number of collaborating users) with respect to frequency (the number of occurrence for each item-set). The main scatter-plot illustrates how the frequency decreases when the number of users (length) increases, a natural feature of frequent item-sets.

Figure 9.1 also depicts the distribution as histograms. The top histogram shows the distribution of frequency and the histogram on the right hand side shows the distribution of the length of the learned item-sets. The top histogram illustrates a significant density of user collaboration to occur at low frequency, between 4–6. This is natural as the frequency of user participation decreases for most of the users. Noticeable on the length distribution is the fact that the density is higher for two and three
participating users than for just one. This is because there exist more combinations of users than the number of single users.

Association rules supporting the hypothesis of user participation based on other users activities were computed from the calculated frequency item-sets. Resulting in 55,166 rules. Table 9.1 shows descriptive statistics for all the computed rules. It can be noted that although the confidence median and mean is low, the high level of lift indicates high dependency of the learned rules, i.e., the computed rules show that out hypothesis is valid and users tend to follow each other. As our dataset is big, with many users and many posts, the low support mean and median is expected. Moreover, it is noticeable that users are not active in all posts but more on a subset of them.

Figure 9.2 depicts the distribution, Confidence, Lift, Conviction and Frequency respectively in our learned model. The figures are illustrated as violin-plots which represents the kernel density (shown as height and depth) in addition to normal box-plots with outer quartiles as thin lines, the inner quartiles as bold lines and the mean as a white dot.
### Table 9.1: Descriptive statistics of 55,166 computed rules.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>median</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>0.05</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>Confidence</td>
<td>0.43</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Lift</td>
<td>18.97</td>
<td>9.38</td>
<td>24.64</td>
</tr>
<tr>
<td>Conviction</td>
<td>1.83</td>
<td>1.32</td>
<td>1.18</td>
</tr>
</tbody>
</table>

Figure 9.2: Distribution of values in learned association rules.

(a) Support distribution

(b) Confidence distribution

(c) Lift distribution

(d) Conviction distribution

Figure 9.2a shows a dense distribution of support at 0.025 and interestingly a higher density at 0.20. The confidence distribution is illustrated in Figure 9.2b, interestingly there is a dense distribution around 1.0, i.e., there is a significant number of learned rules with high confidence that the rule is accurate. Figure 9.2c shows that the lift measure have a heavy tail distribution. Figure 9.2d illustrates a distribution of conviction to be concentrated between zero and five.

Table 9.2 presents learned rules in tree sections. Each section is sorted firstly by Confidence, Lift and Conviction respectively and secondly by the number of supporting users. The rule \{u_{429}, u_{578}\} \Rightarrow \{u_{19}\} should be interpreted as user–429 together with user–578 influence participation of user–19. Notable, when sorting by confidence and lift, the conviction is infinite (this is due to the confidence of 1.0) as shown how conviction is calculated in (9.4). All of the rules in Table 9.2 have high confidence and show high dependency (via the lift metric), i.e., the top five rules sorted
by either Confidence, Lift or Conviction are relevant for predicting user participation.

The rule, \{u_{580}, u_{861}, u_{1352}, u_{1466}\} \Rightarrow \{u_{896}, u_{1291}\} presented in Table 9.2 with confidence of 1.0 and lift of 152.5 strongly indicates that the left-hand-side user set influences the right-hand-side user set, i.e., when the left-hand-side user set is active on a post the right-hand-side user set also will be active. A confidence of 1.0 means that 100\% of the posts where the left-hand-side user set is active, the right user set also will be active. A lift value of 152.5, in this specific rule, shows that the right-hand-side user set is dependent on the left.

Considering rules where at least two separate users affect another user with a confidence of \geq 95\%. We can reduce the 55,166 rules to 4,959 rules, which have a median lift of 4.80 and a median support of 21\%. In other words, we have close to 5,000 rules that strongly indicates that users are affected by each other when it comes to participating in online social networks. From learned rules, we can also identify influential users, i.e., the users that exists on the left side of multiple rules.
9. Predicting User Participation in Social Media

9.7 Conclusion and Discussion

Users within online social networks creates a large amount of generated data in form of interactions (comments and likes). Not enough attention has been put on the prediction of how users influence each other and how to predict the behavior of users within Facebook groups. Therefore, we have, in this paper, crawled a significant amount of user data and then by using machine learning, implemented and examined how users influence each other. Based on the results and analysis, we are able to determine that users influence other users to participate and interact in new groups.

From the group OccupyTogether, 2,443 active users have been extracted. They interact on 610 posts with a total of 14,117 comments. From this dataset, the association rules were computed. Resulting in almost 5,000 rules with high confidence of correctness, \( \geq 95\% \). These rules were proven to be dependent of the active users, via the lift metric. Therefore, the hypothesis of user participation influences can be accepted. The results also proved that using association rule learning, influential users can be identified. Moreover, users on the left-hand-side, in a rule with high confidence and high lift, are influencing users on the right-hand-side to participate in the conversation.

At present, information on Facebook are filtered by a secret algorithm. This poses a potential validity threat to our results. Even external recommender systems might pose a threat as data might be bias since users can only see a subset of all posts.

For future work, it would be interesting to compare the results across different Facebook groups, e.g. politics-related Facebook group is different from news-related Facebook groups. Additionally, methods for association rule learning that supports number of occurrence and order of items in each transaction also needs to be investigated further. Finally, investigating the temporal aspects of users participation, e.g. whether users influence each other over time, or if a user participates throughout a discussion or only in the beginning, is something that needs to be considered and which could hopefully improve the prediction results.
Acknowledgement

This work was partially supported by the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 316097 [ENGINE] and by The National Science Centre, the decision no. DEC-2013/09/B/ST6/02317

9.8 References


9. Predicting User Participation in Social Media


9. Predicting User Participation in Social Media


Finding Influential Users in Social Media Using Association Rule Learning

Fredrik Erlandsson, Piotr Bródka, Anton Borg, Henric Johnson

Abstract

Influential users play an important role in online social networks since users tend to have an impact on one other. Therefore, the proposed work analyzes users and their behavior in order to identify influential users and predict user participation. Normally, the success of a social media site is dependent on the activity level of the participating users. For both online social networking sites and individual users, it is of interest to find out if a topic will be interesting or not. In this article, we propose association learning to detect relationships between users. In order to verify the findings, several experiments were executed based on social network analysis, in which the most influential users identified from association rule learning were compared to the results from Degree Centrality and Page Rank Centrality. The results clearly indicate that it is possible to identify the most influential users using association rule learning. In addition, the results also indicate a lower execution time compared to state-of-the-art methods.

10.1 Introduction

Online social networks are playing an important role in our society and have created a platform for people to communicate and express their thoughts. With the use of online social media, we have created a way to mimic real human communication in an online environment. Facebook alone attracts 1.3 billion users with 640 million minutes spent each month on the site. Consequently, discovering trending topics or influential users is of interest for many researchers interested in areas such as marketing [1]. Several
studies have tried to identify user influence; however, most have used Page Rank Centrality [2, 3] or Degree Centrality [3, 4] based approaches to identify influential users. This paper builds on the initial discoveries on association rule learning in social networking sites: [5].

In this article, we argue that users on Facebook groups are following each other and that it is possible to detect influential users and predict user participation. For example, if users A, B, C and D share common interests, there is a chance that if A, B, and C already have commented on a topic, D will also comment on it. Therefore, this paper relates to how users perform actions (e.g., comments or likes) on posts in Facebook pages. In addition, we use association rule learning to discover relationships between users in our dataset [6]. Given a list of posts from a specific domain, we extract users’ actions, such as comments and likes. Using association rule learning on the data, we argue that it is possible to predict if a particular user will or will not participate on a post discussion based on the other users’ activity.

This article has three major contributions: firstly, possibilities to identify influential users using association rule learning are presented; secondly, we present time performance of well-known methods for ranking users in social media together with our approach using association rule learning; and finally, we show how association rule learning can be used to predict user participation.

For evaluation, several experiments are conducted, which include building association rules that can be used to predict if a specific user will be active in a particular post. The prediction is done based on the activeness of users within current posts. In addition, an extended social network analysis is conducted to verify the findings of influential users.

The paper is organized as follows: in Section 10.2, related work is discussed; in Section 10.3, association rule learning and the evaluation metrics are discussed; in Section 10.4, the dataset is presented; and finally, the results are presented in Section 10.5 and discussed in Section 10.6.

10.2 Related Work

Online social networks and social media analysis are popular research areas in contemporary network science. The main focus in social network research
is on link prediction [7] and social connection prediction [8]. Different teams around the world also work on: (i) personality prediction for micro blog users [9], (ii) churn prediction and its influence on the network [10, 11], (iii) community evolution prediction [12, 13], (iv) using social media to predict real-world outcomes [14], (v) predicting friendship intensity [15, 16], (vi) affiliation recommendations [17, 18], and (vii) sentiment analysis and opinion mining [19].

Other popular areas of research focus on popularity prediction in social media based on comment mining [20], predicting information cascade on social media [21], and predicting patterns of diffusion processes in social network [22]. An important factor is often the user’s role in the different processes. As such, identifying influential users are of interest to understand and/or affect the spread of information, e.g., viral marketing. The ability to identify influential users might also affect the research into other areas of related work (e.g., ii or iii).

Research into detecting influential users on Twitter indicates that, while a large amount of followers seem to be present among influential users, predictions of which particular user will be influential is unreliable [23]. Depending on the social network, how to define influence differs, e.g., influence on Twitter might be defined by retweets or mentions, while, on Digg, votes generated are used to measure influence [1, 24, 25]. While some initial research has been done using clustering algorithms to identify top users, based on influence features, e.g., likes and replies, evaluation is lacking [26]. Similarly, linear regression has been used to identify influential (categorical) users based on influence features [25].

While some research on identifying influential users use learning based approaches, another popular approach to identifying influential users is the Page Rank algorithm or adaptations of the Page Rank algorithm [27–29].

Nancy et al. [30] explore the association rule between a course and gender in the Facebook 100 university dataset. This was performed to discover the influence of gender in studying a specific course. Yu et al. [31] introduce the scheme for association rule learning of personal hobbies in social networks, while Schmitz et al. [32] tackle the problem of mining association rules in folksonomies and try to find out how association rule learning can be applied to analyze and structure folksonomies.
Initial research used association rule learning to identify influential users and predict user participation in online social networks [5]. Association rule learning has been previously used in social network and social media analysis.

While online social network analysis is popular, there is, according to our review, a lack of research on using association rules for predicting user participation in online social media discussions.

### 10.3 Association Rule Learning

Association rule learning is a machine learning technique that aims to find out how one item affects another by analyzing how frequently certain items appear together in a specific dataset. This is done by using two criteria, namely, support and confidence. Support indicates the frequency of such items, while confidence indicates how many times those rules in the whole dataset are correct. An example of an association rule is the following: “Ninety-percent of transactions that purchase bread and butter also purchase milk” [33].

As stated in Section 10.1, we are trying to assess user participation in a post based on previous interactions with other users on common posts within one page. We assume that if user A participates in most of the posts where user B is participating as well, there is a high chance of A participating in a new post where B is already active, either because participation of B influences A to participate and/or they both have similar interests. The method of matching items in different transactions is called association rule learning. We apply association rule learning to the domain of social media where we model the data as follows. Items correspond to users on Facebook and transactions correspond to posts. A user is considered to be active and part of the transaction as an item if the user comments on a post.

From the selected dataset described in Section 10.4, we firstly count the frequency of all posts where A and B are active, respectively. Secondly, we count all posts where $A \cup B$ both participate. This gives us two measures, length (the number of participating users in the set) and frequency (the sum of all posts where the users are participating). These two steps can be summarized as building frequent item-sets ($\{I\}$). Finally, all possible rules
from the computed \(I\)'s are generated. In this step, we also compute the evaluation metrics described below.

