

# Protecting attributes and contents in online social networks

*Yuhao Yang*

Submitted to the graduate degree program in the Department of Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy

## Thesis Committee:

---

Dr. Bo Luo: Chairperson

---

Dr. Arvin Agah

---

Dr. Jun Huan

---

Dr. Prasad Kulkarni

---

Dr. Alfred Tat-Kei Ho

---

Date Defended

© 2014 Yuhao Yang

The Doctoral Committee for Yuhao Yang certifies  
That this is the approved version of the following thesis:

**Protecting attributes and contents in online social networks**

Committee:

---

Dr. Bo Luo: Chairperson

---

Dr. Arvin Agah

---

Dr. Jun Huan

---

Dr. Prasad Kulkarni

---

Dr. Alfred Tat-Kei Ho

---

Date Approved

# Abstract

With the extreme popularity of online social networks, security and privacy issues become critical. In particular, it is important to protect user privacy without preventing them from normal socialization. User privacy in the context of data publishing and structural re-identification attacks has been well studied. However, protection of attributes and data content was mostly neglected in the research community. While social network data is rarely published, billions of messages are shared in various social networks on a daily basis. Therefore, it is more important to protect attributes and textual content in social networks.

We first study the vulnerabilities of user attributes and contents, in particular, the identifiability of the users when the adversary learns a small piece of information about the target. We have presented two attribute-reidentification attacks that exploit information retrieval and web search techniques. We have shown that large portions of users with online presence are very identifiable, even with a small piece of seed information, and the seed information could be inaccurate.

To protect user attributes and content, we adopt the social circle model derived from the concepts of “privacy as user perception” and “information boundary”. Users will have different social circles, and share different information in different circles. We introduce a social circle discovery approach using multi-view clustering. We present our observations on the key features of social circles, including friendship links, content similarity and social interactions. We treat each feature as one view, and propose a one-side co-trained spectral clustering technique, which is tailored for the sparse nature of our data. We also propose two evaluation measurements. One is based on the quantitative measure of similarity ratio, while the other employs human evaluators to examine pairs of users, who are selected by the max-risk active evaluation approach.

We evaluate our approach on ego networks of twitter users, and present our clustering results. We also compare our proposed clustering technique with single-view clustering and original co-trained spectral clustering techniques. Our results show that multi-view clustering is more accurate for social circle detection; and our proposed approach gains significantly higher similarity ratio than the original multi-view clustering approach.

In addition, we build a proof-of-concept implementation of automatic circle detection and recommendation methods. For a user, the system will return its circle detection result from our proposed multi-view clustering technique, and the key words for each circle are also presented. Users can also enter a message they want to post, and the system will suggest which circle to disseminate the message.

# Contents

<b>Acceptance Page</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>7</b>
2.1 Privacy Threats during Communication . . . . .	8
2.2 Privacy Threats within Social Network Sites . . . . .	8
2.2.1 Private information disclosure . . . . .	8
2.2.2 Information aggregation attacks . . . . .	9
2.2.3 Inference attacks . . . . .	9
2.3 Privacy Threats in Published Social Network Data . . . . .	10
2.3.1 Attribute re-identification attacks . . . . .	10
2.3.2 Structural re-identification attacks . . . . .	10
2.4 Social Network Privacy Models . . . . .	11
2.5 Automatic Social Circle Detection . . . . .	13
<b>3 Content Vulnerability and Attribute-Reidentification Attacks</b>	<b>16</b>
3.1 Motivation and Overview of Attribute-Reidentification Attacks .	16
3.2 Information, Vulnerabilities and Attacks . . . . .	19
3.2.1 Information and Vulnerabilities . . . . .	19
3.2.2 Attacker Models . . . . .	20
3.3 Resourceful Attackers . . . . .	22
3.3.1 Data Collection . . . . .	23
3.3.2 Re-identification attacks . . . . .	27

3.3.3	Cross-database aggregation . . . . .	35
3.4	Tireless Attackers . . . . .	37
3.4.1	Tireless Attackers . . . . .	37
3.4.2	Smart Tireless Attackers . . . . .	40
3.5	Analysis and Reflection . . . . .	42
<b>4</b>	<b>Social Circle: Observations, Properties, and Automatic Detection</b>	<b>46</b>
4.1	Motivation . . . . .	46
4.1.1	Privacy Protection Using Social Circles . . . . .	49
4.1.2	Automatic Social Circle Detection and Content Distribution	51
4.2	Data Collection . . . . .	54
4.3	Structural-based Social Circle Detection Using SCAN . . . . .	56
4.3.1	SCAN . . . . .	56
4.3.2	Evaluation & Observations . . . . .	59
<b>5</b>	<b>Multi-View Clustering for Social Circle Detection</b>	<b>73</b>
5.1	Multi-View Clustering for Social Circle Detection: Motivation . .	73
5.2	Ego Network Modeling . . . . .	74
5.3	Multi-View Clustering . . . . .	80
5.3.1	Notations and Operators . . . . .	80
5.3.2	Co-trained Spectral Clustering: A Revisit . . . . .	80
5.3.3	Selective Co-trained Spectral Clustering . . . . .	82
<b>6</b>	<b>Evaluation of Multi-view Clustering for Social Circle Detection</b>	<b>86</b>
6.1	Evaluation Metrics . . . . .	86
6.2	Data Collection and View Construction . . . . .	88
6.3	Experiment Design . . . . .	90
6.4	Results and Performance Analysis . . . . .	92
6.5	Manual Evaluation . . . . .	95
6.6	Keyword Extraction for Clusters . . . . .	96
6.7	Discussions . . . . .	98
<b>7</b>	<b>Automatic Circle Recommendation and Proof-of-Concept Implemen- tation</b>	<b>101</b>
7.1	Automatic Circle Recommendation . . . . .	101

7.1.1	User-perception-based Circle Recommendation . . . . .	102
7.1.2	Attribute-based Circle Recommendation . . . . .	103
7.1.3	Content-based Circle Recommendation . . . . .	104
7.2	Implementation . . . . .	105
7.2.1	System Architecture . . . . .	105
7.2.2	Search Interface . . . . .	106
7.2.3	Automatic Circle Detection Result Presentation . . . . .	107
7.2.4	Automatic Message Designation . . . . .	111
<b>8</b>	<b>Conclusion</b>	<b>113</b>
	<b>References</b>	<b>116</b>

# List of Figures

3.1	Estimate the risk of <i>resourceful attacks</i> . . . . .	31
3.2	Cross-database aggregation for three cities. . . . .	35
3.3	Success rate of tireless attackers. . . . .	38
3.4	Results of successful <i>tireless attacks</i> with seed attribute tuple $\langle FN, AF \rangle$ . . . . .	39
3.5	Results of successful <i>tireless attacks</i> with seed attribute tuple $\langle FN, S \rangle$ . . . . .	40
3.6	Results of successful <i>smart tireless attacks</i> with seed attribute tuple $\langle FN, AF \rangle$ . . . . .	41
4.1	Social Circle Example . . . . .	50
4.2	Social Network Circle Creation Example . . . . .	52
4.3	Tag distribution of seed 15373743 . . . . .	67
5.1	Labeled Real World Online Social Network Subnet . . . . .	75
6.1	Normalized Similarity Ratio on Six Views. In each figure, the y-axis represents NSR and x-axis represents the number of update iterations. Blue dot curve represents <i>SC</i> approach, green dash curve represents <i>CSC</i> and red solid curve represents our <i>SCSC</i> approach. . . . .	93
6.2	Total similarity ratio (TSR) of all data sets. . . . .	94
6.3	Normalized Similarity Ratio of all Seed Users on Friend View. . . . .	94
7.1	System Architecture . . . . .	107
7.2	System Interface . . . . .	108
7.3	Name Suggestion . . . . .	108



7.4	Result Presentation . . . . .	109
7.5	Circle Detail . . . . .	109
7.6	Detailed Circle User Info . . . . .	110
7.7	Seed User Info . . . . .	111
7.8	User Input Message . . . . .	112
7.9	Designation Result . . . . .	112

# List of Tables

3.1	Seed attributes in the resource database created by a resourceful attacker. . . . .	24
3.2	Approximate information on attributes. . . . .	25
3.3	Information gain (IG) by knowing a single seed attribute with precise values. . . . .	33
3.4	Information gain (IG) by knowing seed attribute with approximate values. . . . .	34
3.5	Information gain (IG) by knowing multiple seed attributes. . . . .	35
4.1	Attributes of user profiles and tweets data. . . . .	55
4.2	Clustering result for seed 15373743 . . . . .	61
4.3	Clustering result for 7 seeds. . . . .	61
4.4	Circle Quality Evaluation Using TF-IDF for seed 15373743 . . . . .	64
4.5	Circle Quality Evaluation Using TF-IDF for 7 randomly selected seeds . . . . .	65
4.6	Circle Quality Evaluation Using Tags for seed 15373743 . . . . .	68
4.7	Representative tags for clusters of seed 15373743 . . . . .	69
4.8	Interaction-based Circle Quality Evaluation for 7 randomly selected seeds . . . . .	71
6.1	Attributes of user profiles and tweets data. . . . .	89
6.2	Sparse Degree of Six Views. The SD is calculated as the ratio between number of zeros in a matrix and the number of elements in that matrix. . . . .	90
6.3	Size of Each Cluster in the Clustering Result. <i>std</i> stands for standard deviation of all group sizes. . . . .	93

6.4 Representative tags for clusters of a seed . . . . . 97

# Chapter 1

## Introduction

With the development of internet technologies, people start to carry out more and more activities online, as online shopping, online studying, online banking, etc. One of the most significant creation in the Web era is online social network, which tries to connect users socially online. Social networks' first appearance online can trace back to early 90's last century, and it starts to gain vast popularity from mid 00's. During the last few years, online social network becomes more and more import to people's social life with the development of Facebook, YouTube, Google+ and many other web sites. They are becoming extremely popular, attracting huge amounts of users and Internet traffic. For instance, Facebook recorded one billion active user accounts in late 2012, while approximately 10 million messages are posted every hour. They have significantly changed our information sharing and socialization behavior, especially among the younger generation – it has been reported that 48% percent of Facebook users between 18-34 years old check Facebook when they wake up<sup>1</sup>. With online social networks, users can find their old friends without con-

---

<sup>1</sup><http://www.statisticbrain.com/facebook-statistics/>

tacting for many years, find out how is every friend every day, what is the hot topic recently, and also get to know new friends much more easily. Online social networks provide a place where people can share, interact, and explore in a more unconstrained manner.

However, as the online social life becomes more and more colourful these days, it also causes some privacy issues. The extreme popularity of online social networks has become a double-edged sword. While service providers devote to promote online socialization, privacy issues arise. In the literature, studies have shown a massive disconnection between users' privacy perceptions and their behavior—widely known as the *privacy paradox*. That is, most users do not take appropriate actions to protect their information, although they express concerns on the privacy of such information [10, 75, 110]. For instance, many users are concerned about their location privacy [16, 45, 67], however, a blog/micro-blog post about a local restaurant [79], or blogs with location-indicating words such as “Time Square” [21, 24] could effectively reveal the user's location. Many users ignore privacy settings. They make their profiles and other contents visible publicly. Even though their private attributes are only accessible to a portion of users, attackers can still infer their personal information [56]. In addition, the maturity of other information technologies, as information retrieval, make adversaries easier to collect and analyze information online. This puts online social network users' privacy into risk, as will be shown in Chapter 3.

After the privacy problems originate, the research community have done many works trying to study the characteristics of possible attacks, and also find reasonable solutions for eliminating impacts of the attacks. Most of the exist-

ing works focus on data-publishing and structural-based attacks or protections. Some time, online social network data sets are published for legitimate reasons. Researchers have been working on how adversaries could utilize the published data to obtain sensitive information and how to modify the data sets to make users more secure. Many of the proposed attack and prevention models are based on links among users, which can be represented as graphs with nodes corresponding to users and edges corresponding to links. Attackers can make use of the connectivity information to possibly get the private information of the target [55, 56, 142, 154]. The solutions try to modify this published graph so that attackers can hardly succeed. One direction is to apply anonymization theories, like k-anonymity [131]. The basic idea is to change the graph so that it satisfies some anonymization properties, like each node has the same structural characteristics with at least k-1 other nodes [83].

Although structures are important for user privacy protection in online social networks, it is content that actually bears sensitive information. Even if the attackers successfully re-identified the target in the published anonymized data set, without the contents and attributes information (i.e. user attribute data), they can still hardly get the private stuff, and the users are still safe. In addition, social network data is rarely published, and billions of messages are sent in different social networks in a short period of time. Therefore, it is more important to protect attributes and content in online social networks.

In social science, privacy is more related to user perception, i.e. they are secure if they feel secure. So the proposed protection mechanisms for user privacy are more about how to control the information flow and set its boundary. In our research, we propose to combine social science theories with modern

computer technologies to protect user's privacy automatically and effectively.

Although as stated above, there have been plenty of works studying the privacy issues in online social networks using structural knowledge, protection focusing on attributes and content was mostly neglected. In our research, we first study the vulnerability of content on the internet, which we will state in detail in Chapter 3. We construct two types of attackers: resourceful and tireless. We assume that they both have a piece of information of the target as the seed, and try to re-identify the target using some methods. The former attackers are able to create resource database as knowledge base and re-identify the target within it, and can also combine multiple resources to get more sensitive information of the target. The later do not have such resources. The only tools they have are the ordinary online search engines and their own time and energy. They search the web, inputting the seed information as the query, and examine the results to check whether they can re-identify the target. By designing some experiments, we have shown that large portions of users with online presence are very identifiable, even with a small piece of seed information, and the seed information could be inaccurate.

To protect user content and attributes, we propose to incorporate the social circle model derived from social science theories. In this model, there are circles, i.e. highly clustered groups of people, in social networks. Circles have boundaries, which can be utilized to constrain information flow and therefore, protect user privacy. The notions of social circles and information boundary have been proposed, to protect private information and to facilitate secure socialization. However, the problem of social circle discovery remains open and challenging. We propose an automatic social circle detection mechanism uti-

lizing multi-view clustering, which is based on the method proposed in [68]. In particular, we propose a one-side co-trained spectral clustering technique, which is tailored for the sparse nature of our data. For the multi-view clustering approach, we start with our observations that users belonging to the same circle are very likely to: (1) be friends and share many common friends; (2) be interested in similar content; (3) have more interactions with each other. We model the ego network with 6 different views, and we argue that features from different views would complement each other. We tested our algorithms with real-world social networking data collected from Twitter and compared it with several other clustering techniques, as structural-based clustering, in particular SCAN, proposed in [144], and single-view spectral clustering. Experiment results show that our approach is both effective and efficient. Multi-view clustering is more accurate for social circle detection; and our proposed approach gains significantly higher similarity ratio than the original multi-view clustering approach.