### 10.3.1 Evaluation Metrics

Several metrics exist that will help understand the learned association rules. The first measure, *Support*, shows how big of a portion of \(D\) the item-set covers. It is calculated by dividing the frequency of a given item-set, \(I\), with the total number of transactions (posts) in our dataset, \(D\), or the number occurrences of \(\{A, B\}\) divided by the number of items in \(D\). As shown in Equation (10.1):

\[
support(\{A, B\}) = \frac{\{A, B\}}{|D|}. \tag{10.1}
\]

The second measure, *Confidence*, indicates the proportions of transactions that contain \(\{A, B\}\) that also will contain \(C\) in the set of transactions in \(D\), given the following rule \(\{A, B\} \Rightarrow C\). *Confidence* is calculated as shown in Equation (10.2). Say that \(\{A, B, C\}\) participates in four common posts and \(\{A, B\}\) participates in eight posts in total. This leads to \(4/8 = 0.5\), or the *confidence* that \(C\) will participate on a post where \(A\) and \(B\) already are active is 50%:

\[
confidence(\{A, B\} \Rightarrow C) = \frac{support(\{A, B, C\})}{support(\{A, B\})}. \tag{10.2}
\]

The third measure, *lift*, shows the ratio of interdependence of the observed values. As we see from Equation (10.3), if *lift* is 1, it implies that the rule and the items are independent from each other. However, if *lift* is > 1, the *lift* indicates the dependency of our item-sets:

\[
lift(\{A, B\} \Rightarrow C) = \frac{support(\{A, B, C\})}{support(\{A, B\}) \times support(\{C\})}. \tag{10.3}
\]

Finally, *conviction* is the ratio of the expected *support* that \(\{A, B\}\) occurs without \(C\) as shown in Equation (10.4). Notably, *conviction* is infinite (due to division with zero) when the *confidence* is 1:

\[
conviction(\{A, B\} \Rightarrow C) = \frac{1 - support(\{A, B\})}{1 - confidence(\{A, B\} \Rightarrow C)}. \tag{10.4}
\]
The described measures enable understanding of the learned rules in \( D \), where higher numbers of all four measures indicate that the learned rule has relevance for prediction.

### 10.3.2 Usage of the Eclat Algorithm

To build association rules from our dataset, we evaluated several implementations. Agrawal et al. [34] presented the Aprori algorithm, which was proven to be an efficient method for association rule learning. However, this algorithm is proven to have efficiency issues in large datasets [35], and the identified implementation for Python is very slow (considering that in our dataset it was not possible to get a result within a reasonable time). Hence, other algorithms were tested, in particular, the Eclat algorithm [36]. The Eclat algorithm quickly discards items with low frequency by considering a minimum number of associations as input parameters. We have found that a reasonable trade-off between resolution and speed is four, in our dataset, where a lower frequency of items is ignored. The use of four as a lower bound was identified empirically by starting at the number of comments divided by the number of users and then calculating the item-sets with decreasing threshold until the execution speed reached 10 s. At 10 s, all available RAM memory in our experiment environment was exhausted, and we stopped the execution. For one of the investigated pages, we saw that with a threshold of five, we can generate 4230 item-sets in 350 ms, and with a threshold of four, we can generate 9117 item-sets in 600 ms. A threshold of three fills up available resources and never completes the calculations.

### 10.4 Data Model

The data used in this study have been obtained from the crawler described by Erlandsson et al. [37]. This crawler gathers complete posts from Facebook. In this context, the term complete, stands for posts that contain all likes and comments created up to the crawling time as well as the data about the users who have created them. Our current dataset, captured from public pages and groups on Facebook, consists of over 56 million posts, 560 million comments and 7.3 billion likes made by 820 million Facebook users. The crawled data was parsed and made available from an SQL database, structured as described in [38], making all fields needed for our task available. In this study, we assume that the investigated posts will
not get any new comments. We simplify the dynamics of social media by saying that the posts we are investigating were “dead” when the data was collected, in which the term of dead posts refers to posts that no longer attract attention, new comments, or likes.

This study is limited to only active users. Thus, we exclude posts with less than 20 comments and users who had less than five comments, as they are considered to be occasional visitors and not real page participants.

**Data Selection**

We have sampled 195 pages from our dataset, varying in terms of the number of users, posts, comments and user activity to make the sample of Facebook data as broad and as diverse as possible. Despite the fact that we have calculated the rules using a server with 144 GB of RAM memory and a 24 core processor, we could not calculate the rules for the biggest pages (44 of them), thus we had to remove them from our dataset. An example of such a page is *Fox News* with 837,176 users 4485 posts, 6,967,304 comments, and a lifetime of 2034 days (almost six years). An additional 43 pages had to be removed because they were too small, *i.e.*, having less than 10 posts with more than 20 comments and/or less than 10 users with more than five comments. After the preprocessing, we still had 108 pages ranging from 152 to 675,200 active users, from 18 to 161,264 posts, and from 577 to 1,340,730 comments. Table 10.1 presents the descriptive statistics of this dataset.

For the initial results, the page [39] has been selected. This page was selected based on the following properties: it is active, it has a high number of users, and it is political with a biased user group (most of the users have positive perceptions of the Occupy movement). It was also selected as it is a page in the median range of the complete dataset with respect to the number of active users, 2443, and active posts, 610.

<table>
<thead>
<tr>
<th>Type</th>
<th>Mean</th>
<th>Std.</th>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>69,678</td>
<td>130,564</td>
<td>152</td>
<td>4282</td>
<td>17,995</td>
<td>62,194</td>
<td>675,200</td>
</tr>
<tr>
<td>Posts</td>
<td>7431</td>
<td>19,329</td>
<td>18</td>
<td>784</td>
<td>2157</td>
<td>5758</td>
<td>161,264</td>
</tr>
<tr>
<td>Comments</td>
<td>147,721</td>
<td>264,711</td>
<td>577</td>
<td>7886</td>
<td>33,437</td>
<td>133,421</td>
<td>1,340,730</td>
</tr>
</tbody>
</table>

*Table 10.1: Filtered descriptive statistics of the dataset of 108 pages.*
10.5 Experiments and Results

To verify the findings, several experiments were executed. These experiments were firstly performed on the page OccupyTogether, and were extended to the whole dataset described in Section 10.4 for verification of the results. First, a comprehensive experiment of association rule learning was conducted. Secondly, the learned rules were evaluated with respect to prediction accuracy of user participation using a training test split (80/20). Finally, social network analyses for each page were performed to verify and evaluate ranked users identified as influential by the first experiment.

10.5.1 Item-Sets and Rules

Using the methods described in Section 10.3.2, an experiment was performed to create frequent item-sets and build association rules for these sets. The resulting frequent item-sets are depicted in Figure 10.1 for the page OccupyTogether. This figure illustrates frequency, or the number of occurrences for each item-set, with respect to the length of elements, or the number of collaborating users. The main scatter-plot illustrates how the frequency decreases when the number of users (length) increases, a natural feature of frequent item-sets. Figure 10.1 also depicts the distribution as histograms. The top histogram, in green, shows the distribution of frequency and, the histogram on the right hand side, in red, shows the distribution of the length of the learned item-sets. The histogram to the right (in green) illustrates a significant density of user collaboration that occurs at a low frequency, between 1 and 10. This is natural as the frequency of user participation decreases for most of the users. Noticeable on the length distribution (in red) is the fact that the density is higher for two and three participating users than for just one. This is because there exist more combinations of users than the number of single users.
Association rules supporting the hypothesis of user participation based on other users’ activities were computed from the calculated frequency item-sets. This resulted in 55,166 rules for the page OccupyTogether. Table 10.2 shows descriptive statistics for all the computed rules. It can be noted that although the confidence median and mean is low, the high level of lift indicates a high dependency of the learned rules, i.e., the computed rules show that our hypothesis is valid and users tend to follow each other. Since our dataset is big, with many users and many posts, a low support mean and median is expected. Moreover, it is noticeable that users are not active in all posts but more on a subset of them.

Table 10.2: Descriptive statistics of 55,166 computed rules.

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>Mean</th>
<th>Median</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>0.05</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>Confidence</td>
<td>0.43</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Lift</td>
<td>18.97</td>
<td>9.38</td>
<td>24.64</td>
</tr>
<tr>
<td>Conviction</td>
<td>1.83</td>
<td>1.32</td>
<td>1.18</td>
</tr>
</tbody>
</table>
Figure 10.2 depicts the distribution, Confidence, Lift, Conviction and Frequency respectively in our learned model. The figures are violin-plots, which illustrate the kernel density (shown as height and depth) in addition to normal box-plots with outer quartiles as thin lines, inner quartiles as bold lines and the mean as a white dot.

Figure 10.2a shows a dense distribution of support at 0.025 and, interestingly, a higher density at 0.20. The confidence distribution is illustrated in Figure 10.2b, in which we obtained a dense distribution around 1.0, \textit{i.e.}, there are a significant number of learned rules with high confidence, thus, the rule is accurate. Figure 10.2c shows that the lift measure has a heavy tail distribution. In addition, Figure 10.2d illustrates a distribution of conviction to be concentrated between zero and five.

Table 10.3 presents learned rules in three sections. Each section is sorted firstly, by Confidence, Lift and Conviction, respectively, and secondly by the number of supporting users. The rule \{u_{429}, u_{578}\} $\Rightarrow$ \{u_{19}\} should be interpreted as user 429 together with user 578 influencing the participation of user 19. Notably, when sorting by confidence and lift, the conviction is infinite (this is due to the confidence of 1.0) which is shown in how conviction is calculated in Equation (10.4). All of the rules in Table 10.3 have high confidence and show high dependency (via the lift metric), \textit{i.e.},
the top five rules sorted by either Confidence, Lift or Conviction are relevant for predicting user participation.

The rule, \( \{ u_{580}, u_{861}, u_{1352}, u_{1466} \} \Rightarrow \{ u_{896}, u_{1291} \} \) presented in Table 10.3 with a confidence of 1.0 and a lift of 152.5, strongly indicates that the left-hand-side user set influences the right-hand-side user set, \( i.e. \), when the left-hand-side user set is active on a post, the right-hand-side user set also will be active. A confidence of 1.0 means that 100% of the posts where the left-hand-side user set is active, the right-hand-side user set also will be active. A lift value of 152.5, in this specific rule, shows that the right-hand-side user set is dependent on the left.

Table 10.3: Top 5 rules sorted by different metrics for the Facebook page Occupy-Together.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Confidence</th>
<th>Lift</th>
<th>Conviction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( { u_{179}, u_{538}, u_{580}, u_{938}, u_{992}, u_{1096} } \Rightarrow { u_{11} } )</td>
<td>1.00</td>
<td>10.17</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{11}, u_{31}, u_{80}, u_{179}, u_{992}, u_{1093} } \Rightarrow { u_{580} } )</td>
<td>1.00</td>
<td>4.80</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{11}, u_{31}, u_{179}, u_{580}, u_{992}, u_{1093} } \Rightarrow { u_{80} } )</td>
<td>1.00</td>
<td>9.53</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{11}, u_{179}, u_{538}, u_{580}, u_{938}, u_{953} } \Rightarrow { u_{429} } )</td>
<td>1.00</td>
<td>4.84</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{179}, u_{1094}, u_{1096}, u_{1113}, u_{1171}, u_{1352} } \Rightarrow { u_{1378} } )</td>
<td>1.00</td>
<td>101.67</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{580}, u_{861}, u_{1352}, u_{1466} } \Rightarrow { u_{896}, u_{1291} } )</td>
<td>1.00</td>
<td>152.50</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{580}, u_{861}, u_{1352}, u_{1466} } \Rightarrow { u_{896}, u_{1291} } )</td>
<td>1.00</td>
<td>152.50</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{31}, u_{80}, u_{179}, u_{580} } \Rightarrow { u_{11}, u_{992}, u_{1093} } )</td>
<td>1.00</td>
<td>152.50</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{19}, u_{64}, u_{673}, u_{685} } \Rightarrow { u_{54}, u_{581} } )</td>
<td>1.00</td>
<td>152.50</td>
<td>∞</td>
</tr>
<tr>
<td>( { u_{580}, u_{861}, u_{1291}, u_{1466} } \Rightarrow { u_{896}, u_{1352} } )</td>
<td>1.00</td>
<td>152.50</td>
<td>∞</td>
</tr>
</tbody>
</table>

Considering rules where at least two separate users affect another user with a confidence of \( \geq 95\% \), we can reduce the 55,166 rules to 4959 rules, which have a median lift of 4.80 and a median support of 0.21. In other words, we have close to 5000 rules that strongly indicate that users are affected by each other when it comes to participating in online social networks. From learned rules, we can also identify influential users, or the users that exists on the left side of multiple rules as presented in Section 10.5.3.