The contributions of this dissertation are three-fold: (1) We take a first step towards studying private information online, especially the online social networks data. We intensively examine the vulnerability of private information in online sources as well as the validity of different types of attribute-based privacy attacks. In particular, we introduce an information-theory-based approach to evaluate the values of personal information items to the attackers; (2) We are the first to integrate structural, content and interaction features to identify social circles in online social networks. We introduce a novel selective co-trained spectral clustering method to better handle view inconsistency and view sparsity. We implement and evaluate our methods against real-world



social networking data, and demonstrate the superior performance of the proposed approaches; (3) We build an automatic social circle detection and suggestion proof-of-concept implementation, using multiple state-of-the-art web techniques, as JSP, JavaScript, JQuery, etc., which is both innovative and user-friendly.

In the following dissertation, background and related works are mentioned in Chapter 2. Threats to online attributes and content are stated in detail in Chapter 3. Social circle model motivation and circle discovery is talked in Chapter 4. In Chapter 5 and 6, we will present our multi-view clustering method thoroughly, and in Chapter 7 we will introduce our proof-of-concept implementation of automatic circle detection and recommendation. Finally, we will make a conclusion in Chapter 8.

# Chapter 2

## Related Work

In recent years, many online social network services (SNS) such as Facebook and Google+ have become extremely popular, attracting huge number of users. As their popularity grows, more users willingly put their (personal) information to social network sites so that they can share them with other users. With the advancement of information retrieval and search engine techniques, on the other hand, it becomes easier to do web-scale extraction of users' personal information that is readily available in various social networks (e.g., [3,9,20,71,108]). Therefore, malicious or curious users could take advantage of these techniques to collect others' private information [76,147]. We have been overwhelmed by news reports on social network privacy: threats, tragedies, and public concerns [19,35,52,60,63,65,98,112]. Unfortunately, so far, such concerns have not been sufficiently answered by research community and IT industry.

## **2.1 Privacy Threats during Communication.**

Communications are quite common in online activities. It greatly helps web users' in all kinds of areas. However, due to its underlying implementation, it will also create possibilities for attackers to infringe users' security of privacy. During Web browsing, users implicitly reveal their private information (e.g., IP address) through network communications. Anonymous communications are proposed to hide user identities in the Internet [33,50,111]. Meanwhile, privacy issues connected with ubiquitous social computing are investigated [26,58,97,115].

## **2.2 Privacy Threats within Social Network Sites**

### **2.2.1 Private information disclosure**

Although, private information disclosure is usually controlled carefully, it may be mistakenly disclosed from trusted social networks in some situations: publicly-available archives of closed social networks [39], social network stalkers [35], code errors, add-ons and apps [19,60,66]. Meanwhile, users may voluntarily give out private information: people publicize private information if they feel "somewhat typical or positively atypical compared to the target group" [59]; 80% of the Facebook users adopt identifiable or semi-identifiable profile photos, and less than 2% made use of the privacy settings [53]. In addition, recent user studies show that users' privacy settings violate their sharing intentions [85,90], and they are unable or unwilling to fix the errors [90]. [72,117] studies the discrepancies between users' perceptions on their privacy disclosure and the exposure allowed by conventional privacy policies (espe-

cially to atypical access patterns). Furthermore, the study of [91] explores and classifies three types of private information (e.g. vacation plans, medical conditions) shared in the textual content of tweet messages. On the other hand, users may post messages and later regret doing so, for various reasons [139]. Also, advanced techniques, as impersonation attacks, have been proposed [13] to steal private (friends-only) attributes by cloning or faking user identities.

### **2.2.2 Information aggregation attacks**

Information aggregation attacks are introduced in [76,87,147]: online social network users voluntarily release pieces of personal information, e.g., profile attributes and blog posts. Significant amount of privacy is recovered when such pieces are associated and integrated. In particular, people are highly identifiable with very little information [51, 104, 130], which make cross-network aggregations quite feasible and dangerous. A large scale experiment in [8] confirms that a significant amount of user profiles from multiple social networks could be linked by email addresses.

### **2.2.3 Inference attacks**

In this type of attacks, the private information is obtained by the attackers using some inference rules. In [55,56,154], hidden attributes of target users are inferred from friends' attributes with a Bayesian network. Recently, [96] shows that unknown user attributes could be accurately inferred when as few as 20% of the users are known. On the other hand, friendship links and group membership information can be used to (uniquely) identify users [142] or infer sensitive hidden attributes [154], e.g., membership of a local engineer society discloses

user's location and profession. Most of the inference attacks' inference rules are based on target users' explicit social connections (i.e. direct friends and social groups explicitly constructed by different online social network applications). However, the inference based on implicit friendship relations and social groups are rarely touched. In our work, we will propose mechanisms for discovering the implicit structures and privacy protection based on them.

## **2.3 Privacy Threats in Published Social Network Data**

### **2.3.1 Attribute re-identification attacks**

When social network data sets are published for legitimate reasons, user identity and personal information are removed. However, with combination of some pieces of un-identifiers attackers is possible to successfully re-identify the targets and compromise their privacies. Some of the well-known techniques for defending against these types of attacks include *k-anonymity* [4, 132], *l-diversity* [89] and *t-closeness* [78]. The basic ideas of them are to make the attributes, which can probably identify the target, as usual as possible within the published data, and also make the sensitive information of users having same values in these attributes as diverse and distant as possible.

### **2.3.2 Structural re-identification attacks**

Graph structure from anonymized social network data could be utilized for re-identification. A good survey about this type of attacks could be found at [156]. Notably, [6] identifies the problem that node identities could be inferred through passive and active attacks, where passive attacks try to re-identify the

node based on the original structure of the social network when possessing some structural or topological information. To evaluate the vulnerability of social networks in the structural re-identification attacks, *topological anonymity* is proposed. *Topological anonymity* quantifies the level of anonymity using the topology properties of network graph [122]. An adversary who has the knowledge of user's neighbors could re-identify the user from network graph [155]. One property of graphs that are capable to survive under such assaults is *K-Degree Anonymity*. *K-Degree Anonymity* requires each node to have the same degree with at least  $k - 1$  other nodes [83]. In addition, [54] models three types of adversary knowledge that could be used to re-identify vertexes from an anonymized social network graph. On the other hand, [84] handles social network as a weighted graph, in which edge labels are also considered sensitive.

## 2.4 Social Network Privacy Models

One of the most important privacy models is  $k$ -anonymity [132]. It inherits the privacy definition of "un-identifiability." In this model, when a record (user profile) could not be identified from  $k$  other records, it is considered private.  $K$ -anonymity and its successors [78,89] are good for anonymized data publishing, but not online interactions. Reputation based models such as [1] alerts users when they provide information to untrusted are better for real-time protections. In another direction, the Platform for Privacy Preferences (P3P) [17,27,29] introduces a standard for machine-readable privacy notices, and uses agents to advise users [28]. However, P3P only helps to establish a "contract" between users and web sites, but does not guarantee proper enforcement.

Recently, [14] and [151] build social networks from multiple resources while ensuring the privacy of participants. [7] introduces a social network platform that stores and exchanges encrypted content, and access is enforced through key management. [121] builds a platform that enforces privacy control on third party applications. On the other hand, with the observation that it is difficult to explicitly define access control for large number of friends (hundreds to thousands), tools have been built to help users manage their privacy settings. For instance, *Privacy Wizards* [40] builds a machine learning model to predict and configure users' privacy rules based on limited input, and *PViz* [92] is proposed to help users comprehend their privacy configurations based on the automatically labeled groups. [127] predicts privacy policies for newly uploaded images based on their content similarities with existing images with known policies. Other approaches [61, 143, 148, 150] help users group their contacts, by exploiting the topology relationships among the users' friends. Finally, audience visualization tools [18] are also proposed to help users perceive and control their disclosure boundary. However, none of the above mentioned approaches prevents privacy leakage during normal socialization (e.g. unwanted private information disclosure and cross-site information aggregation attacks as described in Section 2.2). By and large, some of them lack theoretical foundations from sociological/psychological perspectives, while others do not have formal constructs. Solutions need to find a balance between the privacy and usability. In our work, we will propose a protection mechanism based on social circles, which is effective for controlling user privacy and at the same time, user friendly.

## 2.5 Automatic Social Circle Detection

Identifying social circles from a user's online social networks is important for the individual to exert appropriate access control on information sharing [31, 124]. However, manually managing groups on social network sites might present a burden for users [62, 73], which triggered the idea on using automatic sociocentric network clustering algorithms. The feasibility of this idea has been demonstrated in the findings from [48, 62].

Sociocentric network clustering, which is usually referred to as community detection, aims to divide people into groups within which they are more similar [2] and have more connections [101] or relationships. Unlike traditional personal network studies, which focus on attribute-based data such as age, sex [94], most of graph-based methods in community detection [44], which include traditional methods like graph partitioning [64] and hierarchical clustering [49, 101], modern methods through maximization of a likelihood like [57, 100], and more recent methods based on matrix factorization like [109, 152], only consider topological structure and linkage information. But there is a trend in recent research based on graphs which combined link information and content or attribute information [81, 113, 146] or interaction information between individuals [159].

Compared with graph-based methods, another class of approaches attach greater importance to content or link context information. [22, 86, 157, 158] use state-of-the-art method like topic modeling to take full advantage of semantic information, such as email, tweet messages, and documents, in detecting communities from a social network. [138] proposed a method to find like-minded people who share more semantically relevant tags.



A recent research which is more related to ours is [114]. It proposed generative Bayesian models to utilize not only topics and social graph topology but also nature of user interactions to discover latent communities in social graphs. The difference between their work and ours is that we also used tag annotation method to analyze content information generated by users, which concerns the understanding of the information and is more meaningful in finding similar topics.

Based on privacy concerns and automatic social circle detection, [93] developed a model to discover social circles by using both network structure and user profile information; [125] proposed an approach based on apriori algorithm to identify hidden groups by dynamically detecting grouping criteria, i.e. certain combinations of properties of a user's contacts, such as relationship, location, hobbies, age, privacy, etc. The difficulty in utilizing this kind of methods is that automatically collecting attributes of users through online social network is a nontrivial task although traditional personal network studies can collect these information through interviews more easily.

On the algorithm aspect, we employ the multi-view clustering framework for social circle detection. This framework frees us from the difficult task of manually merging multiple sources before clustering. More importantly, empirically study has shown its superior performance [12,68,69,137]. In our application, we only have access to the similarity matrix, not the feature matrix, we base our algorithm on the co-trained spectral clustering [68] approach, which works directly on user similarity.

Spectral clustering [119] is a single-view clustering technique that exploits properties of the Laplacian of the graph, whose nodes are samples and edges

are similarities between samples. The top  $k$  eigenvectors of the normalized graph Laplacian are shown to contain discriminative information for the  $k$  clusters. Hence the algorithm performs  $k$ -means on rows of the eigenvectors to obtain the clustering result.

Co-trained spectral clustering (CSC) is an extension of spectral clustering to the multi-view setting based on the idea of co-training [15], where samples are described by multiple feature sets, a.k.a. views. The underlying assumption of CSC is view consistency, that is, if two samples are in the same cluster in one view, they should be in the same cluster in all other views. CSC has the advantage of further restricting the hypothesis space (the space of possible functions that map each sample to a cluster) by imposing consistency constraint across views.

A limitation of CSC is the equal treatment of each view, whereas in our application, interaction views are too sparse to be completely consistent with other views. Therefore, in this dissertation we tailor *CSC* and propose the selective co-trained spectral clustering (*SCSC*) algorithm. Our key idea is to encourage clustering results in a sparse view to be transferred only if the corresponding similarity is not zero. Experimental results show that *SCSC* not only effectively boosts the performance while many views are sparse, but also has the advantage of efficient convergence rate.

# Chapter 3

## Content Vulnerability and Attribute-Reidentification Attacks

### 3.1 Motivation and Overview of Attribute-Reidentification Attacks

The Internet has changed the ways we publish, search and consume information. Even with the static web, a huge amount of personal-related content has been made available online. More recently, various types of online social network (OSN) products have been introduced to the Internet, which further promotes the sharing of personal information. In addition to the great commercial success and social impacts of the OSNs, they also brought new challenges to the research community (e.g., [46, 74, 82]). With enormous number of users and tremendous amount of personal information available over various online social networks, it is critical to ensure that user privacy is well preserved. However, although many researchers have been working on extracting information or learning knowledge from online social networks, very little research effort

has been put so far into the study of security and privacy issues until very recently [6, 54–56, 83, 87, 149, 153–155].

In online social networks, users voluntarily share personal information within the community under some implicit assumptions that: (1) this information is only accessible to the targeted readers; (2) one's true identity cannot be discovered if he/she only provides limited/incomplete profile information (e.g. an email address and a phone number); (3) a small amount of information is not significant and the disclosure will not hurt one's privacy; and (4) it is very difficult, if not impossible, to collect and link pieces of information scattered over various online social networks or data sets, and associate them to one's real identity. Unfortunately, these assumptions are proven to be either false or at least questionable, in both research literatures and news reports. Several types of privacy attacks in social networks have been proposed, such as the structural re-identification attacks [6, 54, 83, 149, 153, 155], the inference attacks [55, 56, 154], the information aggregation attacks [76, 87], and the traditional attribute re-identification attacks [53, 59]. Although different types of attacks and countermeasures have been proposed in recent literature, only a few of them have been well tested on real data. Moreover, most of the attacks and corresponding protection mechanisms are based on the graph topologies of social networks. Privacy attacks that focus on the attributes are not well studied.

Personal information is scattered over various sources, including online social networks and the general Web. We believe a thorough understanding of the nature of how these information are distributed and retrievable is the key to an effective defense. Our work takes a first step towards studying private information online, especially the online social networks data. In this chapter, we

intensively examine the vulnerability of private information in online sources as well as the validity of different types of attribute-based privacy attacks. In particular, we define two types of attackers, *resourceful attacker* and *tireless attacker*, based on their different attack capabilities and strategies. Both types of attackers obtain small amounts of information about their targets, known as seed attributes, from external sources and launch advanced re-identification attacks. The seed information could be non-identifiable attributes, such as names of schools where the target gets degrees. A resourceful attacker is capable of retrieving a large amount of personal information about potential targets from online social network sites and creating his/her own resource database, and re-identifies the target by checking the seed attributes against his/her resource database. On the other hand, a tireless attacker only submits such attributes to search engines, and tirelessly browses and studies the results for clues. We have simulated both types of attacks on our database, with 3 million records collected from an online social network and a phonebook data set, to check their reality and severeness. From the results, we can see that large portions of users with online presence are identifiable even with a small piece of seed information, where the seed information could be inaccurate. Our simulation also shows that it does not require extensive resources or efforts to successfully conduct attribute-based attacks to hurt user's privacy online.

## 3.2 Information, Vulnerabilities and Attacks

### 3.2.1 Information and Vulnerabilities

With the Internet explosion, huge amounts of information have been made online. Moreover, advances in information retrieval techniques and Web search engines have enabled easy access to such information. However, large amount of personal information is also exposed to public, not always with the consent of the information owner. In particular, we believe there are three primary channels for personal information disclosure:

**Personal information on the general web.** In the Web 1.0 era, especially in the early days, personal homepages sometimes contain large amount of personal information. Such information is usually published by owners who are somewhat familiar with the Web. They usually understand the risks better than the novices, hence, the contents may be carefully tailored to protect privacy. On the other hand, some personal information maybe published in sources such as news, employee directories, etc. Overall, this channel is better administered although sensitive information could be disclosed by careless users.