The learned rules of the complete dataset are presented in Table 10.4, after filtering out rules with Confidence \( \geq 95\% \).
Table 10.4: Descriptive statistics of learned rules with of Confidence $\geq 95\%$ from the complete dataset.

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>Mean</th>
<th>Std.</th>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q4</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of rules</td>
<td>33,426.89</td>
<td>87,457.39</td>
<td>2.00</td>
<td>151.00</td>
<td>2351.00</td>
<td>32,053.50</td>
<td>724,510.00</td>
</tr>
<tr>
<td>Confidence</td>
<td>1.00</td>
<td>0.00</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lift</td>
<td>38.06</td>
<td>42.14</td>
<td>1.41</td>
<td>10.86</td>
<td>25.34</td>
<td>47.91</td>
<td>217.53</td>
</tr>
<tr>
<td>Conviction</td>
<td>19.39</td>
<td>4.61</td>
<td>5.88</td>
<td>18.07</td>
<td>19.79</td>
<td>20.70</td>
<td>29.46</td>
</tr>
</tbody>
</table>

10.5.2 Verification of Learned Rules

To test how well association rule learning works for predicting user participation, a split, learn and test pattern have been used. For the page in question, we sort all comments based on creation time and use the first 80\% for learning and the last 20\% of the posts for testing. The learning part is performed as described in Section 10.3.2, and the testing part is carried out as follows: for each post with comments in the testing set, the active users are considered by finding rules that affect the users with respect to temporal order. Say that user $D$ is commenting on a post (in the testing set), and there exists a rule saying that $A, B & C$ affect user $D$, this rule will only be considered to be valid if all of $A, B & C$ have made at least one comment each before $D$ makes a comment. Of the 787 intersecting users between the learning and test sets, it is possible to predict 113 (14.36\%) users, making use of 5310 (9.63\%) of the original 55,166 rules.

To calculate accuracy and precision of learned rules, we have defined true/false positive/negatives as follows: A true positive is a rule that predicts user activeness, and the user is active. A false positive is when a rule predicts user activeness, but the user is not active. A true negative is when no user is active, and there is no rule. A false negative is when a user is active, but there is no rule. An example of all four classes are shown in Table 10.5.

Table 10.5: Example of false positives and false negatives. Capital letters indicates users and $P_{1-4}$ corresponds to different posts.

<table>
<thead>
<tr>
<th>Example rule: ${A, B, C} \Rightarrow {D}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1 = {A, B, C, D}$ → true positive</td>
</tr>
<tr>
<td>$P_2 = {A, B, C}$ → false positive</td>
</tr>
<tr>
<td>$P_3 = {F, G, H}$ → true negative</td>
</tr>
<tr>
<td>$P_4 = {D, E}$ → false negative</td>
</tr>
</tbody>
</table>
For the page OccupyTogether, an accuracy of 0.886, precision of 0.291, and recall of 0.071 was calculated, with a testing time of 9175 s. This result is quite low since all learned rules are being considered. To portray a more realistic view of user influence, the rules were limited to only consider rules with confidence $\geq 95\%$ and rules affecting a single user. Rules affecting more than one user are already covered by the rules affecting a single user, reducing the number of learned rules from 46,170 to 4469 and the execution time down to 890 s. Showing an accuracy of 0.927, precision of 0.794, and recall of 0.017. The testing was also performed on the rest of the pages and the results are reported in Table 10.6. The recall is low because there are many false negatives (calculated with $\frac{TP}{TP+FN}$). The relatively high accuracy is then achieved with a relatively high number of true negatives used in $\frac{TN+TP}{TP+FP+TN+FN}$. In general, the unfiltered rules show a lower accuracy, precision, and recall compared to the filtered rules. Furthermore, the complexity of the rule set is reduced by filtering the rules, indicating the beneficial use of rule filtering. The rules set was on average reduced by approximately 93%. A less complex rules set could be easier to test and also to understand.

Table 10.6: Testing of learned rules based on a 80/20% learn and test split. SD stands for standard deviation.

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>OccupyTogether</th>
<th>OccupyTogether$^a$</th>
<th>All pages (SD)</th>
<th>All pages$^a$ (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of rules</td>
<td>46,170</td>
<td>4469</td>
<td>99,237 (248,968)</td>
<td>7092 (14,965)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.886</td>
<td>0.927</td>
<td>0.858 (0.135)</td>
<td>0.906 (0.128)</td>
</tr>
<tr>
<td>Precision</td>
<td>0.291</td>
<td>0.794</td>
<td>0.286 (0.287)</td>
<td>0.633 (0.343)</td>
</tr>
<tr>
<td>Recall</td>
<td>0.071</td>
<td>0.017</td>
<td>0.138 (0.193)</td>
<td>0.165 (0.258)</td>
</tr>
</tbody>
</table>

$^a$ Reduced set of rules limited by having Confidence $\geq 95\%$ and only affected one user.

10.5.3 Identifying and Verifying Influential Users Using Social Network Analysis

The state-of-the-art method for identifying influential users is social networks analysis (SNA), using the methods Page Rank Centrality [3] or Degree Centrality [40] for ranking users. It is of interest to see how well influential users identified using association rule learning (ARL) match the state-of-the-art techniques. Therefore, we have conducted an SNA of our pages as follows: for each page, we have created social networks in such a way that two users are linked together if they commented on the same post; next, for all social networks, Page Rank [3] and Degree [40] measures have
been calculated; and, based on those measures, two ordered (descending) user lists were created, one for each of them.

We have created a similar list for the most influential users from association rule learning. Most influential users are defined as the top-\( k \) users from the left side of the rules, with a confidence level of greater than 95%, that affect other users to comment on posts. In the most influential users list, users are ranked based on how often they appear on the left side of the rule, e.g., if user \( A \) has appeared three times in all rules and users \( B, C \) and \( D \) have appeared one, five and four times, respectively, and the list will look as follows: \([C, D, A, B]\).

Finally, we compared the most influential users identified from association rule learning with top users according to the degree and Page Rank. Comparison between association rule learning, Degree and Page Rank are considered the top 1%, 5%, 10%, 25%, 50%, 75%, and 100% of the most influential users identified by association rule learning, respectively. The comparison was made as an intersection of two sets created from two lists. For example, if the top four users are \([A, B, C, D]\) for Degree and \([F, A, C, D]\) for association rule learning, the intersection of those two sets will be \([A, C, D]\) and the size of that set is three, and, in this case, the similarity is 75%.

The example of the SNA analysis for one of the pages OccupyTogether is presented in Table 10.7. The table shows that for the top 209 users on the page OccupyTogether (the 50% most influential users from association rule learning), there is a similarity of 95% between the users ranked by Page Rank and Degree. When considering users ranked from association rule learning, there is a similarity of 51% compared to Degree and 53% compared to Page Rank.

<table>
<thead>
<tr>
<th>Percent of Top Users</th>
<th>Users</th>
<th>Degree ∩ ASR</th>
<th>Page Rank ∩ ASR</th>
<th>Page Rank ∩ Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 %</td>
<td>4</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>5 %</td>
<td>20</td>
<td>0.45</td>
<td>0.45</td>
<td>0.95</td>
</tr>
<tr>
<td>10 %</td>
<td>41</td>
<td>0.488</td>
<td>0.512</td>
<td>0.927</td>
</tr>
<tr>
<td>25 %</td>
<td>104</td>
<td>0.462</td>
<td>0.49</td>
<td>0.923</td>
</tr>
<tr>
<td>50 %</td>
<td>209</td>
<td>0.512</td>
<td>0.526</td>
<td>0.947</td>
</tr>
<tr>
<td>75 %</td>
<td>313</td>
<td>0.502</td>
<td>0.556</td>
<td>0.92</td>
</tr>
<tr>
<td>100 %</td>
<td>418</td>
<td>0.517</td>
<td>0.565</td>
<td>0.928</td>
</tr>
</tbody>
</table>

Table 10.7: Comparison of similarity of influential users for the page OccupyTogether.
From the SNA analysis, we detected yet another interesting insight into users’ behavior in social media pages. We noticed that 10% of users with the highest value of degree measure, created an average of 82.64% posts, and an additional 10% of the most important users add only four more percentage points of posts, i.e., 20% of users with the highest value of the degree measure, create 86.84% posts on average. In Figure 10.3, the distribution of that phenomena is depicted for all pages.

As described above, the three different approaches were used to detect the most influential users. The intersection between the different user lists were then calculated to evaluate how much each method differs from the others. To detect whether any statistical significant difference exists, Friedman’s test was used with the Nemenyi post hoc test. Friedman’s test is a non-parametric statistical test that ranks the methods over datasets [41]. When a normal distribution cannot be assumed and several datasets are used, Friedman’s test has been suggested as preferable when comparing algorithms [42]. The Nemenyi post hoc test evaluates between which intersections a significant difference exists. The means and standard deviation for the intersections of several posts are presented in Table 10.8. A low standard deviation indicates that the expected value, i.e., the intersection between two sets, is close to the mean. However, there might still exist results which are not close to the mean, e.g., as seen in Table 10.7.

Figure 10.3: Distribution of posts created by top users over 108 sampled pages.

The average shows that, regardless of the size of the intersection, Page Rank $\cap$ Degree has more users in common than the other intersections, while
Page Rank and Degree, considered state-of-the-art, have a high amount of users in common (see \textit{Page Rank} $\cap$ \textit{Degree} in Table 10.8), the rule based learner has fewer users in common with both the Page Rank (\textit{Page Rank} $\cap$ \textit{ARL}) method and the Degree method (\textit{Page Rank} $\cap$ \textit{Degree}).

### Table 10.8: Average intersection measurement and average rank using Friedman’s test.

<table>
<thead>
<tr>
<th>Percent of Top Users</th>
<th>Degree $\cap$ ASR (SD)</th>
<th>Page Rank $\cap$ ASR (SD)</th>
<th>Page Rank $\cap$ Degree (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>0.092 (0.173)</td>
<td>0.131 (0.227)</td>
<td>0.822 (0.238)</td>
</tr>
<tr>
<td>5%</td>
<td>0.081 (0.145)</td>
<td>0.095 (0.158)</td>
<td>0.805 (0.251)</td>
</tr>
<tr>
<td>10%</td>
<td>0.115 (0.158)</td>
<td>0.133 (0.173)</td>
<td>0.830 (0.219)</td>
</tr>
<tr>
<td>25%</td>
<td>0.181 (0.188)</td>
<td>0.194 (0.198)</td>
<td>0.836 (0.167)</td>
</tr>
<tr>
<td>50%</td>
<td>0.231 (0.212)</td>
<td>0.257 (0.228)</td>
<td>0.848 (0.129)</td>
</tr>
<tr>
<td>75%</td>
<td>0.266 (0.243)</td>
<td>0.286 (0.249)</td>
<td>0.868 (0.119)</td>
</tr>
<tr>
<td>100%</td>
<td>0.286 (0.261)</td>
<td>0.304 (0.264)</td>
<td>0.886 (0.114)</td>
</tr>
</tbody>
</table>

Average Rank: 3 2 1

Friedman’s test shows that there are some significant differences between the intersects, $\chi^2 = 9.210$, $df = 2$, $p = 0.01$. The Nemenyi test result (see Table 10.9) demonstrates that the \textit{Page Rank} $\cap$ \textit{Degree} set performs significantly better than the \textit{Degree} $\cap$ \textit{ARL} set at a confidence level of both 0.95 and 0.99.