**Digitalized public records.** With governmental and industrial efforts, a large amount of public records (e.g. phone books) have been digitalized and made available online. Many of them are indexed by commercial search engines, while others require a minimum subscription fee for full access – the barrier is usually low for an adversary to query or even collect the entire databases. Some public information could be highly personal (e.g. salaries of faculty members in public universities).

**Online social networks.** As online social networks get extremely popular, they

become gold mine for adversaries. Large volume of personal information have been collected at social network sites for socialization, career development, and other purposes. As shown in [53], most social network users are poorly protected and their personal information is highly accessible. In this way, social network users may be very vulnerable.

All types of information summarized above are accessible to adversaries, who strive to collect personal information about the targeted users. From the adversaries' perspective, user information could be categorized as (i) private information, (ii) identifiable information, and (iii) non-identifiable information. In the literature, a lot of work has been done on the risks associated with (i) and (ii), and on preventing (i) private information from being disclosed to the Internet. However, seed information obtained by the adversary (from offline) is not always identifiable, hence, the attacker's first objective is to discover the true identity of the target (i.e. from category iii to ii).

### 3.2.2 Attacker Models

In this work, we define and simulate two types of attackers, *resourceful attackers* and *tireless attackers*, with different attacking capabilities and strategies.

**Resourceful attacker:** a *resourceful attacker* is assumed to have enough resource (bandwidth, storage, technique, etc) to construct his/her own database by collecting information from the Web. The database could be constructed in three ways: (1) crawling the general web, extracting personal information from web pages, and storing the data in a local database; (2) implementing a focused crawler to collect data from online public record datasets; (3) crawling online social networks, or downloading research data sets published by social net-

work sites.

In the information retrieval community, many work has been done for entity extraction from the “surface Web”, e.g. [23]. However, to populate a local database requires intensive crawling of a significant portion of the surface web, which is very time-consuming. Comparably, collecting information from public records and online social network user profiles is more feasible since the information has been concentrated on such websites. Moreover, considering the user data are usually published in well-structured templates, resourceful attackers can easily implement niche parsers to extract structured personal information. One practical obstacle could be the restriction for massive crawling, which usually violates the terms of use for most online social networks. However, with technical assists, e.g. anonymous routing [33], such crawling is very doable and hard to detect. As such, it is reasonable for us to assume a resourceful attacker has certain technical capability to crawl from typical online data sources.

With the collected databases, resourceful attackers compare his/her external knowledge about the target with information in the database, and search for candidate records for further examination. Meanwhile, if the target is identified from one database, it becomes trivial to use the discovered identity to retrieve more information from other databases. A real-world example for cross-database attacks is given in [76]. The example in [76] is a manually executed attack, but the risk is valid when a resourceful attacker possesses multiple overlapping databases.

**Tireless attacker:** a *tireless attacker* does not have the resources or techniques to create and maintain a local database. As a compensation, a tireless attacker



devotes more time and labor in the attacking process to maximize the chance of success. In particular, a tireless attacker knows some of the attributes of his/her target (seed attributes), and submits such attributes to search engines, and tirelessly browsing and examining the results for clues. Due to the size of the Web, the results returned from search engines are mostly noise, and the attacker needs to be very patient to discover any useful information. The chance of success highly depends on the amount of information provided by the seed attributes. For instance, if the attacker knows that the target get a Bachelor's degree from a large public university and nothing else, it is very unlikely to identify the target in tireless attack. However, if the attacker knows the first name of the target, and the fact that he/she gets a Ph.D. from a small university, the attacker is more likely to discover the true identity (full name) and more personal information of the target.

Meanwhile, a tireless attacker also tries to search or browse in social networks, public records, etc. Furthermore, besides the "brute-force" attack, a tireless attacker can get "smarter" by constructing advanced queries with his/her knowledge about the attacker. For instance, if the attacker knows that the target is currently employed at a university, it is more likely that the target's information will be discovered from webpages within the domain of the university. This type of "advanced search" functions are provided by all major search engines.

### **3.3 Resourceful Attackers**

In this section, we focus on two types of privacy attacks (i.e. the re-identification attack and the cross-database aggregation attack) conducted by a resourceful

attacker, who is capable of maintaining a private database of a large volume of publicly available online user profiles. In our study, we simulate the power of the resourceful attackers by crawling user data from two publicly available resources, a social networking site and an online phone book data repository, and study the feasibility, difficulty, and the success rate of resourceful attacks with different types of seed attributes.

### 3.3.1 Data Collection

To implement a proof-of-concept attacking mechanism, we design niche crawlers to collect data from two resources to simulate the proposed resourceful attacker.

#### 3.3.1.1 Collecting data from LinkedIn

LinkedIn<sup>1</sup> is a professional online social networking site that provides open access to detailed identifying user profiles. We implemented a specialized crawler to retrieve data based on the public index of LinkedIn.com, using methods and technologies that are available to any potential resourceful attacker. We collected approximately 9 million (8,943,014) user profiles in total in 10 months. The crawled html profiles are indexed alphabetically by the last name of profile owners and stored in a MySQL database for further offline processing, which includes two major procedures, *data extraction* and *data cleaning*.

**Data extraction:** The LinkedIn profile contains rich information about one's educational history that is useful in identifying a target. However, the raw data are in html profiles, which need to be extracted to reconstruct corresponding

---

<sup>1</sup><http://www.linkedin.com>

records in the resource database. We implemented a specialized parser to do that. Currently, our parser only extracts data from three fields *name*, *work* and *education* fields, which contain the most useful information for re-identification. In the future, we consider to extend our parser to include more fields such as working experiences. Data from the three fields are further processed and categorized into 11 **seed attributes**, as shown in Table 3.1. For instance, data in *name* field are segmented as first, middle and last names. Data in *work* field are decomposed to *current title*, *affiliation* (e.g. Software Engineer at XYZ company), *industry* type (e.g. Internet, Higher Education, Research), and current *location* (e.g. San Francisco Bay Area). Similarly, *school name*, *degree* earned, *major*, *degree starting time* and *ending time*, and the entire degree *time period* are extracted from *education* field. Please note that one profile may have multiple education records. Also, not all the profiles contain education information. In some profiles, the education field is either left blank or hidden from non-registered LinkedIn users.

**Table 3.1.** Seed attributes in the resource database created by a resourceful attacker.

<b>Name</b>		<b>Work</b>		<b>Education</b>	
<i>FN</i>	first name	<i>TI</i>	title	<i>S</i>	school name
<i>LN</i>	last name	<i>AF</i>	affiliation	<i>D</i>	degree
		<i>IND</i>	industry	<i>ST</i>	start time
		<i>LO</i>	location	<i>ET</i>	end time
				<i>T</i>	time period

To simulate the attacks where the attacker only had approximate information about the target, we consider two common scenarios. In the first case, the attacker only knows the initial of the target, and in the second case, the attacker only know the approximate location of the school that the target has

attended. Therefore, we added 5 new attributes to our seed attribute table, as shown in Table 3.2. After cross-checking with the school reference lists, we have successfully added country and continent information to all schools and state information to all the US schools, for 80% of the records.

**Table 3.2.** Approximate information on attributes.

Attribute	Approximation	Notation
name	initials	<i>N.in</i>
school	state	<i>S.st</i>
	region	<i>S.re</i>
	country	<i>S.ct</i>
	continent	<i>S.cn</i>

**Data cleaning:** The collected data may have redundant or ambiguous contents, which makes data cleaning operations important in data collection. Some of the ambiguity is caused by inaccurate or wrong inputs of careless users, for instance, a user mistakenly includes department name or year of graduation as part of school name. A more common problem resulting in redundant content is that many universities are referred by different names. For example, we noticed University of Cambridge is referred as Cambridge University instead of its formal name in some profiles. We corrected this problem by cross-checking with the school reference lists that contain formal forms for most of the schools. After a quick browsing of the data, we created manually coded heuristic rules to map most of the school names to their formal forms, and removed all the redundant elements and special characters <sup>2</sup>.

Another important processing we took in data cleaning is to set aside the schools with less than 3 attendees, which we think are highly likely to be invalid

---

<sup>2</sup>Due to lack of referencing lists for high schools and lower level schools, we have to remove all education records at high school level or lower.

or mistaken entries (proved by later manual check). After all the operations, we successfully obtained about 2,466,721 clean profiles, with 3,417,550 clean education records.

### 3.3.1.2 Collecting data from online phone books

Many data sets with private personal information are now publicly available for commercial or administration purposes. Such information is open to public, unless data owners explicitly opt out. Residential phone book data is such a resource, which has been made online through various sources. All the online phone book sites list phone numbers and residential locations for free access. For the registered users, more detailed residential information are also available. Moreover, a few of online phone books even show the names and addresses of the holders as unlisted phone book entries, for instance, while the phone numbers are hidden, the owners' info is displayed on <http://www.phonesbook.com>.

We assume the resourceful attackers are capable of retrieving all types of online data to enrich their resource databases. Therefore, we crawled residential phone book entries for three regions, two college towns and one state capital city<sup>3</sup>, from an online phone book data repository to simulate attackers' knowledge in this category.

After creating his own resource database, the resourceful attacker is capable of launching two types of attacks, the *re-identification attack* and the *cross-network attack*.

---

<sup>3</sup>City names are anonymized as required by double-blind review policy.

### 3.3.2 Re-identification attacks

The re-identification attack is to explore the identity (and/or other information-of-interest) of the target by linking or matching the known information about the target to the data in the resource database. In this section, we first simulate a number of re-identification attacks over the crawled LinkedIn data to assess the risk of re-identification attack against profile data that users voluntarily submitted to online social networks. Then, we employ an information theory based approach to theoretically estimate the re-identification risk.

#### 3.3.2.1 Re-identification attack model

To launch a re-identification attack, the attacker needs to know some information about the target. It is assumed that the attacker obtains such knowledge from external resources. When the attacker obtains offline information about the target, he expresses this knowledge in the form of *seed attributes* that he collects for the resource database.

The attacker's knowledge about each target varies. In some cases, the attacker knows only one seed attribute about the target, e.g. "John has a bachelor's degree". In other cases, the attacker may know more about the target, which can be interpreted as multiple seed attributes, e.g. "John graduated from college in 2004". Sometimes, the knowledge about the target is not accurate. For example, the attacker may only know that "John graduated from a school in Midwest". Since the inaccuracy in the name and school location fields are addressed by the new approximate attributes in Table 3.2, we can simulate certain inaccurate inputs in attacker's knowledge. For instance, the attacker may know that: "John graduated from a school in Midwest", which indicates

Attribute *SchoolRegion* = “Midwest”.

Therefore, we model the attacker’s knowledge about a target as an identity-attribute tuple  $\langle I, v_1, \dots, v_t \rangle$ , where  $I$  is the identity of the target, and  $\{v_1, \dots, v_t\}$  are the values of the known seed attributes  $\{A_1, \dots, A_t\}$ . For instance, the attacker’s knowledge “John graduated from college in 2004” can be expressed as:

Identity :  $I = John$

Attribute *FirstName* :  $v_1 = “John”$

Attribute *EndTime* :  $v_2 = “2004”$

In the defined resourceful attack model, to re-identify the target, the attacker needs to send the known identity-attribute tuple into the resource database that is built upon the data retrieved from online sources. The severeness of such re-identification attack highly depends on the completeness and identifiability of the records in the resource database. Therefore, the first-step approach towards assessing the risk of such re-identification attack is to study the resource database. In particular, we explore the *identifiability* of the crawled LinkedIn user profiles in our simulated resource database to assess the re-identification risk.

### 3.3.2.2 Assessing risk with profile identifiability

The resource database and the seed information are two key components for a successful re-identification. Consider a resource database  $\mathbb{D}$  with  $n$  records, where each record is associated with one identity. To an attacker, the ideal case for the resource database is that it is large enough to contain records of all the

targets and each record contains all information about the target. The ideal case for the seed information is that it is accurate and adequate to distinguish the target from records of others in the database. However, it is very difficult, if not impossible, to meet both conditions in real-world cases. Therefore, for a resourceful attacker, it is important to measure the identifiability of the records in the resource database  $\mathbb{D}$ .

**Definition 1.** For a database  $\mathbb{D}$  whose scheme is  $\mathbb{D}(A_1, \dots, A_t)$ , we define the **identifiability of a target  $T$  in  $\mathbb{D}$**  as  $I_T^{\{v_1, \dots, v_r\}} = k$ , if  $T$  cannot be distinguished from other  $k-1$  profiles with known seed attributes  $\{attr_1, \dots, attr_r\} = \{v_1, \dots, v_r\}$ , where  $r \leq t$ .

This definition is similar to the *k-anonymity* concept of privacy in data publishing, but interpreted from the attacker's perspective. For each target whose record is in  $\mathbb{D}$ , given any adequate and accurate seed information  $\{v_1, v_2, \dots, v_t\}$ , his/her identifiability should be 1, which means he/she is uniquely identified. Typically, since the attacker's seed information is limited, the identifiability of a target,  $k$ , is much larger than 1. However, for the attacker, the size of potential profile set (that may contain the target) under this definition is successfully decreased from  $n$  to  $k$ .