### Table 10.9: Paired rank comparison of intersections using the Nemenyi post hoc test. The upper triangle shows difference between intersections. Lower triangle shows pairs with statistical significance.

<table>
<thead>
<tr>
<th>Compared Measures</th>
<th>Degree $\cap$ ASR</th>
<th>Page Rank $\cap$ ASR</th>
<th>Page Rank $\cap$ Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree $\cap$ ARL</td>
<td>-</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Page Rank $\cap$ ARL</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
</tr>
<tr>
<td>Page Rank $\cap$ Degree</td>
<td>* *</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* significant at $p < 0.05$, CD: 1.253; ** significant at $p < 0.01$, CD: 1.557.

The three different methods were investigated to identify influential users. The amount of time needed to identify influential users differs between the methods. This is shown in Table 10.10. Rule based learning is suggested to be the fastest method, and Page Rank the slowest. This might be explained by Page Rank being a global measure compared to the Degree, which is a local measure. The execution time of the different methods with the confidence intervals are also presented in Figure 10.4, where intuitively it would seem that the rule based learner has a significantly lower execution time than the other methods.
Table 10.10: Mean execution time for ranking users.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>329.135</td>
<td>(2345.996)</td>
</tr>
<tr>
<td>Page Rank</td>
<td>633.152</td>
<td>(4602.607)</td>
</tr>
<tr>
<td>ASR</td>
<td>9.033</td>
<td>(22.497)</td>
</tr>
</tbody>
</table>

Figure 10.4: Execution time for different social network analysis methods.

Whether there is any statistical significant difference is evaluated using a Kruskal–Wallis test followed by a pair-wise Wilcoxon post hoc test [41]. The Kruskal–Wallis test is used to see if there is a significant difference between any of the methods, and the post hoc test is used to detect between which methods the differences exist. The Kruskal–Wallis test detected a significant difference between the methods ($\chi^2 = 6.626$, $df = 2$, $p < 0.05$). The Wilcoxon post hoc tests showed a significant difference between Rule based and Degree ($p < 0.05$, $w = 14130$). No other statistical significant differences were found. While there exists a large difference in mean, there is no detectable significant difference between the Association Rule based method and Page Rank($p = 0.054$, $w = 13704$). This might be due to the high standard deviation.

10.6 Discussion

Users within online social networks create a large amount of generated data in the form of interactions (comments and likes). Not enough attention has been put on the analysis of how users influence each other and how to predict the behavior of users within Facebook groups. In this paper, we have collected a significant amount of user data and then by using association rule learning, implemented and examined how users influence each other. Based on the results and analysis, we are able to determine
to what extent users influence other users to participate and interact in new groups.

To verify the results from the page OccupyTogether, an additional 195 pages were sampled to verify our assumptions. These pages were reduced to 108 due to size constraints. Arguably, pages that were too large could have been processed by limiting the time span, i.e., instead of considering all six years of the page, a time span of the latest six months could have been considered. Association rules were computed for each page in our dataset. For association rules with confidence $\geq 95\%$, the mean was 33,426.89 ($sd = 87,457.39$), and a median of 2351 was found for the number of rules.

The computed rules were tested resulting in an average of 0.913 ($sd = 0.115$) for accuracy, 0.614 ($sd = 0.340$) for precision, and 0.141 ($sd = 0.256$) for recall when predicting user activity on a post. In other words, it is possible to predict a subset of users’ future participation with high correctness.

The results also indicate that influential users can be identified using association rule learning. That is, users on the left-hand-side, in a rule with high confidence and high lift, are influencing users on the right-hand-side to participate in the conversation. These results have been verified and compared with the traditional network analysis methods, Page Rank Centrality and Degree Centrality. Showing that at best $\sim 30\%$ of the users ranked using association rule learning overlap with the users ranked using traditional methods.

Interestingly, association rule learning are magnitudes faster in execution time for ranking users than other methods. Another finding related to the ranking of users is that we see no significant difference between ranked influential users based on Page Rank or Degree. However, we show that Page Rank is a more time consuming algorithm.

The main disadvantage of association rule learning is the fact that we cannot extract rules for the biggest pages in our dataset. We have not shown in this paper that association rule learning is better/or worse than other approaches. However, it was not the point of our research. Since there is no ground truth, it is not possible to say which approach is better (or worse). Our objective was to present a different approach for identifying influential users and leave the final decision of which approach to use to the researcher.
Furthermore, from the list of influential users, presented in Section 10.5.3, it is also possible to limit the size of the item-set. This will result in an increasing speed when building rules without a significant decrease in quality of the rules. As a validation threat, information on Facebook is filtered by a secret algorithm. This poses a potential validity threat to our results as users are presented posts filtered by the algorithm. For example, a reason for a user not commenting on a post might be due to visibility (the filtering algorithm is not presenting the post to the user) rather than by topic.

10.7 Conclusions

This article presents four contributions. Firstly, insights on user behavior on public pages on Facebook indicates that the top 10% and top 20% of users corresponds to a vast majority of the content. Secondly, it is possible to identify influential users using association rule learning. The results indicate no statistically significant difference between our rule based method compared to Page Rank. Thirdly, execution times of well known methods for ranking users in social media together with our approach using association rule learning are investigated. The results suggest that rule based ranking of users has lower execution time compared to state-of-the-art methods, 9.0 vs. 633.1 and 329.1 seconds on average. Finally, the article verifies how association rule learning can be used to predict user participation in social media pages on Facebook. The results indicate an average prediction accuracy of 0.913 (sd = 0.115) for the association rule learning approach.

For future work, it would be interesting to investigate rule creation with a time series perspective of the data e.g., using a sliding window approach. Additionally, methods to investigate a subset of users for rule creation need to be investigated.

Acknowledgments: This work was partially supported by the European Union’s Seventh Framework Program for research, technological development and demonstration under grant agreement No. 316097 [ENGINE] and by The Polish National Science Center, decision No. DEC-2013/09/B/ST6/02317

Author Contributions: Fredrik Erlandsson and Piotr Bródka conceived and designed the experiments; Fredrik Erlandsson performed the experiments; Fredrik Erlandsson, Piotr Bródka and Anton Borg analyzed
the data; Henric Johnson enabled the work and also contributed with critical revision. All authors have written, read, and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## 10.8 References


1. Finding Influential Users in Social Media Using Association Rule Learning


10.8. References


10. FINDING INFLUENTIAL USERS IN SOCIAL MEDIA USING ASSOCIATION RULE LEARNING


Seed selection for information cascade in multilayer networks

Fredrik Erlandsson, Piotr Bródka, Anton Borg

Abstract

Information spreading is an interesting field in the domain of online social media. In this work, we are investigating how well different seed selection strategies affect the spreading processes simulated using independent cascade model on eighteen multilayer social networks. Fifteen networks are built based on the user interaction data extracted from Facebook public pages and tree of them are multilayer networks downloaded from public repository (two of them being Twitter networks). The results indicate that various state of the art seed selection strategies for single-layer networks like K-Shell or VoteRank do not perform so well on multilayer networks and are outperformed by Degree Centrality.

11.1 Introduction

Since the emergence of Network Science [1] one of the most interesting research questions was: How the influence and information spread through the network of social interactions and how to maximize it? [2] There are many approaches to maximize the final coverage of the spreading and one of them is selecting proper set of initial seeds which will initialize the process. This set should consist of nodes with the highest combined potential to reach as big portion (in terms of no. of members) of network as possible. Those node are often called Influential users and play an important role in information propagation on online social networks as they have the highest impact on other users in the network.

While the problem of seed selection is quite well investigated in single layered networks with many state of the art methods like K-Shell [3] or
11. Seed selection for information cascade in multilayer networks

VoteRank [4]. The question is if those approaches will still be the best for multilayer networks which are an relatively new trend in how to model complex networks? [5][6] Therefore, in this paper we evaluate four seed selection strategies: Degree Centrality [7], K-Shell [3], VoteRank [4] and ARL [8] (Section 11.2.3), using Independent Cascade Model (ICM) [9] to simulate the spreading process (Section 11.2.2) over fourteen multilayer networks are built based on the user interaction data extracted from Facebook public pages (Table 11.1) and tree multilayer networks downloaded from a public repository (Table 11.2).

The results are presented in Section 11.3 and indicate that various state of the art seed selection strategies for single-layer networks like K-Shell or VoteRank do not perform so well on multilayer networks and are outperformed by simple Degree Centrality.

11.2 Methods

This section describes the dataset used in our research and social networks created based on it, the information cascade model and various seed selection methods together with the statistical methods used to evaluate our findings.

11.2.1 Dataset and network model

The dataset used in this study is a subset of public Facebook pages collected by Erlandsson et. al. [10] and is publicly available at Harvard Dataverse [11]. The data from these pages were parsed and for each post the corresponding likes and comments were extracted. We considered each page a separate dataset/network. Table 11.1 shows the basic information about investigated 14 Facebook pages.

From each page we build two bipartite networks, one for users’ comments and one for users’ likes. An example of these two networks are shown in Fig.11.1a where the network shown to the left illustrates comments for the users A – E towards the posts 0 – 10, and the network on the right illustrates likes (from the same set of users to the same set of posts). From these two networks we create a multilayer network as shown in Figure 11.1b. In the multilayer network the posts have been removed and the interactions towards the post were replaced by direct connection between users interacting with that post. Nodes represents users, and
11.2. Methods

Table 11.1: Descriptive information of used pages. The columns nodes and edges show the number of elements for the projected networks created for Comments, and Likes respectively.

<table>
<thead>
<tr>
<th>Page id</th>
<th>Posts</th>
<th>Users</th>
<th>Comments</th>
<th>Likes</th>
<th>C edges$^\dagger$</th>
<th>C nodes$^\dagger$</th>
<th>L edges$^\dagger$</th>
<th>L nodes$^\dagger$</th>
<th>Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86</td>
<td>297</td>
<td>50</td>
<td>549</td>
<td>31</td>
<td>24</td>
<td>5,200</td>
<td>270</td>
<td>685</td>
</tr>
<tr>
<td>2</td>
<td>301</td>
<td>303</td>
<td>227</td>
<td>502</td>
<td>130</td>
<td>48</td>
<td>542</td>
<td>157</td>
<td>1,030</td>
</tr>
<tr>
<td>3</td>
<td>1,163</td>
<td>2,326</td>
<td>499</td>
<td>2,161</td>
<td>4,361</td>
<td>273</td>
<td>146,231</td>
<td>1,332</td>
<td>3,823</td>
</tr>
<tr>
<td>4</td>
<td>1,777</td>
<td>801</td>
<td>1,932</td>
<td>4,170</td>
<td>1,770</td>
<td>359</td>
<td>4,996</td>
<td>549</td>
<td>7,879</td>
</tr>
<tr>
<td>5</td>
<td>1,013</td>
<td>1,636</td>
<td>1,463</td>
<td>6,880</td>
<td>3,036</td>
<td>403</td>
<td>85,684</td>
<td>1,502</td>
<td>9,356</td>
</tr>
<tr>
<td>6</td>
<td>5,819</td>
<td>5,861</td>
<td>1,466</td>
<td>25,125</td>
<td>4,832</td>
<td>366</td>
<td>2,437,479</td>
<td>5,670</td>
<td>32,410</td>
</tr>
<tr>
<td>7</td>
<td>9,391</td>
<td>23,431</td>
<td>18,571</td>
<td>19,623</td>
<td>11,694</td>
<td>3,462</td>
<td>904,901</td>
<td>14,492</td>
<td>47,585</td>
</tr>
<tr>
<td>8</td>
<td>538</td>
<td>13,222</td>
<td>11,274</td>
<td>36,033</td>
<td>285,095</td>
<td>5,566</td>
<td>2,249,954</td>
<td>11,141</td>
<td>47,845</td>
</tr>
<tr>
<td>9</td>
<td>1,697</td>
<td>33,004</td>
<td>16,398</td>
<td>39,914</td>
<td>808,650</td>
<td>10,086</td>
<td>5,396,069</td>
<td>26,206</td>
<td>57,916</td>
</tr>
<tr>
<td>10</td>
<td>1,445</td>
<td>22,488</td>
<td>1,946</td>
<td>58,695</td>
<td>11,335</td>
<td>1,219</td>
<td>16,109,395</td>
<td>21,626</td>
<td>62,086</td>
</tr>
<tr>
<td>11</td>
<td>14,736</td>
<td>37,090</td>
<td>26,559</td>
<td>44,124</td>
<td>151,619</td>
<td>9,325</td>
<td>2,950,437</td>
<td>24,324</td>
<td>85,419</td>
</tr>
<tr>
<td>12</td>
<td>14,159</td>
<td>69,424</td>
<td>31,209</td>
<td>147,710</td>
<td>1,600,003</td>
<td>14,637</td>
<td>33,547,079</td>
<td>56,641</td>
<td>193,078</td>
</tr>
<tr>
<td>13</td>
<td>1,187</td>
<td>104,558</td>
<td>18,568</td>
<td>278,173</td>
<td>352,789</td>
<td>11,722</td>
<td>100,171,084</td>
<td>100,541</td>
<td>297,928</td>
</tr>
<tr>
<td>14</td>
<td>10,781</td>
<td>40,368</td>
<td>84,484</td>
<td>420,257</td>
<td>2,097,013</td>
<td>14,554</td>
<td>49,337,665</td>
<td>36,294</td>
<td>515,522</td>
</tr>
</tbody>
</table>