To assess the **identifiability of  $\mathbb{D}$** , we further count  $n_{k-}$ , which is the number of profiles that cannot be identified from at most  $k$  other profiles given seed information  $\{attr_1, \dots, attr_r\}$ . In other words, for every possible value set  $\{v_1, \dots, v_r\}$  in the seed attribute tuple space  $\mathbb{R}^r$ ,

$$n_{k-} = \text{sum}(I_T^{\{v_1, \dots, v_r\}}), \text{ for } I_T^{\{v_1, \dots, v_r\}} < k.$$



Then, we calculate  $k$ -or-less proportion  $p(k)$  as an indicator of the identifiability of  $\mathbb{D}$ , where

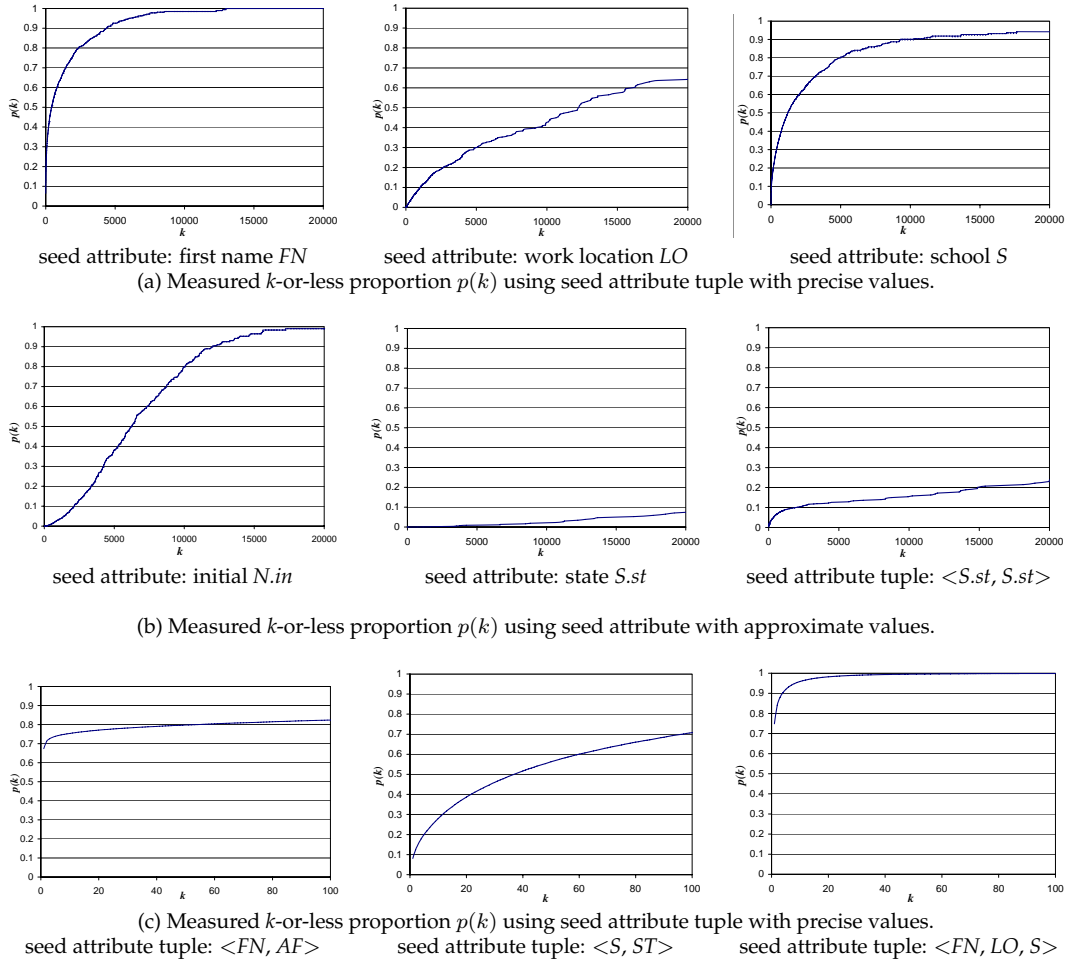
$$p(k) = \frac{n_{k-}}{n}, \text{ for } k \in [1, \max(k)],$$

and  $\max(k)$  is the largest  $k$  for all possible values of seed attribute tuple  $\{attr_1, \dots, attr_r\}$ .

Next, we select several seed attribute tuples, and assess the identifiability of the resource database with crawled LinkedIn data. First, we simulate the scenario where the attacker only knows a single seed attribute value about the target. Then we measured the  $k$ -or-less proportion  $p(k)$  for each seed attribute in Table 3.1. The results of three seed attributes, first name  $FN$ , work location  $LO$ , and school name  $S$ , are shown in Figure 3.1(a). In the figure, a slowly growing curve indicates better anonymity, since less people are identifiable among smaller sets. As we can see from the figure, users' identifiability shows different patterns for different attribute. Overall, when the adversary only knows one attribute, most people cannot be identified among a relatively large set.

Then, we consider the scenario in which a weaker attacker only knows approximate values of the attributes, as summarized in Table 3.2. Some of the results are shown in Figure 3.1(b), when the attacker knows (i) the first and last initials ( $N.in$ ) of the target, but not the name, e.g. the attacker knows "JD", not "John Doe", or (ii) the region where the target goes to school ( $S.re$ ), e.g., "the person went to school in West coast". As we have expected, knowing approximate values on an attribute usually gives the adversary very limited information.

The third type of scenarios that we examine is that the resourceful attacker knows multiple attributes about the target. Figure 3.1(c) shows the population vs.  $k$ -anonymity curves when the resourceful attacker knows (i) first name and



**Figure 3.1.** Estimate the risk of *resourceful attacks*.

affiliation:  $\langle FN, AF \rangle$ , e.g. “John works at XYZ company”; (ii) school name and starting time:  $\langle S, ST \rangle$ , e.g. “the person went to Stanford in 2001”; and (iii) first name, work location and school name  $\langle FN, LO, S \rangle$ , e.g. “John went to Berkeley, and now works at New York”. Note that the  $k$  axis is scaled to  $[1, 100]$ . As we can see from the figure, users become very vulnerable when the adversary knows multiple seed attributes.

We also consider the case where the adversary knows approximate information on multiple attributes. The right-most figure in Figure 3.1(b) shows the population vs.  $k$ -anonymity curve when the attacker knows the states in which

the target goes to school (given that the target goes to at least two schools), e.g. “the person went to school in California and Massachusetts”. Obviously, the database is less identifiable under approximate seed information.

### 3.3.2.3 Assessing risk using information gain

To quantify the *amount of information* provided by an attribute, we further analyze the problem from an information theory perspective. In our scenario, the goal of the attacker is to identify the particular record which corresponds to the target. Without any prior knowledge, all the records are equally likely to be the target. Hence, to achieve the goal, the average amount of information that the attacker needs to collect (i.e. adversary’s expected information gain) is denoted as:

$$E(I(X)) = H(X) = -\log_2 \frac{1}{N}$$

where  $N$  is the number of records in the database. In our simulation,  $E(I(X)) = 21.23(\text{bits})$ , i.e. on average, the attacker needs to obtain *21.23 bits* of information in order to identify a target from our database.

When the attacker knows the value  $v$  of attribute  $attr$ , the conditional entropy is denoted as:

$$H(X|attr = v) = -\log_2 \frac{1}{N_{attr=v}}$$

where  $N_{attr=v}$  is the number of records that satisfy the condition `attr=v`. On

**Table 3.3.** Information gain (IG) by knowing a single seed attribute with precise values.

Category	Attribute	IG (bit)
Name	<i>FN</i>	13.348
	<i>LN</i>	16.461
Work	<i>TI</i>	14.433
	<i>AF</i>	12.979
	<i>IND</i>	6.405
	<i>LO</i>	8.011
Education	<i>S</i>	11.8231
	<i>D</i>	1.8336
	<i>ST</i>	5.149
	<i>ET</i>	5.026
	<i>T</i>	7.537

average, the information gain of knowing attribute  $A$  is denoted as:

$$\begin{aligned}
 I(X; A) &= H(X) - H(X|A) \\
 &= H(X) - \sum_{v \in V_A} p(A = v) H(X|A = v)
 \end{aligned}$$

where  $H(X|A)$  is the conditional entropy of knowing attribute  $A$ . In our settings, an information gain of  $m$  bits indicates that the attacker has successfully discovered that the target is among  $\frac{N}{2^m} = \frac{2,466,721}{2^m}$  records, on average. In addition, the attacker will need to further obtain  $21.23 - m$  bits of information in order to exactly identify the target. Most importantly, if we assume that our data set is a random sample of the general population, attackers' information gain will be the same if he obtains the same attribute in the general population. In that case,  $H(X)$  and  $H(X|A)$  increases proportionally, while  $I(X; A)$  will remain the same (statistically).

In Table 3.3, we show the information gain of the attacker when he/she knows one seed attribute. As we expected, the last name carries the largest

amount of information, while first name and school name also carries significant amount of information. However, knowing one attribute alone is not enough for the attacker to identify the target, or to narrow down to a very manageable range. Attribute *ln* is somehow an exception, which on average narrows the search to less than 30 candidates (i.e.  $H(X|A) = 4.77bit$ ). When the attacker only know approximate information on an attribute, the information he/she learns from the knowledge is even less, as shown in Table 3.4.

**Table 3.4.** Information gain (IG) by knowing seed attribute with approximate values.

Attribute	IG (bit)	Attribute	IG (bit)
<i>N.in</i>	8.807	<i>S.st</i>	4.795
<i>S.re</i>	2.360	<i>S.ct</i>	1.853
<i>S.cn</i>	1.328		

In the scenario that the attacker knows multiple attributes, the information gain is denoted as:

$$I(X; A_1A_2) = H(X) - H(X|A_1A_2)$$

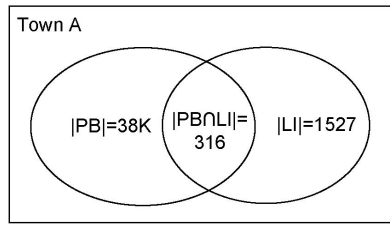
When two attributes  $A_1$  and  $A_2$  are independent, we should have:

$$\begin{aligned} I(X; A_1A_2) &= H(X) - H(X|A_1A_2) \\ &= H(X) - H(X|A_1) - H(X|A_2) \end{aligned}$$

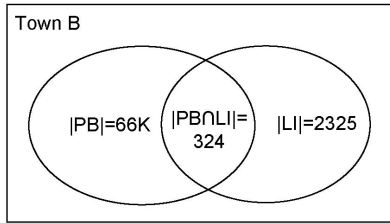
Table 3.5 shows the information gain when the attacker knows multiple attributes.

**Table 3.5.** Information gain (IG) by knowing multiple seed attributes.

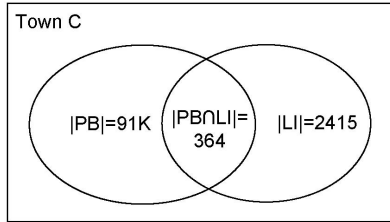
Attributes	IG (bit)	Attributes	IG (bit)
$\langle FN, S \rangle$	20.316	$\langle S, ST, ET \rangle$	16.549
$\langle FN, S.st \rangle$	15.068	$\langle FN, S.ct \rangle$	12.848
$\langle FN, ST \rangle$	16.092	$\langle FN, ST, ET \rangle$	17.362
$\langle FN, ET \rangle$	15.679	$\langle D, ST, ET \rangle$	5.685



(a)



(b)



(c)

**Figure 3.2.** Cross-database aggregation for three cities.

### 3.3.3 Cross-database aggregation

As we have introduced, a resourceful attacker is capable of collecting multiple databases from different sources. When the attacker identifies the target (i.e. discovers the full name of the target) from one of the databases, it becomes trivial to retrieve relevant records from other databases to learn more about the

target.

In our experiments, we simulate cross-database aggregation attack by matching LinkedIn data with online phone book data. We have crawled phone book data for three cities: two college towns and a state capital city. We try to link records from both databases by matching full names. The results are shown in Figure 3.2. As we can see, approximately 20% of the LinkedIn users from town A could be identified in phone book, while 14% and 14% of the LinkedIn users from town B and town C are re-identified, respectively. According to the literature [51,76], with known full name and location information, people are very identifiable. We are confident that most of the linked records are true positives (i.e., the two linked records reflect one unique offline identity). For linked records, the attacker will further learn the home address and phone number of the user. In many cases, the attacker also learns the names of the family members of the user.

In cross-database aggregation attacks, when a resourceful attacker identifies a target using attribute-reidentification attacks on one of his databases, it is likely that he can learn more information about the target. In our experiments, we only collected information about users whose phone numbers are listed. As we have mentioned, there are websites (e.g. <http://www.phonesbook.com/>) that publish addresses of users who opt to exclude their information from the phone book. From our observation, this website contains 20% more user records than the phone book data set we crawled. Meanwhile, with a small fee, the attacker could subscribe to various databases that collect personal information from public and commercial records. Therefore, a resourceful attacker has great potential to become more powerful than we have demonstrated in this

work.

We can also see that the phonebook size is much larger in state capital C, which shows that a relatively larger population who do not have LinkedIn accounts (or configured their accounts as private), but are still visible in the phone book. In this case, although these users are not actively releasing their information online, or are successfully protecting their online identities, unfortunately, their personal information is still accessible from online sources.

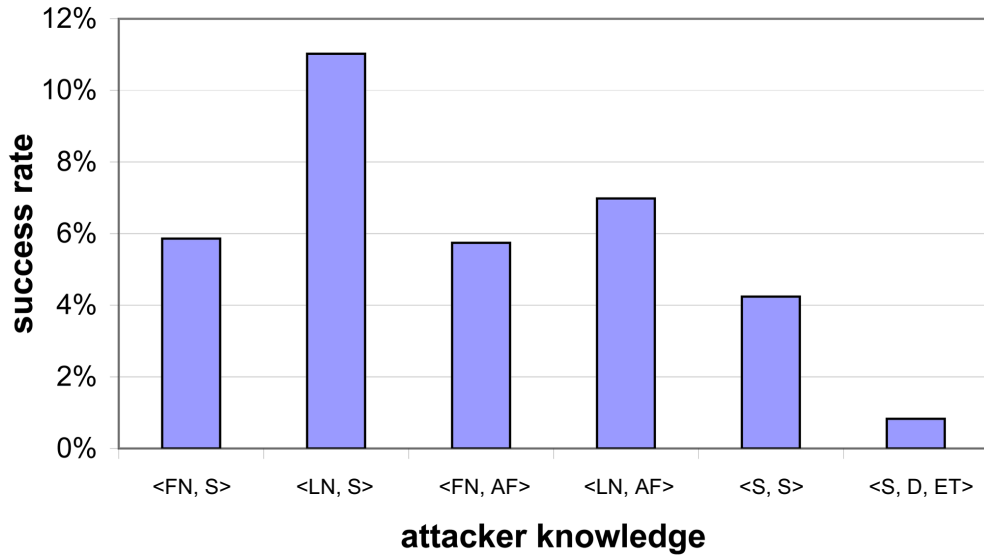
## 3.4 Tireless Attackers

### 3.4.1 Tireless Attackers

Tireless attackers do not possess a local database of personal information, as a compensation, they devote their time and energy. In our simulation, the tireless attacker knows some (non-identifiable) attributes about the target. The attacker queries a Web search engine (we use Google in our experiments) with the known attributes, and examines the results returned by the search engine for any clue.

To simulate tireless attacks, we have randomly sampled 50,000 users from education and healthcare industry, including faculty, students, researchers, doctors, etc. We simulated tireless attacks on different combinations of known attributes. In Figure 3.3, we show the success rate when the tireless attacker knows the target's: (1) first name and the name of last school that the target attended  $\langle FN, S \rangle$ ; (2) last name and school  $\langle LN, S \rangle$ ; (3) first name and current affiliation  $\langle FN, AF \rangle$ ; (4) last name and current affiliation  $\langle LN, AF \rangle$ ; (5) names of two schools, knowing that the target has attended two or more schools  $\langle S,$

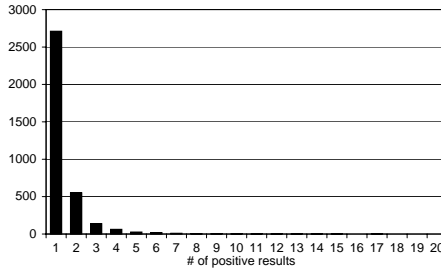




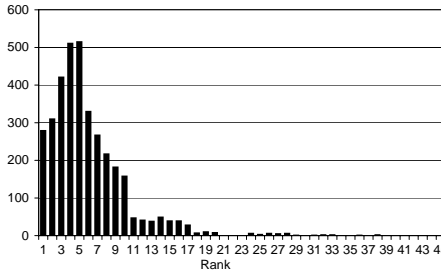
**Figure 3.3.** Success rate of tireless attackers.