$^\dagger$ nodes represent users, disconnected nodes (without edges) have been removed for clarity.
$^\ddagger$ edges are present if two users have acted on the same post.

edges between two users indicates that they interacted with the same post, i.e. either they both liked it or they both commented on it. The blue layer represents comments and the green layer represents likes. Each node represent the same user on each layer, i.e., node A in the Comments layer is the same user as node A in the Likes layer.

To complement that and to ensure that our findings are not a result of some Facebook properties or the way in which we have prepared our networks we have added to our experiments three social networks from an open repository$^1$, shown in Table 11.2. Please note that in order to be able to compare the results if some network has more than two layers we are using just two of them.

Table 11.2: Descriptive information of used networks.

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Users</th>
<th>Interactions</th>
<th>L1 nodes</th>
<th>L1 edges</th>
<th>L2 nodes</th>
<th>L2 edges</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Podgett Florentine Families</td>
<td>15</td>
<td>35</td>
<td>15</td>
<td>20</td>
<td>11</td>
<td>15</td>
<td>[12]</td>
</tr>
<tr>
<td>17</td>
<td>Marthin Luther King 2013</td>
<td>327,707</td>
<td>378,462</td>
<td>288,738</td>
<td>291,083</td>
<td>79,070</td>
<td>82,987</td>
<td>[13]</td>
</tr>
</tbody>
</table>

11.2.2 Independent Cascade Model

In this study the Independent Cascade Model (ICM) [9] was used for modeling information spreading. ICM requires a set of activated nodes at

$^1$http://deim.urv.cat/manlio.dedomenico/data.php
11. Seed selection for information cascade in multilayer networks

Figure 11.1: A toy example of multiple user interactions as a multilayer network. (a) The bipartite networks of comments (left) and likes (right). (b) The projected multilayer network from the networks shown in (a). The green layer represents likes (users like the same post) and the blue layer represents comments (the users are commenting on the same post). Edges between layers (coupling edges) exists between users liking and commenting on the same post. Node B in the Comments layer represents the initial seed. Nodes and edges with shades ranging from yellow to dark red represents the infection spread.

the beginning (seeds) and runs over a limited number of diffusion steps where recently activated nodes has one chance to activate each of its neighbors with the currently configured activation probability. Thus, if node A is activated on step 3 it can activate it’s neighbors only in step 4, but not in the following steps. In our case we ran experiments with the activation probability 1%. This value was selected due to the high average node degree (edges connected to each node) in the multilayer network. We have limited the ICM to 10 diffusion steps as the number of activated nodes converge here, and the number of seeds to be 1% of the number of nodes in the network. Here 1% means that we identify 1% seeds from each layer and then iteratively select one seed from each layer until we have 1% in total. Say that we have \{A, B, C, D\} from layer 1 and \{E, F, G, H\} from layer 2. This will result in the following seeds \{A, E, B, F\}. The used implementation of ICM activates nodes separately per layer on each step and the set of activated nodes is updated after each step by summing activated nodes sets from each layer. I.e., diffusion is computed fist for the Likes layer and then for the Comments layer, if node B is activated in the Likes layer in the current step the node B will also be activated on the Comments layer before continuing to the next diffusion step.
Figure 11.1b illustrates the spreading process for our toy example. The initial seed is the node $B$ in the Comments layer (shown in white). As we are considering each node to be infected after each infection step we let node $B$ in the Likes layer to also be infected (illustrated as yellow). After the first infection step the node $E$ in the Comments layer and nodes $C$ and $A$ in the Likes layer activated. Resulting in the nodes $C$ and $A$ in the Comments layer and node $E$ in Likes layer also to become infected. In step two the node $D$ in the Likes layer is activated, thus activates node $D$ in the Comments layer. With this model and toy network all nodes are activated after just two steps.

11.2.3 Seed selection

Influential users or, activation seeds were selected using three network based state of the art methods Degree Centrality [7], K-Shell [3], and VoteRank [4], together with a machine learning method, ARL [8] as an efficient and accurate method for ranking users on social networks. We also included a Random sample of seeds as a baseline. All of the investigated methods for seed selection are ranking methods and we select the top nodes with highest rank to use as seeds for the ICM.

Degree Centrality is a network measure which indicates how many connections with the rest of network each node has, it has the advantage of being easy to compute once the network is created. K-shell is a measure that is determined using shell decomposition. The highest K-Shell number is considered to be the “core” of the network. To efficiently rank seeds we combined the K-Shell rank with degree. By doing this we have created hybrid measure which eliminate K-Shell disadvantage i.e. it does not have enough granularity and multiple nodes can belong to the same K-Shell thus one would have to choose the seeds randomly. VoteRank selects seeds iteratively by letting each nodes’ neighbors vote using a penalized model where nodes close to an already selected seed will have a decreased voting score/power, and already selected node will not have voting rights.

Using ARL to identify seeds have the advantage of not requiring creating the network before identifying influential users. The chosen ARL algorithm Eclat [14] also have the advantage of being able to reduce the dataset by using a threshold, saying that a user must be active on at least a predefined number of posts to be included in the computation. In this work we do
not use a fixed value of this threshold, instead we start with an relatively large number and decrease this number until we hit a computational limit and then we use the lowest successful limit and return the computed result. The reported timings shown in Fig. 11.4 illustrate the computation time for the final selected threshold as in a real world setting this threshold value will be configured before running the ARL algorithm. The major time consumption of ARL is in building list of when users appear together. For example, user A is active on posts 1, 2 & 3, user B is active on posts 2 & 3 et cetera. A typical rule is \( \{A, B\} \Rightarrow C \), i.e. if A and B appears so will C. A limitation of ARL is that it only ranks a subset of the users. The ranked users are then used as seeds for ICM. The ranked number of seeds identified by ARL is used by VoteRank to limit the number of seeds computed and to compare the models fairly.

11.2.4 Statistical evaluation

The coverage is used as an evaluation metric at each step in the ICM. As such, it is possible at each step to measure how quickly the information spreads. Most research today uses the final coverage as the primary evaluation metric [3, 4, 15]. However, as can be seen in Figure 11.2, the coverage tend to stabilize over seed selection methods after a certain amount of steps. As such, there are drawbacks to using the final coverage as evaluation metric. First, there is always a chance that different algorithms converge to the same final coverage. Second, evaluating the mean or the median of the coverage will also give misleading measurements, as it doesn’t take into account the development rate of the coverage. Consequently, evaluating only on the final coverage is inadvisable.

As such, in this study the primary evaluation metric is the area under curve of coverage (AUC), i.e. how much area will there be under the coverage curve. A larger area denotes a faster rise in coverage, a higher coverage, or both. In this study is the AUC normalized based on the number of diffusion steps computed.

The AUC captures the development of the coverage over the steps in the ICM. Consequently, comparing the AUC allows the comparison of the methods performance on pages. The AUC is calculated using the MESS R-package. It should be noted that the AUC is not to be confused with the
11.3. Results

AUROC (Area Under Receiver Operating Characteristics curve), which is often colloquially referred to as AUC.

To investigate whether any statistical significant difference exist between the different methods, the Friedman test is used [16]. The Friedman test is a non-parametric test that evaluates different treatments (in this case different seed selection algorithms) over multiple datasets. A non-parametric test is chosen over a parametric as normality cannot be assumed over the different datasets. As the test only detects whether a statistical significant difference exists, and not where the difference exists, a post-hoc test is necessary to determine where the difference is located. The Nemenyi test is used as a post-hoc test [16].

11.3 Results

We have run experiments for eighteen multilayer networks, 1% activation probability, 1% of nodes as initial seeds and five seed selection strategies. This resulted in 90 combinations of experiment parameters. For each combination we run 10 simulations of spreading process using Independent Cascade Model (ICM). The results show that selecting seeds with high Degree Centrality performs the highest activation coverage and also is the simplest and thus fastest method for seed selection.

To illustrate how different activation probabilities and how ICM behaves in both single- and multilayer networks we ran ICM on one of the pages with different settings. Figure 11.2 shows the spreading process for the page no. 8 for different activation probabilities (1% for Fig 11.2a and Fig 11.2c, and 10% for Fig 11.2b and Fig 11.2d) and two different network types. This two types are a multilayer network created from the users’ Comments (first layer) and Likes (second layer), shown in Fig 11.2a and Fig 11.2b; and a single layer network created from the users’ Comments, shown in Fig 11.2c and Fig 11.2d. Please note that the plots for the multilayer graph reaches higher coverage faster than the plots for the single-layer graph as the multilayer graph is more dense, see Table 11.1 for more information.

11.3.1 The final coverage for various seed selection methods

Figure 11.3 shows the resulting mean AUC for the 17 pages investigated. The relatively low AUC is due to some of the multilayer networks have
many connected components and that the seeds are just selected from a few of these components.

The Friedman found significant differences between the seed selection methods over the pages ($\chi^2 = 43.333$, $df = 4$, $p = 8.824e^{-09}$), with respect to activation coverage with an activation probability of 1%. The Nemenyi post-hoc test, presented in Table 11.3, shows statistical significant differences between Degree Centrality and a Random sample, ARL, and K-Shell. Further, There were also a statistical significant difference between VoteRank and a Random sample when comparing the AUC.

As such, the results indicates that Degree Centrality perform significantly better than the other seed selection methods (except VoteRank), i.e.
11.3. Results

Figure 11.3: Mean AUC of the activation coverage for different seed selection methods. The AUC is calculated on 10 steps and 1% activation probability of the activation coverage from ICM in a multilayer network.

Table 11.3: Nemenyi post-hoc test for detecting statistically significant differences between the different seed selection methods with an AP = 1% with respect to the mean AUC of activation coverage.

<table>
<thead>
<tr>
<th></th>
<th>Degree</th>
<th>Random</th>
<th>ARL</th>
<th>K-Shell</th>
<th>VoteRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>—</td>
<td>3.059</td>
<td>2.088</td>
<td>2</td>
<td>0.794</td>
</tr>
<tr>
<td>Random</td>
<td>*</td>
<td>—</td>
<td>0.971</td>
<td>1.059</td>
<td>2.265</td>
</tr>
<tr>
<td>ARL</td>
<td></td>
<td>*</td>
<td>—</td>
<td>0.088</td>
<td>1.294</td>
</tr>
<tr>
<td>K-Shell</td>
<td>*</td>
<td></td>
<td>—</td>
<td>—</td>
<td>1.206</td>
</tr>
<tr>
<td>VoteRank</td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

*: Significant difference at $p < 0.01$ Critical Difference: 1.765

**: Significant difference at $p < 0.001$ Critical Difference: 2.103

the AUC for this method were significantly larger than for the other seed selection methods in general. Further, VoteRank is significantly better than a Random sample. Interestingly there is no significant difference between a Random sample and either ARL or K-Shell, i.e., selecting seeds using these methods were not statistically better than selecting seeds at random.
11.3.2 Time complexity of seed selection methods

Figure 11.4 show the time complexity of the investigated pages for seed selection with the four different methods. Both VoteRank and ARL are slower than the other methods. The execution time for each page in Fig. 11.4 is an average from ten runs, and the error bars are indicating the standard deviation.

For the three network based seed selection algorithms (Degree, K-Shell and VoteRank) the major time complexity consumer shown in Fig.11.4 is the network creation from our dataset. On the other hand, the major time consumer for the ARL method is the building of item sets. Further more, we only calculate the same number of seeds for VoteRank as the ARL method identified, while for Degree and K-Shell we calculated the ranking for the whole network.

A Friedman test shows significant differences between seed selection methods ($\chi^2 = 29.314$, $df = 3$, $p = 1.923e^{-06}$). The Nemenyi post-hoc test found significant differences between Degree Centrality and all other methods when comparing time complexity for seed selection, e.g. Degree
performed significantly faster than the other methods. K-Shell, VoteRank, and ARL are significantly slower than Degree Centrality and there is no internal significant difference between these methods, see Table 11.4.

Table 11.4: *Nemenyi post-hoc test for detecting statistically significant differences between the different seed selection methods with respect to time complexity.*

<table>
<thead>
<tr>
<th></th>
<th>Degree</th>
<th>K-Shell</th>
<th>ARL</th>
<th>VoteRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>—</td>
<td>1.429</td>
<td>2.286</td>
<td>2.286</td>
</tr>
<tr>
<td>K-Shell</td>
<td>*</td>
<td>—</td>
<td>0.857</td>
<td>0.857</td>
</tr>
<tr>
<td>ARL</td>
<td>*</td>
<td>*</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>VoteRank</td>
<td>* **</td>
<td>* **</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

*: Significant difference at $p < 0.05$ Critical Difference: 1.254

**: Significant difference at $p < 0.001$ Critical Difference: 1.832

11.4 Conclusion

We have evaluated five seed selection strategies to see how they affects information cascade in multiplex networks. The evaluation was made on 14 public pages on Facebook, two datasets with Twitter data, and one dataset describing Florentine families in the Renaissance.

The results show that Degree Centrality and VoteRank performs best for seed selection in multiplex networks. The results show that although ARL can be used for seed selection in an information cascade setting it is not preferred as it performs equally as a Random sample. Further, Degree Centrality is significantly faster than the other methods. If we take into consideration both time complexity and final number of activated users the Degree Centrality is the most optimal seed selection strategy for all tested networks.

Acknowledgement

This work was partially supported by The Polish National Science Centre, the decision no. DEC-2016/21/D/ST6/02408; the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 691152 (RENOIR) and the Polish Ministry of Science and Higher Education fund for supporting internationally co-financed projects in 2016-2019 (agreement no. 3628/H2020/2016/2).
11.5 References


Do we really need to catch them all? A new User-guided Social Media Crawling method

Fredrik Erlandsson, Piotr Bródka, Martin Boldt, Henric John-son

Abstract

With the growing use of popular social media services like Facebook and Twitter it is challenging to collect all content from the networks without access to the core infrastructure or paying for it. Thus, if all content cannot be collected one must consider which data are of most importance. In this work we present a novel User-guided Social Media Crawling method (USMC) that is able to collect data from social media, utilizing the wisdom of the crowd to decide the order in which user generated content should be collected to cover as many user interactions as possible. USMC is validated by crawling 160 public Facebook pages, containing content from 368 million users including 1.3 billion interactions, and it is compared with two other crawling methods. The results show that it is possible to cover approximately 75% of the interactions on a Facebook page by sampling just 20% of its posts, and at the same time reduce the crawling time by 53%. In addition, the social network constructed from the 20% sample contains more than 75% of the users and edges compared to the social network created from all posts, and it has similar degree distribution.

12.1 Introduction

In the age of big data, billion of people are using social media, such as Facebook, Snapchat, Twitter, and Instagram, to socialize, interact and create new content at a remarkable rate [1, 2]. Facebook alone increased
its number of users with 13% between 2015 and 2016, and in May 2017 the count had reached 1.75 billion active users [2]. This massive amount of data is now also available (to some extent) to crawl. However, with limited resources and due to the complexity and speed in which new content is generated, there is a need for improved strategies on what content to focus on.

In previous studies, we have developed the Social Interaction Network Crawling Engine (SINCE) [3, 4] that collects publicly available Facebook data. Over a period of four years we have collected content generated by 1.6 million unique Facebook users interacting on 110 million posts through 1.9 billion comments and 31 billion likes. The SINCE crawler is novel and unique as it is the first crawler capable of gathering data in depth by covering all interactions within posts. An important challenge throughout the crawling process is how to measure the quality of the collected data as it is based on different aspects related to the application for which the data is intended to be used, e.g., whether the data is used for measuring similarity among users’ interactions; whether the data provides diversified perspectives on certain topics; or whether the data is a statistically representative sample of the complete data.

Crawling data from social media comes with two inherent problems. First, that the data volume is so large that it is close to impossible to continuously gather all content. Secondly, that only a subset of the data is relevant for a specific application, or is interesting to researchers. The crawler used for collecting social media data in this work (i.e. SINCE) struggles under both these inherent problems. This is why this work introduces and evaluates a novel method for crawling social media data more efficiently, without requiring any priori knowledge about the network itself.

This study considers publicly available data published in open pages and groups on Facebook. The aim of the study is to investigate how to efficiently and precisely crawl quality data from Facebook’s social network using the introduced User-guided Social Media Crawling (USMC) method. We investigate if the novel prioritization and ranking techniques in USMC can be used to exclude posts that are of less interest during crawling, in order to both reduce the crawling time and at the same time increase the number of included social interactions. USMC ranks posts based on metadata metrics, such as the number of likes and then selects the highest
ranked posts. Thus, these metadata metrics are used for estimating the number interactions that posts are likely to receive.

The goal of the study is to evaluate to what extent the proposed USMC method is able to estimate the importance of content by relying on the wisdom of the crowd and without any a priori knowledge about the underlying social network. That is, utilizing the users’ interactions in the social network for pointing the crawler to which data that is of most importance. Using this approach we then investigate the trade-off between crawling speed and degree of data coverage in the crawling process. Finally, the proposed USMC method is evaluated against a random sampling without replacement approach [5, 6] as well as a novel chronological crawling approach where posts are sampled based on their lifetime, i.e. for how long posts have been active.

### 12.1.1 Motivation

Social media interactions, and especially Facebook data, have been growing massively during the last decade [2], and there is an interest from both the research community, industry and the society at large to be able to collect and analyze this data [7]. The interaction data and online data in general are essential for social media analytic solutions, reputation tracking systems, brand monitoring and other big data solutions [7, 8]. The fact that Facebook has no intention to sell this data (due to its value) has been a motivation for developing and presenting a novel way to collect social interaction data from publicly available pages.

Data from social media is big, both in terms of volume and in terms of velocity (new data is constantly created and grow faster than we can crawl). Since it is unfeasible to collect all the data there is a need to address the issue of how to prioritize the data while crawling. So that as much data as possible is collected for future research. This is especially important as the crawling make use of both limited time and computational resources. However, data from social media can be treated differently depending on the requirements associated with the intended use of the data, i.e. the future application. The users’ interactions are highly relevant for social network analysis in different areas such as: identification of important users; or seed selection for information and influence spreading in complex networks. As such, we adopt a quantitative data measuring strategy by
regarding the quality of the crawled data as equivalent with the proportion of all available social interactions in the social media services.

The USMC method is interesting for anyone with limited resources that systematically wants to collect content from social media services, or similar web-based sources. However, future users need to be aware of the limitation and potential bias enforced with the USMC method, i.e. that the resulting data excludes low interaction volumes. Examples of analyzes made possible using data crawled by the USMC method include community detection [9] and identification of influential users [10, 11].

12.1.2 Limitations

The USMC method, like all types of sampling, introduces some limitation/bias to the collected data. Fortunately, with the USMC method the bias introduced is known in advance, since the method disregards posts with low interactions and will most likely omit outliers and special cases present in low interaction posts. For other sampling approaches the bias on the resulting data might not be known a priori, e.g. for chronological sampling the most recent posts are collected but nothing is known about how much of the interactions that are captured.

12.1.3 Related work

There is a lack of research concerning the quality of data in social media and social network research. There are studies on social media and social networks, mainly using data from Twitter. This data is, however, typically collected using Twitter’s free garden hose API with a risk of being unbalanced and an unrepresentative sample of the complete data. Studies that investigate the quality of social media data include [12, 13], where the former addresses how social media data from online recommendation systems can be evaluated. Sampling studies of social networks are quite common, including [14, 15] that uses the original graph sampling study by Leskovec and Faloutsos [16] as a baseline. Wang et al. presents an interesting study [17] on how to efficiently sample a social network with a limited budget. The study uses metrics of the graph to make informed decisions on how to transverse it. More recently Rezvanian et al. [18] presents algorithms to sample weighted networks. Chiericetti et al. [19]
further investigate network sampling methods and how to minimize the number of required queries.

On the topic of graph and social media crawling Zafarani et al. [7] present ways to evaluate and understand the data generated in social media. However, many social media crawling studies are obsolete due to updates by Facebook regarding the default privacy policy of users' content, which makes it impossible by default to access Facebook users' content [13, 20–22]. As a consequence, the amount of private Facebook data that can be collected is severely limited. Furthermore, since Facebook does not sell any of its data there is a need for crawling methods that collect social interaction data from publicly available sources, which is the main motivation for this work.

Buccafurri et al. [23] discuss different methods to traverse social networks from a crawling perspective by focusing on public groups rather than individual users’ profiles. Our approach mainly differs from this study in two ways. First, we do not create a social network to traverse and only treat the social media as data, i.e. our proposed method do not require any knowledge of the underlying network. Secondly, we focus on user interactions represented as so called Social Interaction Network (SIN) graphs [24] as it shows the interactions between users in various communities, i.e. SIN graphs can represent interactions of all users on one particular newsgroup or users interacting on a specific topic. To conclude, there are no prior studies, according to our literature review, that address the challenge of collecting data from Facebook after Facebook changed the default privacy policy of its end-users’ content. Most studies use online data repositories and do not address the issue of how to efficiently collect data directly from Facebook, or other social media sites.

12.2 Results

To prioritize data available for crawling, we need to define a set of quality measures which will allow to rank the posts on a page. In this section, we start by testing which of the metadata metrics most accurately assess the importance of a post in terms of how much new knowledge about users’ interactions on that page it will convey. Next, the identified metadata metrics are used when evaluating to what extent the USMC method can increase the number of interactions collected by the crawler. Finally, we
create a posteriori social networks to validate our findings with network theory.

The SINCE crawler starts by performing an initial crawl of a page, followed by a full crawl of its data [3, 4]. During the initial crawl the SINCE crawler gathers metadata for all posts on a page. For each post, the following three metadata metrics are collected: post lifetime, number of comments, and number of likes. An Ordinary Least Square (OLS) regression test [25] is used to investigate which of the three measures most accurately assesses the total number of interactions (i.e. the total count of likes and comments) on posts based on the sample of 160 randomly sampled Facebook pages. The basic statistics of this dataset is available in Table 12.1 and detailed descriptions of each page are presented in the Supplementary Information Table S2.