$S$ >; and (6) school name, degree and year of graduation  $\langle S, D, ET \rangle$ . When the full name of the target was discovered in a returned web page in the form of “John Doe” or “Doe, John”, we treat the result as *positive*. An attack is successful when at least one positive result is found in the top 200 results returned from the search engine. Please note that in tireless attacks, we exclude all the results from LinkedIn, i.e., an attack is successful only if the target is re-identified from non-LinkedIn sources.

Figure 3.4 and Figure 3.5 give more insights on tireless attacks. Figure 3.4(a) shows a histogram of the number of positive results for successful attacks when the attacker knows the first name and affiliation of the target  $\langle FN, AF \rangle$ . Figure 3.5(a) shows the same histogram for  $\langle FN, S \rangle$  case. We do observe a significant portion of targets who have been re-identified from multiple websites (excluding LinkedIn). Meanwhile, no victim has been re-identified from more than 20 websites. Figure 3.4(b) and Figure 3.5(b) show a histogram of the rank



(a) Number of positive results in each successful attack.

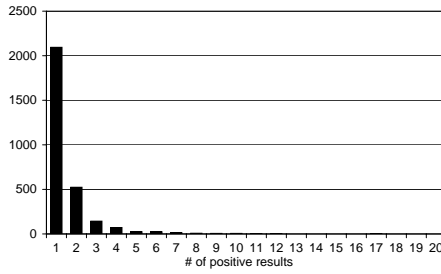


(b) Rank of the first positive result in each successful attack.

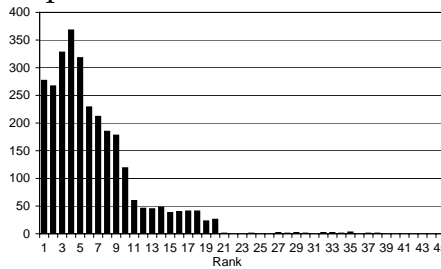
**Figure 3.4.** Results of successful *tireless attacks* with seed attribute tuple  $\langle FN, AF \rangle$ .

of the first positive result for successful attacks of the  $\langle FN, AF \rangle$  and  $\langle FN, S \rangle$  cases, respectively. We observe that most of the positive results came from top 10 results, which indicates that a tireless attacker does not need to be very “tireless” to achieve a successful attack. On the other hand, we also observe that positive results do not always come in top 2 search results.

To further validate the successful attacks, we have manually checked 200 randomly-sampled positive results for each type of attacks. We have discovered that around 70% of them were true positives that also contain further personal information about the target. Meanwhile, we do have some false negatives. For instance, in the  $\langle LN, AF \rangle$  attack, we have found a few pages of conference program committee members. They contain the name of the school, and a person with exactly the same name as the target, but affiliated with a different school or organization. Another major category of false positive ap-



(a) Number of positive results in each successful attack.



(b) Rank of the first positive result in each successful attack.

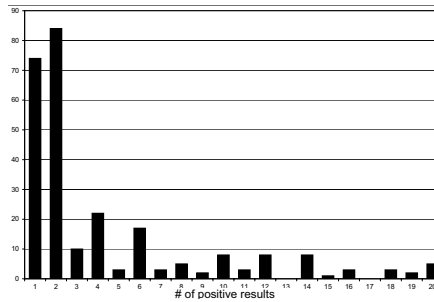
**Figure 3.5.** Results of successful *tireless attacks* with seed attribute tuple  $\langle FN, S \rangle$ .

appears when the name “Doe, John” is discovered in the context of “Jay Doe, John Smith”.

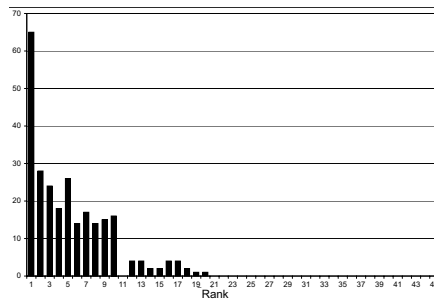
Last but not least, when the targets are identified in *tireless attacks*, we continue the attack by issuing new queries using their identity (i.e. full name) and known attributes. For most of the cases, we can easily discover more sources (again, excluding LinkedIn) that contains further information about the target. A major reason is that we use the LinkedIn user profiles from education and healthcare domains as seeds, and such users are more active on the Internet.

### 3.4.2 Smart Tireless Attackers

Regular *tireless attackers* use a simple textual combination of all the known attributes as the query to be sent to search engines. However, existing web search engines support not only free text queries, but also advanced queries



(a) Number of positive results in each successful attack



(b) Rank of the first positive result in each successful attack

**Figure 3.6.** Results of successful *smart tireless attacks* with seed attribute tuple  $\langle FN, AF \rangle$

(e.g. Google Advanced Search<sup>4</sup>). Tireless attackers can get smarter by utilizing such functions. In our simulation, when the tireless attacker knows the affiliation of the target (e.g. this person works at XYZ University), it is highly likely that information about the target could be found in the employers’ domain (e.g. xyz.edu). A smart attacker first queries the search engine (e.g. Q=“XYZ university”) to get the official website of the employer, which is usually included in the top 3 returned results. The attacker then issues an advanced query, which contains textual terms and a domain constrain. The textual terms include the other known attributes about the target (e.g. first name “John”), while the domain constrain forces to search within the employer’s domain (e.g. “site:xyz.edu”).

<sup>4</sup>[http://www.google.com/advanced\\_search](http://www.google.com/advanced_search)

We simulate smart tireless attacks for the case with seed attribute tuple  $\langle FN, AF \rangle$  (i.e., the attacker knows the first name and current affiliation of the target). We have simulated attacks for 10,000 users, randomly sampled from the 50,000 records that we used for regular tireless attacks. Figure 3.6 shows the simulation results of such smart tireless attacks. As we can see, the re-identification rate of smart tireless attacks is lower than the re-identification rate of regular tireless attacks. It means that, in at least 50% of the successful regular tireless attacks, the targets are identified from information sources other than websites of their workplaces. When we further look into the successful smart tireless attacks, we observe that most of them are true positives. Moreover, as we can see from Figure 3.6(b), on average, the rank of the first positive result is higher in smart tireless attacks. Therefore, smart attacks are more effective – less effort is required for the attacker to browse and examine the results. For both regular and smart tireless attacks, we can see that most of the users are either identified in top results, or never identified. It means that when the user is not “highly visible”, his/her information is most likely buried in the massive amount of online information and becomes invisible. However, consider the fact that only a small portion of the general population have disclosed their information on the Web, people with an online presence is still highly distinguishable.

### 3.5 Analysis and Reflection

**Information.** We have observed a large amount of personal information available over the Internet. Each information item may include both identifiable and non-identifiable attributes. Not all such information is published by the owner (of the identity), or with the consent of the owner. For instance, we have ob-

served webpages such as news stories published by the employer. Moreover, the user might be completely unaware that his/her information has been accessible and searchable over the Internet. From the simulation results, we can see that it is very difficult, if not impossible, to completely hide one's online identity in the Internet age.

On the other hand, we have introduced an information-theory-based approach to evaluate the values of personal information items to the attackers. We believe that the results will help users determine the types and amounts of information to be published on personal and social networking sites.

**Vulnerability.** We have simulated the data collection process of resourceful attackers. We can see that personal information could be easily collected by attackers, especially from social networking or public record sites, where information is published in well-structured templates. On the other hand, automatically and accurately extracting *large amounts* of structured information from free (unstructured) text is not an easy task. Named-entity extraction [23,38] is a very hard problem. Although we have seen successes in controlled datasets or for popular entities that appear on many web contexts, the general problem of arbitrary entity extraction is still far from being solved. In particular, diversity of web documents and limited evidences (e.g. a user's phone number only appears on one webpage) make it very difficult to precisely extract and collect large amounts of entities from the web. However, we have shown that it takes little effort for a human attacker to exploit search engines to locate webpages containing such information.

Next, with the simulation results of resourceful and tireless attackers, we have shown that **people with web presences are highly identifiable**, even with

very limited or approximate information. Moreover, information from multiple resources could be linked to provide more information to the attacker. A major reason behind the phenomenon is that many people do not have a web presence, as confirmed by our cross-network aggregation attacks. On the contrary, people with web presence are very likely to appear in multiple sources. In this sense, we have a group of people who are more active on the Web, while the mass majority of the population mostly remain silent online. As a result, the online population becomes very identifiable. There appears to be a dilemma: if we have more people online, the identity of the existing users will be better “shadowed” than they are right now. However, in this way, we may put more people under risk.

**Attacks.** Recent advances in information retrieval techniques are shown to be a double-bladed sword – they provide great functions to the users, but also reveal their private information to attackers with sufficient capabilities and resources, or strong wills. Intuitively, we can interpret the goal of the attacker as taking a piece of seed information as input against large data that are available online to successfully find a hit.

Ideally, if the seed is precise and adequate and the data is large enough to guarantee that it contains the target, the attack will always succeed. While the results are constrained in reality, the attacker manages to increase his chance and efficiency by meeting the conditions at his most. The first (and often hidden) assumption is that the focused data should be large enough to contain data of a particular target. In the resourceful attack, the focused data is the resource database created by the attacker, which in turn motivates the long-term and multi-source data collection. The second condition that affects the

success rate of the attack is the identifiability of the user with the seed information (in terms of either seed attributes or search terms). In the tireless attack, it is assumed implicitly that the related data should be in the high-rank results returned by search engines. This in turn explains why tireless attack is only effective when the target is highly distinct against proper search terms (or combination of search terms). The study of the identifiability will also shed light on how to tailor one's online presence to shadow his identity within an indistinguishable group.



## Chapter 4

# Social Circle: Observations, Properties, and Automatic Detection

After analyzing the vulnerability of online content and attributes, we will move on to propose our solutions for this problem. In this chapter, we will talk about the motivation of social circles, elaborate the data collection process, and introduce a graph topology based social circle detection technique called SCAN.

### 4.1 Motivation

Offline social networks start to appear as early as the origin of human beings. At that time, social circles emerge due to the geographical separation and difficulty of transportation. The creation of languages makes communication between people much easier, and human intelligence changed the world magnificently. The development of transportation makes the distance “shorter”, paper and books make sharing of knowledge easier, and instruments, mu-

sic and all kinds of sports make people's life much more colorful. Although the development makes people connection dramatically easier, the diversity of modern society makes people tend to form more circles. As the development of human society, different types of works arise. People with different jobs will more possibly form different groups, so as people with different education background, interests, locations, religions, etc. Modern social science research about circles can trace back to late 19th century [88]. Social circles are groups of people that have strong connectivity or higher similarity. They can be used to study different aspects of social networks, including information flow [105], epidemiology [103], security and privacy [41,61,141], etc. In traditional social science, circles are often studied with real people involvement, such as interviews, or questionnaires. Although the development of statistics makes the traditional methods more powerful and scalable, it is still very time-consuming and not suitable for large-scale analysis.

After the innovation of computer technologies, the traditional study of social circles have been largely assisted. The relation of people can be represented by matrices, stored in computers, and analyzed using all sorts of inter computer algorithms. Among them, graph clustering methods have been well developed and applied to inter fields [77,99,116,118,135,140,145], which makes circle detection possibly automatic. When the web applications become more and more popular, in particular, online social networks, sociology has entered a new era. For many people, it has even become a habit to log into social network websites everyday. This has greatly facilitated social network studies, since large amount of social data becomes available. However, new challenges have also appeared. Firstly, because of the vast amount of users online, the social net-

work can be very large, which propose computational issues; Secondly, there are more than one types of relations between user online, as they can become friends, send messages, comment or share other's content, etc, which makes the conventional techniques for one-dimensional graphs not enough, and more advanced methods need to be applied; Thirdly, more information available means more vulnerable people are, as can be seen from our previous work talked in chapter 3, which results in urgent requirement for more efficient and effective security and privacy mechanisms.

In this new age of online social networks, social circles still have their important role. Discovering circle membership can help infer common attributes among group members, as hobbies, location, jobs, etc, and can also be utilized for friends recommendation, personalized advertisement, information flow studies, and so on. More importantly, since it draws boundaries implicitly for social networks, it is well suited for security and privacy enforcement. The idea of using social circles for privacy control can retrospect to former social science studies, which propose the concept of "privacy as user perception". Controlling the visibility of inter information based on inter social circles can make users feel secure. Additionally, most of the previous works focus on circle detection and analysis for the whole network in both social and computer science, which are not only very inefficient but also not suitable for real implementation, where information is sent from user to user and personalized circle privacy enforcement is more appropriate. There are some pioneer research in this field: in [95], the author studies personal networks from the perspective of social science, which shows there are indeed inter groups in people's personal networks representing inter parts of people's lives; in [61], the authors

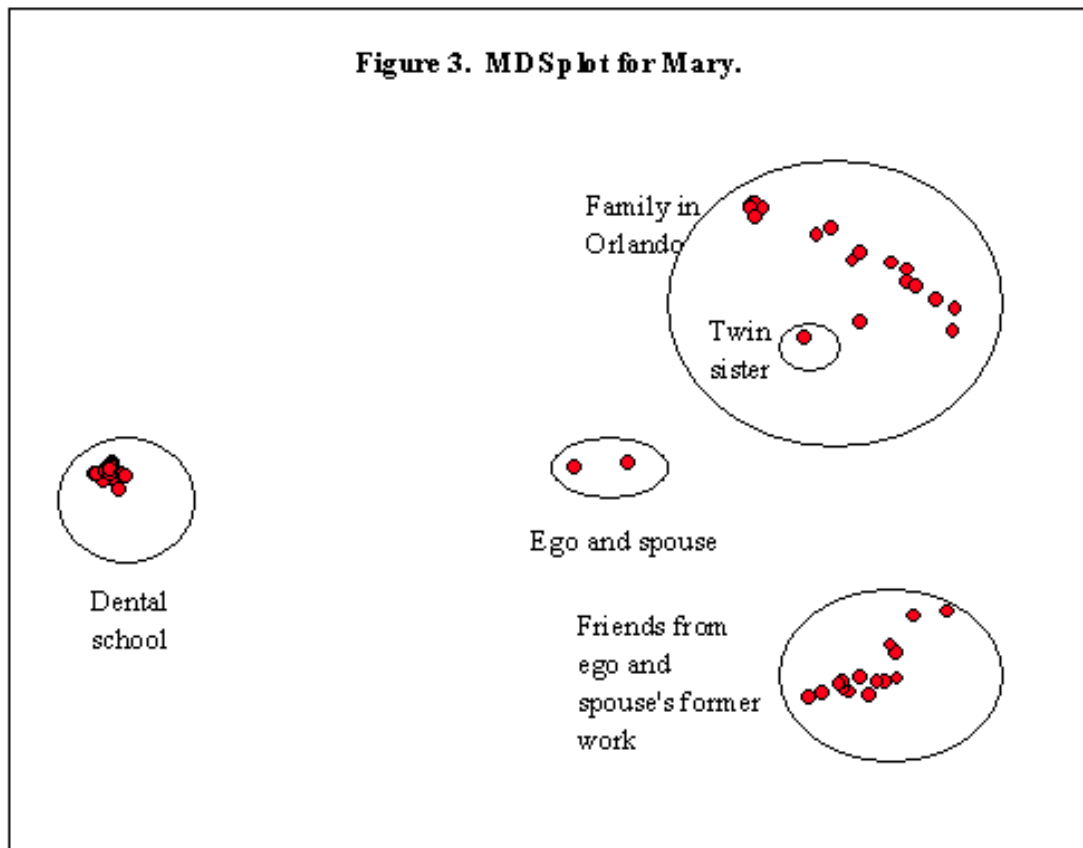
have similar ideas as what we want to propose in this work. They want to use some clustering technique to automatically partition user's personal network and implement group-based privacy. However, they also choose real human interviews and questionnaires as their evaluation methods, and their analysis about group-based privacy is more about the practicability, no automatic privacy policy formulation. In addition, although they consider the content impacts on human beings' privacy policy making, they do not combine it in clustering or make use of it for machine implemented privacy mechanisms.

In conclusion, social circles are an important aspect of online social networks, and particularly, we want to utilize them for security and privacy issues of online users. With the help of advanced graph clustering techniques, we propose an automatic social circle detection and recommendation system.

#### **4.1.1 Privacy Protection Using Social Circles**

The user-centered privacy and HCI research community has introduced the notion of "*restricted access and limited control*" [30, 136] and "*information boundaries*" [120]. In particular, social circles have been proposed for privacy protection [125, 126], so that new messages are posted to designated social circles and the message owners have full control of the information boundary. Meanwhile, social circles are also expected to promote information sharing, since they give users the perception of security and privacy. Social circles provide natural boundaries to disseminate information. As an example from [88], shown in Figure 4.1, we can see that the "Ego" Mary, whom we are studying the personal network about, has the closest circle with her spouse. She also has a family circle who are in Orlando and a special connection with her twin

sister inside the family. Another 2 social circles she have are for her and her husband's former work and her current dental school classmates or teachers. This is a very common case of an ordinary person's social structure.



**Figure 4.1.** Social Circle Example

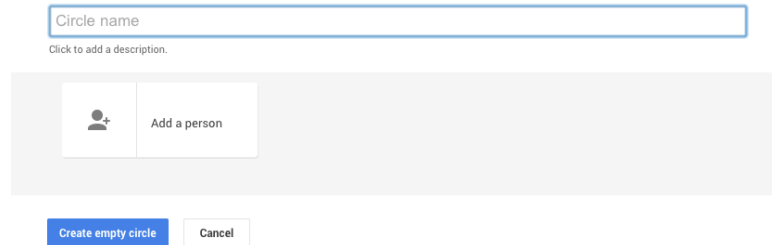
From this example, we can see how social circles can help to protect user privacy. When Mary wants to talk about her children, relations with her husband or some other personal issues, she may prefer not to let her former colleagues or current classmates know, and she wants her family members to discuss these problems with her. But when she wants to post some information about her former work or dental knowledge, the other two circles are more appropriate for distributing the information. Or sometimes, she wants to complain about her

former boss or some colleagues, the family or dental school circle should be the destination. Another scenario is when there are many soccer fans in her former colleague circle, she would prefer to post some messages about World Cup to them. After successfully dividing users' personal networks into appropriate social circles, we can easily control information within certain boundaries to effectively protect users' privacy. Social circles provide natural "information districts", when detected properly, can be utilized to significantly improve users' social experience online. Users nowadays tend to have different aspects of social life, serving for different purposes. Social circles can present these differences and help to protect users' privacy in a truthful and user-friendly way. However, industrial adoption has not been very successful. Various products have been released by commercial social networking sites, such as circles in Google+ and custom lists in Facebook. However, none of them is well-received by users. A major drawback is the usability problem – it is tedious and labor-intensive to assign hundreds of existing friends into circles or lists. We will state more about this in the next section.

#### **4.1.2 Automatic Social Circle Detection and Content Distribution**

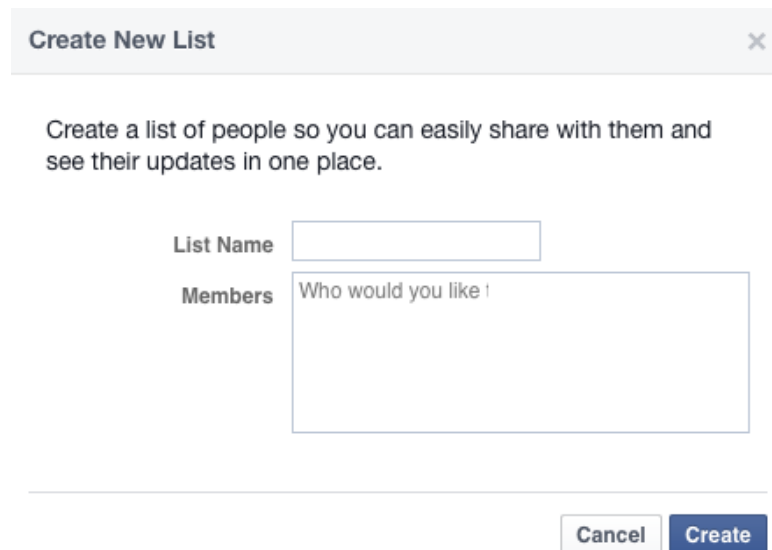
Based on the above statement, to detect hidden social circles properly becomes significantly important. How to discover social circles easily and correctly should be the main concern. One approach is by expert manually detection. Experts are knowledgeable people who are experienced in the social community discovery field. But we may lack such resource and even experts cannot know every person very well to detect social circles exactly as they should be. Another method is to ask users to label social circles themselves. Most online

social network sites apply this method (see Figure 4.2 for the facebook and Google+ example.)



The screenshot shows the Google+ circle creation interface. At the top is a text input field labeled "Circle name". Below it is a link that says "Click to add a description.". In the center is a large grey button with a person icon and the text "Add a person". At the bottom are two buttons: "Create empty circle" (highlighted in blue) and "Cancel".

(a) Google+ Example



The screenshot shows the Facebook "Create New List" interface. At the top is a header "Create New List" with a close button (X). Below is the instruction: "Create a list of people so you can easily share with them and see their updates in one place." There are two input fields: "List Name" and "Members". The "Members" field has the placeholder text "Who would you like?". At the bottom are two buttons: "Cancel" and "Create" (highlighted in blue).

(b) Facebook Example

**Figure 4.2.** Social Network Circle Creation Example

Even though, this method is easier to implement and usually more accurate, it is very labor-intensive and not practical enough to be widely adopted. Users tend to create one large circle of "Friend" and ignore this functionality. As a result, we want to propose an automatic social circle detection technique, which can not only ease users' work, but also render relatively accurate results. Users do not need to inconveniently create social circles by themselves, and at the

same time, can get acceptable quality clustering results. In the following parts of the dissertation, we will mainly talk about automatic social circle detection, including a purely structure-based method and a proposed selective co-trained spectral clustering method. In the rest of this chapter, we will firstly talk about the method based on friendship only, which is also called the structure-based method. Our goal is to find an appropriate computer algorithm, which can help us detect social circles both accurately and efficiently. Based on the characteristics of modern online social networks, we propose to use the multi-view clustering technique to include not only the structural aspect of social networks, but also their content and interaction information. The detailed elaboration can be seen in Chapter 5, where we will present our experiment results, from which we can see the significant improvement of circle quality by our method.

In addition, in the goal of better protecting user privacy, which most users are indifferent of, we want to automatically suggest most appropriate circles, when users want to post some information online. The idea is: even though the social circles are defined properly, if users refuse to utilize them for privacy, the issues are still unsolved. As an example, for Mary in last section, even if all the circles are defined correctly for her, if she makes complains about her former boss visible to her colleague circle, and personal issues visible to the colleague and classmate circles, there would be some problems. So we also want to ease this part of users' online social life, and provide a more secure environment. For this functionality, we will discuss in more details in Chapter 7.



## 4.2 Data Collection

To fulfill our goal of circle based user privacy analysis and implementation, we construct our data set from **Twitter**<sup>1</sup>. Twitter is one of the most newly developed online social network service, which allows users to edit personal profiles, follow other users, upload photos and “tweet” or “re-tweet” about everything they want to share. It is mostly recognizable for its “tweets” function, which can be viewed as micro-blogging. For its convenience, it rapidly gains popularity and hundreds of millions of users can generate over 300 million tweets and billions of search queries per day. Because of its vast disseminated information and popularity, proposing efficient and effective privacy protection applications is significantly important.

We construct our crawler using **Twitter4j**<sup>2</sup>, which is a Java library integrating newest **Twitter API**<sup>3</sup>. It allows users to get a Twitter user’s profile, follower/following list, and tweets given its identifier. To conduct the crawling, we start from a manually selected random user as the seed, which has enough friends for continuous crawling. After crawling one seed, we will select one friend of the seed, who have least common friends with the seed and with number of friends no less than 100, no larger than 500. We do this to make seeds as less correlated as possible, and omit those users who has too many or too few friends. We believe that users with too many friends are often well-known people or special users, as twitter accounts for companies, advertisement, magazines, etc. Users with too few friends are often inactive, for which they can be omitted for privacy settings, and they are also not suitable for the following

---

<sup>1</sup><http://www.twitter.com>

<sup>2</sup><http://twitter4j.org>

<sup>3</sup><https://dev.twitter.com/docs>

seed selection.

For each seed, we construct a folder, containing a text file for its profile information, a text file for its most recent tweets (no more than 2,000), and a file of all its friend ids. It is worth noting that we consider a user that both follows the seed and is followed by the seed as a friend of the seed, due to the easy follow mechanism of Twitter, which may induce noise to our experiment. For each friend of the seed, we also create a folder containing the similar profile, tweets, and friend list files, and put it within the seed folder. The detailed attribute information for profile and tweet data is shown in Table 4.1. The *user ID* and *tweet ID* are number strings, generated by Twitter. Our friend list files contain lists of *user IDs*.

**Table 4.1.** Attributes of user profiles and tweets data.

<b>Profile</b>		<b>Tweet</b>	
<i>Name</i>	user name	<i>ID</i>	tweet id
<i>Screen name</i>	displayed screen name	<i>Create Time</i>	tweet made time
<i>ID</i>	user id	<i>Location</i>	created location
<i>Created at</i>	create time	<i>In reply to</i>	replied user id
<i>Description</i>	personal statement	<i>Content</i>	tweet content
<i># of followers</i>	number of followers		
<i># of following</i>	number of followings		
<i>Location</i>	user location		
<i>Timezone</i>	user timezone		
<i>URL</i>	personal url		

In about 5 months, we collect 160 seeds. The average number of friends for all the seeds is: 249, and the average number of tweets for them is: 932.

### 4.3 Structural-based Social Circle Detection Using SCAN

In this section, we will discuss the structural-based clustering method we use for automatic circle detection. The method is proposed in [144], which is called SCAN (Structural Clustering Algorithm for Networks). A similar work we mentioned [61] also used this method. Firstly, we will give a brief introduction of this method including the related work and step by step algorithm elaboration. Then we will present our evaluation techniques for its clustering result and observations obtained.

#### 4.3.1 SCAN

Our circle detection is largely related to the field of graph clustering, which is motivated in physics, computer science, applied mathematics, etc. Graph clustering is trying to group vertices of graphs so that vertices within clusters are more connected than vertices between different clusters. Before SCAN, there have already been several other types of graph clustering methods. As mentioned in a survey of graph clustering [116], there are hierarchical methods, including divisive [99] direction, which starts with one cluster of all the vertices and continues to separate it to more smaller groups, and agglomerative [25] direction, which oppositely starts from each vertex in a separated cluster and combines smaller clusters in every iteration; there is also spectrum clustering, which utilizes eigenvectors and eigenvalues of adjacency matrices or matrices derived from adjacency matrices [34, 134] to find the membership for each vertex. In contrast, SCAN is motivated by the study of network clustering that may contain special types of vertices: hubs and outliers. Hubs are vertices that connect different clusters and should not be grouped into any of

them. Outliers are isolated vertices with few connections with others. Traditional graph clustering algorithms do not often identify these two types of nodes. Identifying these nodes are essential to many real world applications, as epidemiology, viral marketing, security and privacy, etc. To some degree, SCAN has successfully not only discovered the clusters, but also the hubs and outliers.

SCAN is a similarity based clustering method. Nodes with similarities larger than or equal to the threshold  $\epsilon$  are clustered together. Before introducing the algorithm, we need some preliminary definitions, given a graph  $G = \{V, E\}$ , where  $V$  is the vertex set and  $E$  is the edge set.

**Definition 1 (Vertex Structure).** For a vertex  $v \in V$ , the structure of  $v$  is defined by its neighborhood, denoted as  $\Gamma(v)$ :

$$\Gamma(v) = \{w \in V | (v, w) \in E\} \cup \{v\}$$

**Definition 2 (Structure Similarity).** For vertex  $v, w \in V$ , the structure similarity between  $v, w$ , denoted by  $\sigma(v, w)$ , is defined as:

$$\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

Not stated in the paper, SCAN for weighted graph is also proposed. For weighted graph, the definition for structure similarity has been represented in a different way.

**Definition 3 (Weighted Structure Similarity).** We denote the weight between

vertex  $i, j$  as  $\omega_{ij}$ . For vertex  $v, w \in V$ , the weighted structure similarity between  $v, w$ , denoted by  $\sigma_\omega(v, w)$ , is defined as:

$$\sigma_\omega(v, w) = \frac{\sum_{i \in \Gamma(v), i \in \Gamma(w)} \omega_{vi} \cdot \omega_{wi}}{\sqrt{\sum_{j \in \Gamma(v)} \omega_{vj}^2} \cdot \sqrt{\sum_{k \in \Gamma(w)} \omega_{wk}^2}}$$

**Definition 4 ( $\epsilon$ -Neighborhood).** For a vertex  $v \in V$  and a given similarity threshold  $\epsilon \in \mathbf{R}$ , the  $\epsilon$ -Neighborhood of  $v$ , denoted by  $N_\epsilon(v)$ , is defined as:

$$N_\epsilon(v) = \{w \in \Gamma(v) | \sigma(v, w) \geq \epsilon\}$$

By introducing another threshold  $\mu$ , we can define a special type of vertices, core vertices, which are used as seeds for clusters.

**Definition 5 (Core).** For thresholds  $\epsilon \in \mathbf{R}$  and  $\mu \in \mathbf{N}$ , the core vertex  $v$ , denoted by  $CORE_{\epsilon, \mu}(v)$  is defined as:

$$CORE_{\epsilon, \mu}(v) \Leftrightarrow |N_\epsilon(v)| \geq \mu$$

With these definitions, we can continue to describe of the procedure of SCAN.

- **STEP 1.** Start from an arbitrary core vertex  $CORE_{\epsilon, \mu}(v)$  as the seed.
- **STEP 2.** Group all the vertices in  $N_\epsilon(v)$  with the seed in the same cluster.
- **STEP 3.** For vertices in  $N_\epsilon(v)$  that are core, treat them as seeds.