Fig 12.1 shows the distribution of $R^2$ for the conducted OLS regression test, which indicates a high confidence, $0.80 \pm 0.26$ (std), that the number of likes can be used to predict the number of interactions of posts. A combination of the three metrics gives the most accurate assessment, $0.86 \pm 0.24$ (std), as illustrated in Fig 12.1. However, in a practical setting a combined metric is not possible because of mainly the following two reasons. First, the ratio between the metrics is unknown a priori to the crawling, which spoil any attempt to create a well-balanced combination of two or more metrics. Further, such a balanced combination is required since the number of likes is much higher than number of comments (as shown in Table 12.1), which means a simple sum of both metrics will not work as the number of likes will overshadow the number of comments. The second reason is because each metric has different variance per page. That is, each page would require its own tailored version of such combined metrics. Therefore, we deem that combined metrics for prioritizing which content to crawl is practically infeasible, and therefore turn to investigate each metric individually. However, the use of combined metrics could be interesting for future work as such metrics still show best performance in the OLS analysis.

In Fig 12.1 there is a clearly visible separation between the distributions of each of the three metrics. A Friedman’s test $\chi^2 = 299.73$, $df = 2$, $p \ll 0.001$ shows that there is indeed a statistical difference between the distributions of the three metadata metrics. However, there is no statistically significant difference between the number of likes and combined metrics.
12.2. Results

Figure 12.1: This box plot presents the $R^2$ distribution of OLS Regression test assessing the number of interactions on a post, using the following three metadata metrics: post lifetime (blue), number of comments (green), and number of likes (red). The three metrics are also shown as a combined metric (purple). All box-plots are created from a sample of 160 Facebook pages.

Further, a Nemenyi post-hoc test shows (as expected from Fig 12.1) that the number of likes metric is the strongest predictor for the number of interactions, and that all three distributions are statistically different at significance level 0.001.

As identified in the OLS regression analysis, the number of likes is a suitable predictor of the number of interactions on a post. Thus, we use it to rank posts for each page and use that ranking to guide the SINCE crawler on which posts to crawl. We compare the results in terms of number of collected interactions with a traditional random sampling without replacement [5, 6] approach as well as a chronological crawling approach. The results presented in Fig 12.2a show that by implementing the USMC method it is possible to cover a vast majority of the interactions in a page by considering only a fraction of all available posts. For example, on average we need to crawl merely 20% of the posts in order to gather 75% of all interactions when using the USMC method to rank posts based on their number of likes. In addition, a sample size of 20% covers only 20% and 40% of the pages’ interactions using random sampling and chronological crawling approaches respectively. For individual results of all 160 Facebook pages please see Table S1 and Fig S1.

Fig 12.2b illustrates the fraction of crawling time (x-axis) needed to collect a desired proportion of the interactions. It shows that it is possible
to collect just over 50% of the interactions in less than 25% crawling time. That is, approximately twice as many interactions than collected by the random sampling and chronological crawling methods given the same crawling time. The number of interactions collected at any given crawling time has a linear relationship for the random sampling method. For the USMC method this relationship is more favorable when below roughly 80% crawling time. For crawling times longer than 80% the gained efficiency over the random sampling method decreases since the USMC method is gathering the posts that received the least interactions from the crawled page.

Figure 12.2: Average interaction coverage for all 160 Facebook pages for (a) different sample size (represented as a percentage of all post on Facebook a page) and (b) crawling time. It is presented for the USMC method with rankings based on number of likes (red), number of comments (blue), post lifetime (green), chronological crawling (purple), and random sampling method (yellow). For the best and the worst approach, we included error bars showing the standard deviation. The individual results for all 160 Facebook pages are available in the supplementary material, see Fig S1.

The Cohen’s d scores for the findings in Fig 12.2a show that there are large \(d > 0.8\) separation between the three metadata metrics for all sample sizes smaller than 95%. Regarding the crawling time, the Cohen’s d scores show large differences between the metadata metrics for all crawling times shorter than 80%, and medium differences for crawling times between 80 – 95%.

Both Fig 12.2a and Fig 12.2b show that the most efficient approach for USMC is to use the number of likes metric for ranking posts. Therefore, the
next experiments only consider number of likes when comparing the USMC method with both the random sampling and the chronological crawling approaches.

To further validate the proposed USMC method, we investigate how complete and useful the resulting social networks are when constructed from the gathered data. Please note that due to the limitation in computational power we had to exclude the two largest pages from this analysis.

We have created three social networks based on the social interactions collected by the USMC method as well as by the random sampling and chronological crawling methods. For this we relied on the following sample sizes: 1%, 10%, 20%, 30%, 60% and 90% of all posts on each Facebook page. Fig 12.3 shows the number of nodes (a) (Facebook users) and edges (b) (interactions between users) in each social network. It is clear that the USMC method both collects content from significantly more users as well as more social relations between them, compared to the other two methods. Thus, the social network constructed from the data crawled by the USMC method is more complete. In fact, even with merely 20% of the collected posts it is possible to create a network that contains more than 75% of the users and their interactions.

![Figure 12.3: The fraction of all nodes (a) and edges (b) for the social networks constructed from the 1%, 10%, 20%, 30%, 60% and 90% samples of all available posts, using USMC (red), chronological crawling (purple), and random sampling (yellow) approaches. The plot is showing mean and standard deviation for all 160 pages.](image-url)
Next, we performed a social network analysis with respect to degree distribution for each created network. Fig 12.4 presents the degree distribution for the three social networks created from the three representative Facebook pages. These three pages are representatives of the first quantile (Q1), the Median and the third quantile (Q3) regarding the number of posts per page distribution for 158 pages. Fig 12.4 include measurements for the following four sample sizes: 10%, 20%, 30% and 60% out of all posts on each of the three Facebook pages. Fig 12.4 shows that even with a relatively small sample of 20%, the USMC method is able to create a social network with more than 75% of all users and interactions included, and with a degree distribution very similar to the complete social network created from all available data. This result can be seen for all 158 Facebook pages that were analyzed, see supplementary material Fig S2 for the details.

12.3 Discussion

Many times when considering large-scale data gathering from social media services it is not possible to collect all available data as it is too large and the continuous influx is simply too fast to keep up with. In those situations one needs to decide on one of two available data gathering strategies: the deadline-based and the coverage-based strategies. Each one of these strategies consider when the dataset is “good enough” for the intended use of the data. The deadline-based data collection strategy should be adopted when the data collection process has a point in time when it has to be finished, e.g., an upcoming presidential election in four weeks. Following that example, as much data as possible needs to be collected within the given time frame, say three weeks. That way, strategic decisions based on the collected online behavior can help pinpoint which national regions to focus on during the last week before the election day.

The second type of data collection strategy is the coverage-based that specifies a particular sample size of the full dataset that is needed, e.g., that 75% of the original data is required for credibility of particular study. As an example of this strategy think of a particular page that would take 100 days to crawl in full length. By using random sampling a 75% sample would be reached in approximately 75 days, or the USMC method could be used that would collect the required 75% of the interactions in about 45 days. That is, by using USMC a time-saving of 30 days could be expected.
12.3. Discussion

Figure 12.4: The degree distribution for three social networks created out of three representative Facebook pages. Each column shows the sample size (10%, 20%, 30% and 60% of all posts on a page), and each row presents a representative page for each quantile in the number of posts per page distribution for 158 pages. The first row shows the Facebook page Chateau Elan Winery & Resort with 1,131 posts, 25,008 users, and 4,814 comments (Q1); the second Liftopia with 3,973 posts, 47,001 users, and 50,065 comments (Median); and the third San Francisco | The Official Guide with 13,305 posts, 735,183 users, and 116,336 comments (Q3). As shown, a 20% sample of all pages collected by the USMC method allows for the creation of social networks that have almost identical degree distribution compared to social networks created from all available data.

(Fig 12.2b) when compared to random sampling, which is equivalent to a 33% time saving. Further, a time saving of 55 days (or 70%) could be expected by using USMC when compared to collecting the full dataset. Some might object that it is just matter of adding the tight amount of additional resources to speed up the process to solve the data gathering problem. However, very often this is not possibility due to either API restrictions or the equipment available. That is why it is important to study how prioritization of posts could be handled in order to determine where the available resources could be used most efficiently. For the USMC method this translated to benefiting from the wisdom of the crowd of social
media users by relying on their online behaviors for pointing the crawler to which content to target, and in which order.

The goal of covering as many interactions as possible with the limited resources is evaluated in Fig 12.2, which show how the proportion of collected interactions correlate with sample sizes for each of the investigated crawling approaches. For instance, ranking posts based on number of comments covers 78.5 ± 16.7% (std) of the available interactions on a given page, at a sample size of 40% of all posts on that given page. However, ranking posts based on number of likes provides an increased coverage with 86.7 ± 12.5% (std) of all interactions at the same sample size. Fig 12.2b shows the interaction coverage with respect to crawling time for SINCE crawler. These results show that it is possible to decrease the overall crawling time if only the posts that covers the most number of interactions are being crawled, i.e. excluding the posts with least number of interactions.

The evaluation of the UMSc method has revealed that it is a suitable candidate for crawling high-volume data sources from social media services. However, it could be wise to consider other crawling approaches where the a posteriori data analysis is dependent on the interactions on posts with low number of interactions, e.g., Spam mitigation approaches, malicious content detection, or outlier analysis. However, for other application areas the USMC approach is interesting to consider, e.g., community detection analysis, or identification of influential users.

In this study, the USMC method has been evaluated on data from public Facebook pages. However, it is most probable that the same approach could be used for other social media services as well, e.g., Twitter, LinkedIn or ResearchGate. For Twitter, we could rank tweets using the number of re-tweets, likes and responses. Ranking by these attributes would probably allow the collection of social interactions from Twitter to be carried out more efficiently, compared to approaches used today, and at the same time produce representative samples. Similarly, USMC could be applied on the social network at ResearchGate by ranking content based on the number of comments, RG-score, h-index or average number of downloads per article. However, these suggestions need to be validated using research on other social media platforms.
12.4 Materials and Methods

In this section the materials and methods used in this article are described. First, a detailed description of the proposed USMC method is given. Second, the dataset used in the evaluation of the proposed approach is presented. Third, the evaluation methods used in the study are detailed. Fourth, the process of creating a social network from the dataset as well as the social network analysis carried out on that network are being presented. Finally, we describe the various statistical tests used in this study.

12.4.1 User-guided Social Media Crawling

As users interact on social media it is possible to use their actions (e.g. likes or comments) to rank posts. Evaluating data from social media can be made in various forms, but it is hard to computationally evaluate the content. This is why the work proposed in this study makes use of users’ actions in order to make more informed decisions about the social media data, i.e. benefiting from wisdom of the crowd. Users’ actions on posts could be used as indicators of how interesting posts are for the users in the particular community (different communities can have different values and understandings of the subject). The proposed USMC method therefore relies on ”wisdom of the Facebook crowd” to find quality content in social networks as well as a way to rank the posts to capture. In general, the introduced crawling technique ranks content in the social network according to how much attention users give it, i.e. how much interaction each content receives.

In this work, we define social interactions as the type of actions users can take on content in the social network. To put it in Facebook’s terminology, the content is usually a post within a page and the actions are either a like on a post, a comment on a post or a like on a comment on a post. Fig 12.5 illustrates an example of different social interactions as well as how the three sampling methods evaluated in this study can be used for collecting those interactions. This example will be used throughout this section as a platform for describing and discussing various aspects in the crawling process.

In detail, the USMC enabled crawling process works as follows. To begin with the crawler makes a quick initial crawl of a page in order to gather the metadata for each post in that particular page. Next, the USMC
method estimates the total numbers of interactions each post will receive during a given time interval and based on a chosen metric, and then sorts all posts in a list by decreasing order. Regarding the metric used for predicting the total number of interactions that post is likely to receive, the number of likes a post has received (which is available in the meta data) has proved to be a suitable metric. Next, the actual crawling of content from the page starts and continues until either the desired number of interactions has been reached, or the time limit is passed. For each iteration in this process the crawler selects the top most post from the list and carry out a full crawl for that particular post. A complete description of USMC enabled crawling process used by the SINCE crawler is shown in Algorithm 1.