- **STEP 4.** For each of the seed from *STEP 3*, repeat *STEP 1 - 3*.
- **STEP 5.** Continue until no more vertices can be added.
- **STEP 6.** Start from another arbitrary un-clustered core vertex, repeat *STEP 1 - 5*.
- **STEP 7.** Repeat until no more core vertices are un-clustered.
- **STEP 8.** For each un-clustered vertices, if it connects more than 1 clusters, it is classified as a hub, else it is classified as an outlier.

Using the SCAN algorithm, we can cluster personal networks into circles based on neighborhood similarities and also identify hubs and outliers.

### 4.3.2 Evaluation & Observations

In this section, we will present the clustering result of SCAN on our data sets and make some observations from it. The source code of SCAN is provided by the author of this algorithm [144] Xiaowei Xu.

To use SCAN, we need to generate pair files indicating edges of the network. Currently, we used the unweighted SCAN for only edges within the personal network of the seed in our data set. In the future, we propose to use SCAN for weighted graph, in which the weight between friends is:

$$\omega(i, j) = 1.0 + \lambda \times |N(i) \cap N(j)|$$

$N$  stands for the neighborhood set of a vertex.  $\lambda$  is a parameter which can be tuned to different ratios for the neighborhood similarity. The weighted version integrates more neighbor information than the unweighted one.

### 4.3.2.1 Structural-based Circle Quality Evaluation

To apply SCAN in our data set, we use the default  $\mu = 2$ , and every  $\epsilon$  from 0.1 to 1.0 in step of 0.1 is tested for evaluation. From the 10 clustering results of each seed, we selected the one with largest modularity as the representative SCAN clustering result for the seed. Modularity is a widely utilized criteria for evaluating the structural quality of graph clustering [25, 99, 133]. The basic idea behind it is that good clustering tends to have larger distance from the randomly generated graph with the same degree for each vertex. Formally, it can be calculated as:

**Definition (Modularity).** For a given clustering  $\mathbb{C}$  of a graph  $G = (V, E)$ , modularity of  $\mathbb{C}$ , denoted by  $Modularity(\mathbb{C})$ , is:

$$Modularity(\mathbb{C}) = \frac{1}{2m} \sum_{\mathbb{C}} \sum_{i,j \in \mathbb{C}, \mathbb{C} \in \mathbb{C}} (A_{ij} - d_i d_j / 2m)$$

In the definition,  $\mathbb{C}$  is a set of subsets of  $V$ ,  $m = |E|$ , i.e. the number of edges,  $i, j$  denote vertices,  $A_{ij}$  represents the adjacency between  $i, j$  (check the following definition), and  $d_i, d_j$  stand for the degree of  $i, j$ .

$$A_{ij} = \begin{cases} 1, & \text{if } \{i, j\} \in E. \\ 0, & \text{if } \{i, j\} \notin E. \end{cases}$$

It also needs to be mentioned that we only consider clusters obtained from SCAN with at least 4 users, since we believe that clusters with too few members are not reasonable for real life situations, and they are not suitable for our statistical evaluations.

For a random seed, whose user id is: 15373743, the selected clustering has

$\epsilon = 0.4$  and *modularity* = 0.3657. The result is shown in Table 4.2

**Table 4.2.** Clustering result for seed 15373743

Set	Size
HUBS	5
OUTLIERS	46
$C_1$	65
$C_2$	10
$C_3$	9
$C_4$	24
$C_5$	24
$C_6$	7
$C_7$	5
$C_8$	6
$C_9$	4
<i>SUM</i>	205
<i>NumOfFriends</i>	329

We can see the total number of users classified either in some cluster or into hubs and outliers is less than the seed’s total number of friends. This is due to we only input the friend pairs within the seed’s personal network. In the future, we may input more pair information to SCAN for comparing the results.

We also randomly selected 6 other seeds to get some statistical summary. The result is shown in Table 4.3.

**Table 4.3.** Clustering result for 7 seeds.

Seed	Hub	Outlier	# of Clusters	Largest	Sum Classified	Friends	Mod	$\epsilon$
11069482	4	27	5	78	128	179	0.118	0.4
12792062	1	47	5	8	79	447	0.297	0.4
14451009	1	9	3	17	56	165	0.487	0.4
14665656	4	62	8	100	224	359	0.193	0.4
15373743	5	46	9	65	205	329	0.366	0.4
16358915	74	96	4	33	216	299	0.039	0.5
18193317	3	15	5	104	180	248	0.148	0.4
Average	13	43	6	58	155	289	0.235	0.41

Each column of Table 4.3 corresponds to: the seed user id, number of hubs identified, number of outliers found, number of clusters, largest cluster size,



total number of friends classified into hubs, outliers or any cluster, total number of friends of the seed, and the best corresponded modularity, and  $\epsilon$  obtained from SCAN.

From the result, we can see that there is often a dominant cluster for each seed, which may be caused by the cluster growing mechanism that only considers neighborhood similarities. Modularities are often below 0.3. There is one seed who has relatively large value. According to [25], for modularity, greater than about 0.3 can indicate remarkable community structure in networks. We can also observe that seeds often have large outlier groups, which is possibly the result of that many users in online social networks are not active and tend to have few friends. In addition, we can see that  $\epsilon$  tends to be constant, mostly 0.4, which justifies the statement from [144] that the recommended  $\epsilon$  is from 0.5 to 0.8 and  $\mu = 2$ .

Even though, the modularity does not exceed 0.3, we believe that the results of SCAN are relatively good enough. Based on this, we propose our first observation.

**Observation 1.** *Users in the same circle are connected and share many friends in common.*

In the future, we will analyze all the seeds in our data set and have a more complete statistical observation.

#### 4.3.2.2 Content-based Circle Quality Evaluation

To further study the result of SCAN, we propose to analyze the content similarities between users within clusters and users in different clusters.

**Circle Quality Evaluation Using TF-IDF** Firstly, we study the raw tweets content for evaluation. We consider tweets created by each user. We input all the tweets content from a user into a new file, and construct the bag-of-words model from all the tweets files from a personal network. In *bag-of-words* model, a *wordlist* is first constructed by extracting all the different words from the group of files after removing stop words and stemming (i.e. mapping the words to their roots). Then, each file is represented with a *vector*. Each *feature* of the vector corresponds to each word from the word list. One entry of a vector corresponding to term  $t$ , document  $\mathbf{d}$ , denoted as  $\mathbf{d}(t)$ , given the whole corpus as  $\mathbb{D}$ , is calculated as:

$$\begin{aligned}\mathbf{d}(t) &= tf(t, \mathbf{d}) \times idf(t, \mathbb{D}) \\ tf(t, \mathbf{d}) &= f(t, \mathbf{d}) \\ idf(t, \mathbb{D}) &= \log \frac{|\mathbb{D}|}{|\{\mathbf{d} \in \mathbb{D} | t \in \mathbf{d}\}|}\end{aligned}$$

$f(t, \mathbf{d})$  is number of appearances of  $t$  in  $\mathbf{d}$ . We use a tool called **wvtool**<sup>4</sup> for constructing the *tf-idf* matrix from the raw tweets data.

Then we can use cosine similarities for calculating the angle between 2 document vectors to indicate the content similarity of the corresponded 2 users. The cosine similarity for 2 vectors  $\mathbf{a}$ ,  $\mathbf{b}$ , denoted as  $CosSim(\mathbf{a}, \mathbf{b})$ , is:

$$CosSim(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^n (\mathbf{a}(i) \times \mathbf{b}(i))}{\sqrt{\sum_{i=1}^n \mathbf{a}(i)^2} \times \sqrt{\sum_{i=1}^n \mathbf{b}(i)^2}}$$

Then for each cluster  $C$ , we calculate average cosine similarities among all the pairs of users within  $C$  as intra similarity of  $C$ , denoted by  $IntraSim(C)$ .

<sup>4</sup><http://sourceforge.net/projects/wvtool/>

We also calculate inter similarity of  $C$ , denoted by  $InterSim(C)$  by averaging cosine similarities between users in  $C$  and users not in  $C$ .

$$IntraSim(C) = \frac{\sum_{i \in C, j \in C} CosSim(\mathbf{d}_i, \mathbf{d}_j)}{|C|(|C| - 1)/2}$$

$$InterSim(C) = \frac{\sum_{i \in C, j \notin C} CosSim(\mathbf{d}_i, \mathbf{d}_j)}{|C|(|\mathbb{U}| - |C|)}$$

$\mathbb{U}$  is the set containing all the users of a personal network. The intra and inter similarity of each cluster for seed 15373743 is listed in Table 4.4.

**Table 4.4.** Circle Quality Evaluation Using TF-IDF for seed 15373743

Cluster	$IntraSim$	$InterSim$
$C_1$	0.151	0.047
$C_2$	0.031	0.022
$C_3$	0.101	0.040
$C_4$	0.040	0.023
$C_5$	0.048	0.029
$C_6$	0.023	0.027
$C_7$	0.021	0.022
$C_8$	0.020	0.028
$C_9$	0.033	0.023

To compare the clustering quality according to intra and inter similarities for different seeds, we propose to calculate a ratio, called intra similarity proportion, denoted by  $IntraSimPro$ .

$$IntraSimPro(seed) = \frac{IntraSimSum(seed)}{IntraSimSum(seed) + InterSimSum(seed)}$$

$$IntraSimSum(seed) = \sum_{C \in \mathbb{C}_{seed}} IntraSim(C)$$

$$InterSimSum(seed) = \sum_{C \in \mathbb{C}_{seed}} InterSim(C)$$

We calculate the intra similarity proportion for each of the randomly se-

lected seeds, and the result is shown in Table 4.5.

**Table 4.5.** Circle Quality Evaluation Using TF-IDF for 7 randomly selected seeds

<b>Seed</b>	<i>IntraSimPro</i>
11069482	0.648
12792062	0.641
14451009	0.745
14665656	0.712
15373743	0.643
16358915	0.671
18193317	0.717
Average	0.682

From the result, we can see that intra similarities tend to be larger than inter similarities for all the seeds, which fits the requirement for social network clustering that users within circles are more similar. Based on this, we propose our second observation:

**Observation 2.** *Users from the same social circle tend to have similar content.*

**Circle Quality Evaluation Using Tags** In this section we will continue to talk about content evaluation of clustering result based on tags.

Tags are often categorized short texts with clear semantic information. It is nowadays ubiquitous online. You can see tags on wikipedia<sup>5</sup> pages, which link to their corresponding web source. Users of online social networks can tag web articles, photos, videos with some short phrases when sharing. They can also tag their friends when they upload photos containing them. Tags are usually a convenient way for others to understand web content.

How to extract tags from web content automatically becomes an important question. It can largely facilitate researchers in different fields, and im-

---

<sup>5</sup><http://www.wikipedia.org>

prove ordinary users' web experience. Compared to the context where content has more information, short text topic discovery is much harder, and becomes more and more important recently. For our research, short text tagging is more related and suitable. There have already been some works for this problem [37, 80, 102, 106, 107, 128, 129]. Some of them focus on creating a large knowledge base, and try to eliminate the negative impact of short texts' little semantics. Others try to solve the problem independent of other resources. Their methods are often based on some observations about the semantic implications of non-semantic information of the short text, such as length, capitalization of short texts. In this proposal, we propose to use a web service, called **TAGME**<sup>6</sup> for extracting topics of short texts of tweets. It is based on wikipedia knowledge base and trying to link text in the queries with wikipedia pages. The topics of the linked pages can be used as the tags of the tweets.

For time issues, we have just finished topic detection for one seed of our data set: 15373743. So in this section, we will talk about our tag evaluation mechanism based on this user.

For each tweet, we select the top 3 extracted tags by TagMe (select all if the TagMe result contains less than 3 tags) with largest  $\rho$ , which is an indicator of the "goodness" of TagMe annotation about the topics of the query. For seed 15373743, we collect 117201 distinct tags. Figure 4.3 shows the number of appearance for each tag among its friends.

We can see that the distribution complies power-law, where most of the tags appear infrequently, and only a few of them appear many times.

Based on tags, each user can be represented by a tag file containing tags

---

<sup>6</sup><http://tagme.di.unipi.it>

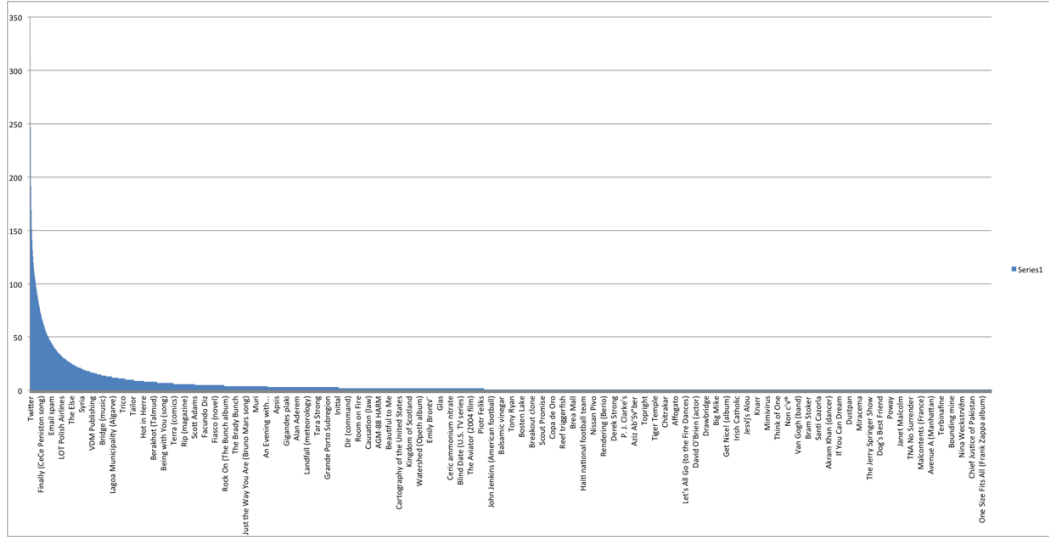


Figure 4.3. Tag distribution of seed 15373743

of its tweets. We can apply the *tf-idf* model on the tag files to calculate the similarities. We use similar calculation as for raw tweet content evaluation, where *t* stands for tags instead of terms. However, to normalize the *tf*, so that there is no bias toward longer files, we propose to use  $tf_{norm}$ :

$$tf_{norm}(t, \mathbf{d}) = \frac{f(t, \mathbf{d})}{\max\{f(w, \mathbf{d}) | w \in \mathbf{d}\}}$$

$f(t, \mathbf{d})$  is the number of appearance of tag *t* in file  $\mathbf{d}$ , and  $\max\{f(w, \mathbf{d}) | w \in \mathbf{d}\}$  returns the max frequency of all the tags in  $\mathbf{d}$ .

Then we can construct vectors for each user, and calculate *CosSim*, *IntraSim*, *InterSim* as presented in section 4.3.2.2. The result is show in Table 4.6.

Although we can see that the tag-based similarity measure has similar *IntraSimPro* as the raw-tweet similarity, more clusters have larger *IntraSim* than *InterSim* in tag-based model. Generally, the tag-based similarity further justifies our **Observation 2**.