Algorithm 1 USMC enabled crawling with the SINCE crawler.

Require: page_id

post_meta-data ← collect_post_metadata(page_id)

sort (post_meta-data) based on USMC

repeat

post_data ← makeFacebookRequest(/post_id)

data ← data + post_data

repeat

likes ← makeFacebookRequest(/post_id/likes)

data ← data + likes

until likes is empty

repeat

comments ← makeFacebookRequest(/post_id/comments)

data ← data + comments

if comments has likes then

repeat

commentLikes ← makeFacebookRequest(/comment_id/likes)

data ← data + commentLikes

until commentLikes is empty

end if

until comments is empty

saveData(data)

until time is up or required data is collected
12.4. Materials and Methods

The social interactions are exemplified in the toy example shown in Fig 12.5. The eight posts in Fig 12.5a include different number of interactions with regards to likes (shown in red next to the ‘thumb up’ icon) and comments (shown in green number next to ‘speech bubble’ icon). Fig 12.5b shows a bipartite network of the interactions between six users \((U_{1-6})\) and each respective post, where green edges represent comments from users on a particular post and red edges represent likes. Fig 12.5c shows the aggregated network built on users’ interactions on posts collected by the following three sampling approaches: \textit{USMC}, \textit{chronological} and \textit{random} sampling. The full network from all eight posts is shown as dashed edges, while the collected interactions are shown as solid lines. Red edges denote likes on posts and green edges denote comments on posts. Fig 12.5d shows the social networks created based on a 37.5\% sample of all posts when collected by each of the three crawling methods, i.e. \textit{USMC}, \textit{chronological} and \textit{random} sampling. The social network in Fig 12.5d is created as a projection from the bipartite networks shown in Fig 12.5b where the nodes are representing users, and where edges are present if the users have interacted on the same post.

12.4.2 Dataset

The dataset used for evaluating the \textit{USMC} method was created by collecting 160 randomly selected open pages on Facebook. The dataset is available on Dataverse [26]. Table 12.1 shows descriptive statistics for these 160 pages included in the study. The SINCE crawler [3, 4] was used for collecting the 160 Facebook pages between July 2014 and May 2016. SINCE is designed to collect publicly open pages using Facebook’s API. We adhere to Facebook’s data privacy policy [27] by anonymizing all data to an extent where it is only possible to backtrack the particular public page that is analyzed. The resulting dataset has a median page size of 5,235 posts, 180,314 users, 45,592 comments and 442,424 likes. In total, the dataset includes some 368 million unique users interacting in little over 1.3 billion social interactions. However, it should be noted that 2 out of the 160 pages had to be excluded from the network analysis part as social networks could not be generated with the hardware resources available since 148 GB of RAM was not enough to fit the projected network. For complete statistics of all pages please see Table S1 in the supplementary material.
12. Do we really need to catch them all? A new User-guided Social Media Crawling method

Figure 12.5: Example of interactions extracted from posts. Fig (a) shows eight different posts with number of likes (‘thumbs-up’ icon), number of comments (‘speech bubble’ icon), and the age of the posts (‘watch’ icon). Fig (b) shows the bipartite networks of interactions between the six users (U1–6) and the eight posts (P1–8), where red edges denote likes on posts and green edges denote comments on posts. The users are the same on all posts. Fig (c) shows the aggregated networks of user’s interactions towards posts collected by three different crawling methods: USMC, chronological and random sampling. The full network from all eight posts is shown as dashed edges, while the collected interactions are shown as solid lines. Red edges denote likes on posts and green edges denote comments on posts. Fig (d) shows the social networks created based on a 37.5% sample of all posts, collected by the three crawling methods: USMC, chronological and random sampling. The full social network from all eight posts is shown as dashed edges, while the collected edges are shown as solid lines.

12.4.3 Evaluation Methods

We evaluate the USMC method by comparing it to both traditional random sampling without replacement and chronological methods. Random sampling [5, 6] in this context is about collecting posts at random, which gives a representative representation of the data (sampled data will represent the original dataset given the current sample size). During the evaluation each random sampling execution was iterated 100 times and the results report the mean and standard deviation as common in scientific work. The
12.4. Materials and Methods

Table 12.1: Descriptive statistics of the dataset of 160 pages.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Mean</th>
<th>Std.</th>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posts</td>
<td>21,590</td>
<td>60,786</td>
<td>7</td>
<td>1,313</td>
<td>5,235</td>
<td>14,758</td>
<td>470,528</td>
<td>3,454,456</td>
</tr>
<tr>
<td>Users</td>
<td>2,588,338</td>
<td>11,460,281</td>
<td>182</td>
<td>26,589</td>
<td>180,314</td>
<td>897,564</td>
<td>113,379,978</td>
<td>414,134,086*</td>
</tr>
<tr>
<td>Comments</td>
<td>608,873</td>
<td>2,584,414</td>
<td>37</td>
<td>10,638</td>
<td>45,592</td>
<td>230,205</td>
<td>27,550,352</td>
<td>97,419,710</td>
</tr>
<tr>
<td>Likes</td>
<td>7,640,858</td>
<td>33,972,674</td>
<td>384</td>
<td>54,923</td>
<td>442,424</td>
<td>2,589,165</td>
<td>308,495,988</td>
<td>1,222,537,425</td>
</tr>
<tr>
<td>Edges</td>
<td>98,688,366</td>
<td>395,490,320</td>
<td>3</td>
<td>83,080</td>
<td>1,331,539</td>
<td>24,018,804</td>
<td>4,238,052,189</td>
<td>15,592,761,948</td>
</tr>
<tr>
<td>Nodes</td>
<td>154,831</td>
<td>382,031</td>
<td>25</td>
<td>5,613</td>
<td>24,461</td>
<td>115,226</td>
<td>3,020,786</td>
<td>24,463,392</td>
</tr>
</tbody>
</table>

* Of which 368,094,952 are unique users, not overlapping on different pages.

The **chronological** method sorts all posts in decreasing order from oldest to the newest, and samples the oldest posts. Looking at the conceptual example in Fig 12.5 when having specified a sample size of 37.5% of all posts, i.e., 3 out of the 8 posts, the **USMC** will collect the three posts with highest number of likes, i.e. post 1, 6 and 7, while the **chronological** method will collect post 1, 2 and 3 and the **random sampling** collects for instance post 2, 5 and 7.

Each page is evaluated with regards to the number of interactions they capture. Five different sample sizes (10%, 20%, 30%, 60% and 90% of all posts at the Facebook page) are used to represent the page. In the evaluation we also investigate the time it takes to crawl the 160 Facebook pages. In the example in Fig 12.5, each method produces a different set of posts: \{1, 6, 7\} for **USMC**, \{1, 2, 3\} for **chronological** and \{2, 5, 7\} for **random sampling**. These total number of interactions included in each set of posts differs, as can be seen in Fig 12.5c, where **USMC** captures 77% of all interactions while **chronological** and **random sampling** captures 32% and 27% respectively.

12.4.4 Social Network Analysis

The three methods are evaluated by comparing the social networks created from the interactions collected by each method. In these social networks, users are represented as nodes and the edges between them represent social interactions. A social network is created as a undirected graph as \(G = <\mathcal{N}, \mathcal{E}>\), with a set of nodes \(\mathcal{N} = \{n_1, \cdots, n_n\}\) to represent users and a set of edges \(\mathcal{E} = \{<n_i, n_j>: n_i, n_j \in \mathcal{N} \land i \neq j\}\) representing relationship between the users \(i\) and \(j\). The social network of interactions between users is projected from the bipartite network of users and posts, where an edge
12. Do we really need to catch them all? A new User-guided Social Media Crawling method

\(< n_i, n_j >\) is present if both of the users \(i\) and \(j\) have commented on the same post.

Fig 12.5d shows the resulting social networks created by three crawling methods (USMC, chronological and random) using the same sample size of 37.5% of all posts, i.e., of 3 out of the 8 posts. It is clear that USMC creates the most complete network since it includes all of the six existing users and 94% of edges. The chronological and random methods include 67% and 50% of the nodes (users), and 28% and 22% of the edges respectively. Please note that the example shows a multilayer social network with two types of edges based on (i) likes and (ii) comments represented in form of a multi-graph Fig 12.5d. However, as we have mentioned earlier the social network used in our experiments is a single layer network where edges are based on comments only.

12.4.5 Statistical Tests

The statistical tests used for evaluation purposes are as follows. First, we used an ordinary least square regression test [5] to investigate which metadata metrics (out of post lifetime, number of comments, and number of likes) was most accurate in predicting the number of interactions on posts. Secondly, the non-parametric Friedman test [6] was used to identify overall differences in the data since it is not normally distributed. Thirdly, a Nemenyi post-hoc test [6] was used to identify individual differences between metadata metrics. Forth, Cohen’s \(d\) [28] was used to quantifying the difference between means. Finally, all reporting of results includes standard measurements such as the test statistic, p-value, mean/median and standard deviation.

12.5 Conclusion

This work introduces the novel USMC method for efficient crawling of data from social network services by utilizing a wisdom of the crowd approach by allowing the users’ interactions to guide the crawler to find content to crawl. The evaluation of the proposed USMC method, through social network analysis [8, 29, 30], shows that it can cover in excess of 80% of the social network by collecting merely 30% of the available posts on a page. The social networks constructed from the collected data are shown to have close to identical degree distribution already at as low sampling sizes as
20% compared to the whole page, which indicates that the social network structure of the 20% sample is nearly identical to the complete page. That is, both the number of users and the number of social relations are similar, as well as the network degree distribution. This corroborates that the USMC method could be a powerful addition to the already available methods for crawling social network services when as many social interactions as possible are needed, while still maintaining a close to identical social network in terms of the network degree distribution.

**Supporting information**

The following are available online, Table S1: Statistics for each of the analyzed Facebook pages, Table S2: Interaction coverage for each of the analyzed Facebook pages, Figure S1: Ranking effect for each of the analyzed pages, Figure S2: Degree distribution for each of the analyzed Facebook pages using different sampling methods.

### 12.6 References


12. Do we really need to catch them all? A new User-guided Social Media Crawling method


12. Do we really need to catch them all? A new User-guided Social Media Crawling method


ABSTRACT

Online social media, such as Facebook, Twitter, and LinkedIn, provides users with services that enable them to interact both globally and instantly. The nature of social media interactions follows a constantly growing pattern that requires selection mechanisms to find and analyze interesting data. These interactions on social media can then be modeled into interaction networks, which enable network-based and graph-based methods to model and understand users’ behaviors on social media. These methods could also benefit the field of complex networks in terms of finding initial seeds in the information cascade model. This thesis aims to investigate how to efficiently collect user-generated content and interactions from online social media sites. A novel method for data collection that is using an exploratory research, which includes prototyping, is presented, as part of the research results in this thesis.

Analysis of social data requires data that covers all the interactions in a given domain, which has shown to be difficult to handle in previous work. An additional contribution from the research conducted is that a novel method of crawling that extracts all social interactions from Facebook is presented. Over the period of the last few years, we have collected 280 million posts from public pages on Facebook using this crawling method. The collected posts include 35 billion likes and 5 billion comments from 700 million users. The data collection is the largest research dataset of social interactions on Facebook, enabling further and more accurate research in the area of social network analysis.

With the extracted data, it is possible to illustrate interactions between different users that do not necessarily have to be connected. Methods using the same data to identify and cluster different opinions in online communities have also been developed and evaluated. Furthermore, a proposed method is used and validated for finding appropriate seeds for information cascade analyses, and identification of influential users. Based upon the conducted research, it appears that the data mining approach, association rule learning, can be used successfully in identifying influential users with high accuracy. In addition, the same method can also be used for identifying seeds in an information cascade setting, with no significant difference than other network-based methods. Finally, privacy-related consequences of posting online is an important area for users to consider. Therefore, mitigating privacy risks contributes to a secure environment and methods to protect user privacy are presented.