**Table 4.6.** Circle Quality Evaluation Using Tags for seed 15373743

<b>Cluster</b>	<i>IntraSim</i>	<i>InterSim</i>
$C_1$	0.145	0.046
$C_2$	0.039	0.028
$C_3$	0.069	0.041
$C_4$	0.039	0.025
$C_5$	0.079	0.036
$C_6$	0.029	0.028
$C_7$	0.028	0.027
$C_8$	0.035	0.036
$C_9$	0.061	0.031
<i>IntraSimPro</i>	0.638	

To continue the content evaluation into a deeper level, we want to find the most unique tags for each SCAN clustered group. To do so, we want to calculate the probability of appearance for each tag in each cluster. The tag with larger bias towards a cluster can better represent the content specificity of the cluster. Instead of calculating the number of users in a group with a tag divided by the group size as the tag group probability, we propose to use  $tf_{norm}$  to normalize the content quantity of each user. Formally, the probability of tag  $t$  in cluster  $C$ , denoted by  $P(t|C)$  can be defined as:

$$P(t|C) = \frac{\sum_{i \in C} tf_{norm}(t, i)}{|C|}$$

As a result, for one group, the more users having the tag  $t$  appearing more times in their tweets, the larger probability the tag has in that group.

For finding the biasness of tags toward certain cluster, we propose to utilize *Kullback-Leibler Divergence* (denoted as KL-divergence). KL-divergence is commonly used to calculate the difference between 2 probability distributions. To apply it for our requirement, we firstly construct 2 discrete probability distri-

butions  $P_t(i), Q_t(i)$ .

$$P_t(i) = \frac{P(t|C_i)}{\sum_{C_i \in \mathbb{C}} P(t|C_i)}$$

$$Q_t(i) = \frac{1}{|\mathbb{C}|}$$

So now, we can use the formula for KL-divergence to calculate the bias of tags:

$$Bias_{KL}(t) = \sum_i (P_t(i) \ln \frac{P_t(i)}{Q_t(i)})$$

The largest  $Bias_{KL}$  we get is for tag: “Global Positioning System”, with  $Bias_{KL} = 1.273$ .

Then for each cluster  $C_i$ , we can find the tags with largest  $Bias_{KL}$ , and having max  $P(t|C)$  in  $C_i$  as the representative tags for  $C_i$ . The top 5 tags for each cluster of seed 15373743 are shown in Table 6.4.

**Table 4.7.** Representative tags for clusters of seed 15373743

Cluster	Representative
$C_1$	Alpha Comae Berenices,Europe,Ampere,Wave,Brazil
$C_2$	Global Positioning System,System,The People,Wine,Running
$C_3$	NPR,Travel,Documentary film,City,Tourism
$C_4$	Essay,E-book,Health care,On Your Own (Blur song),Australia
$C_5$	Packaging and labeling,Tree,Web design,Nike,Web application
$C_6$	Wedding,Filmmaking,Mexico,Stock photography,Workshop
$C_7$	Geek,News,Doepfer A-100,Graphics Interchange Format,Happening
$C_8$	Music download,T-shirt,Propaganda,Graphic designer,Minute
$C_9$	Search engine optimization,Sugar,Web conferencing,United States Treasury security,Google+

From these tags, we can clearly see the content difference between each cluster.  $C_1$  is likely to be formed by people from other countries.  $C_2$  is more about outdoor activities.  $C_3$  focus on traveling, while  $C_4$  seems to concern health and study.  $C_5$  is related to programmer.  $C_6$  is more about art.  $C_7$  seems rather technical about graphics, and  $C_8$  is also about art but a little different from  $C_6$ .  $C_9$  seems to be for very high level topics. Based on these analysis, we can see that



tag-based evaluation can help us find the most specific topics for each social circle.

### 4.3.2.3 Interaction-based Circle Quality Evaluation

In this section, we will propose the methods used for interaction evaluation, show some results and make some observation.

In our scenario, interactions are tweets reply between users. Refer to section 4.2, in our data set each tweet contains *ID*, *Create Time*, *Location*, *In reply to*, and *Content*. Based on this, for each friend of the seed, we can extract every tweet that replied to him/her from the same personal network. Each of these tweets is considered as one interaction. So, given a clustering of a seed's personal network  $\mathbb{C}$ , for cluster  $C_i$ , we can count its number of intra interactions, and inter interactions. We use  $\vec{I}_{i,j}$  to denote an interaction that user  $i$  replies user  $j$ . Formally, intra interaction *IntraIntr* and inter interaction *InterIntr* can be defined as:

$$IntraIntr(C) = |\{\vec{I}_{i,j} | i \in C, j \in C\}|$$

$$InterIntr(C) = |\{\vec{I}_{i,j} | i \in C, j \notin C, j \in \mathbb{U} \vee j \in C, i \notin C, i \in \mathbb{U}\}|$$

As mentioned before,  $\mathbb{U}$  is for the set of all the friends of a seed. To evaluate the intra and inter interaction characteristic of a clustering, similar as content evaluation, we propose the *IntraIntrPro* which stands for intra interaction proportion. To eliminate the size difference between intra cluster and inter cluster, we calculate the average intra and inter interactions. Formally,

*IntraIntrPro* can be calculated as:

$$\begin{aligned}
 IntraIntrPro(seed) &= \frac{AvgInIntrSum(seed)}{AvgInIntrSum(seed) + AvgOuIntrSum(seed)} \\
 AvgInIntrSum(seed) &= \sum_{C_i} \frac{IntraIntr(C_i)}{|C_i|} \\
 AvgOuIntrSum(seed) &= \sum_{C_i} \frac{InterIntr(C_i)}{|\mathbb{U}| - |C_i|}
 \end{aligned}$$

Based on these calculations, we can get the *IntraIntrPro* for each seed. The result for the 7 randomly selected seeds is shown in Table 4.8.

**Table 4.8.** Interaction-based Circle Quality Evaluation for 7 randomly selected seeds

Seed	<i>IntraIntrPro</i>
11069482	0.610
12792062	0.861
14451009	0.702
14665656	0.883
15373743	0.904
16358915	0.752
18193317	0.954
Average	0.809

From the result, we can see that the intra interaction proportion is high for every seed. Especially for seeds 15373743 and 18193317. Recall Table 4.3, we can see that both of them have a large number of friends clustered into one group and the rest groups much smaller. This will certainly increase the intra interaction probability. This type of clustering may be caused by SCAN (its cluster expansion based on neighborhood similarity instead of complete similarity) or the seed just happens to have this structure. We need further analysis about this point. Although, the result may not be accurate enough, we can still believe that intra interaction probability should be higher than inter interaction

probability. Here we propose our third observation:

**Observation 3.** *Users in the same circle are more likely to interact.*

Based on these three observations obtained from evaluation, we will propose a new technique for automatic social circle detection in personal networks next chapter.

# Chapter 5

## Multi-View Clustering for Social Circle Detection

### 5.1 Multi-View Clustering for Social Circle Detection:

#### Motivation

The problem of social community discovery has been studied in the context of social network evolution. Closely-related social groups are examined to analyze the temporal and spatial dynamics of social networks. However, such approaches heavily rely on structural features (i.e., topology of the friendship graph), and may have difficulties on users with too many or too few links (sometimes referred-to as “hubs” and “outliers”). Meanwhile, social circle identification approaches from the user-centered research community often require explicit attributes, e.g. education=“stanford”, age=21, hobbies=“hiking” [125]. Unfortunately, such attributes are not always available in online social networking sites.

In this chapter, we present a multi-view clustering approach to automati-

cally discover social circles in users' ego networks. Besides the topology-based clusters adopted in the literature, we also observe that: (1) friends who are interested in similar topics (contents) and share similar (or sometimes opposite) opinions are more likely to be placed in the same circle (by the user); (2) friends are more likely to interact within circles, than cross circles. Based on the observations, we build computational models to extract multiple quantitative features from users' ego networks. We argue that integrating all structural, content and interaction features will improve clustering performance, and eventually generate more meaningful social circles. We notice that some views are very sparse (e.g. the views for user interactions), but they provide stronger indications, when two friends are associated in such sparse views. To better utilize such properties, we present a *Selective Co-Trained Spectral Clustering* (SCSC) algorithm for multi-view clustering. Last, to measure the performance of the proposed modeling and clustering approaches, we introduce a set of quantitative and user-based evaluation methods. We test our approaches with real-world social networking data collected from Twitter, and show that SCSC outperforms existing solutions.

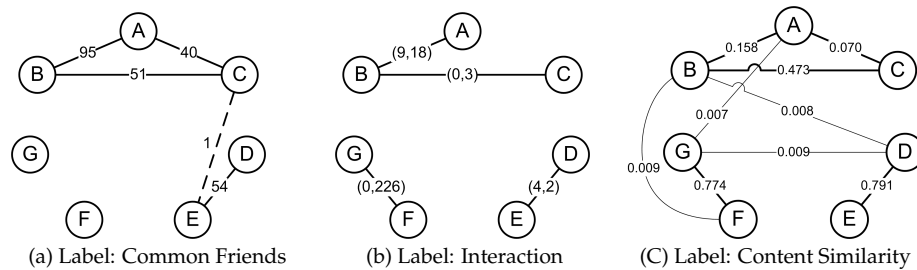
The rest of this chapter is organized as follows: we first present our models of three categories of features in Section 5.2. We then describe the multi-view clustering algorithms in Section 5.3.

## 5.2 Ego Network Modeling

By definition, a user's ego network or personal network includes all the nodes that connect to the user, i.e., all his/her friends. Social circles of a user's ego network are hidden structures of closely connected clusters. For instance,

a user’s high-school friends may constitute a circle, while his/her colleagues belong to a different circle, and his/her family members constitute yet another circle.

Existing research on social community discovery mostly rely on graph topology, i.e., structural features. However, social circles may not be revealed by a single aspect of the ego networks. Instead, they need to be inferred from multiple features. For example, colleagues may interact frequently offline so that they have few online interactions, however, they are highly *connected* to each other in the friendship graph. Meanwhile, a family member may be connected to some close colleagues on the friendship graph, however, he/she will mostly *interact* with other family members, which is a definitive indicator that he/she belongs to the family circle.



**Figure 5.1.** Labeled Real World Online Social Network Subnet

**Example 1:** Figure 5.1 demonstrates a small subgraph from real-world twitter data set. Two users are regarded as friends if they mutually follow each other. The subgraph is extracted from friends of one seed user. For simplicity, the seed user is not displayed. First, Figure 5.1 (a) demonstrates the friendship graph – solid lines indicate direct friendship relations, while dashed lines indicate users without direct connections but sharing neighbors. All the lines are labeled with the number of common friends (excluding the seed user). Next, Figure 5.1

(b) summarizes the interactions among the users. Each edge is labeled with  $(N_{rp}, N_{rt})$ , which indicates the number of replies and re-tweets between the two users, respectively. Last, Figure 5.1 (c) demonstrates the content similarities between each pair of users (only labels  $\geq 0.0065$  are shown). We show edges with labels  $\geq 0.01$  in thick lines.

As shown in the graphs, three views confirm each other in some regions, while they also complement each other. For instance, the strong connection between nodes A and B in Figure 5.1 (a) is confirmed by the frequent interactions in Figure 5.1 (b). The weak connection between C and E in Figure 5.1 (a) could be eliminated given the facts in Figure 5.1 (b) and (c). Nodes G and F are disconnected in Figure 5.1 (a), however, they have a large amount of interactions and very high similarities in their tweet contents, which also indicates a close relationship. In summary, we can identify three social circles from this example:  $\{(A, B, C); (D, E); (G, F)\}$ . As we can see, different perspectives can supplement and confirm each other, which may produce better clustering results.  $\square$

Formally, an *ego network*  $E_S$  is defined as the subgraph of the social network that includes all the friends of a *seed user*  $S$ . Note that the seed user himself is not included in the ego network. In the Twitter data set that we use, two users are defined as *friends* if and only if they follow each other. In the ego network, each vertex  $(N_i)$  represents a friend of the seed user, while the edges are defined differently for different views. In general, we have observed three phenomena about users' grouping behavior:

**Observation 1.** *Users in the same circle are more likely to be connected and share many friends in common.*

**Observation 2.** *Users from the same social circle tend to share interests on similar contents and opinions.*

**Observation 3.** *Users in the same circle are more likely to interact with each other.*

From these observations, we propose to integrate three aspects of information from users’ ego networks to automatically identify non-overlapping social circles. We define six views that belong to three categories to model the ego networks. From the *structural* perspective, we capture: (1) the friendship links, and (2) friends-in-common between pairs of users. From the *content* perspective, we model: (3) similarities between two users’ posted/shared messages. Finally, from the *interaction* perspective, we construct: (4) direct replies between pairs of users, (5) re-tweet (similar to “forward”) of posts between pairs of users, and (6) co-replies of the same message (posted by a third user).

**The Structural Model.** In social networking research, it is widely accepted that a group of intensively connected nodes could be considered as a social community. For each pair of users, they are more structurally connected if they (1) are friends and/or (2) share more friends. We quantitatively capture the structural features in these two layers and create two views correspondingly.

We use an adjacency matrix  $F$  to capture the first layer.

$$F(i, j) = \begin{cases} 1 & \text{if } N_i \text{ and } N_j \text{ are friends.} \\ 0 & \text{if } N_i \text{ and } N_j \text{ are not friends} \end{cases}$$

Meanwhile, the matrix of shared friends ( $H'$ ) for an ego network  $E_S$  is defined as:

$$H'(i, j) = |E_{N_i} \cap E_{N_j}| - 1$$



where  $E_{N_i}$  and  $E_{N_j}$  denote the ego networks of users  $N_i$  and  $N_j$ . Note that, we consider shared friends within and outside of the ego network. We do not count  $S$  as a shared friend, since  $S$  contributes equally to all  $(N_i, N_j)$  pairs. Furthermore, the matrix is normalized by dividing each element by the largest element in the matrix:

$$H(i, j) = \frac{H'(i, j)}{\max_{i, j} H'(i, j)}$$

Eventually, we have generated two views  $F$  and  $H$  to capture the structural relationships between pairs of users in the ego network.

**The Content Model.** From the content perspective, we examine the semantic similarities of contents between pairs of users in  $E_S$ . We collect all the tweets, replies and re-tweet messages posted by a user. We exploit the traditional bag-of-words model, where all the messages posted by the user are represented as a vector ( $D_i$ ) in the vector space. While the conventional TF-IDF model is the most popular method in information retrieval applications, it suffers some drawbacks, especially the ambiguity issue – synonyms are considered orthogonal axes in the term space. Hence, documents about similar content but from different vocabulary will be assessed as highly irrelevant. To tackle this problem, annotation-based approaches have been proposed to label documents with pre-selected unambiguous terms (topics) so that documents are represented in the new unambiguous “topic space”. In this dissertation, we employ TagMe [42], which annotates text corpus with topics in Wikipedia. Each tag is associated with a “goodness” score,  $\rho$ , which denotes the annotating confidence. By setting a threshold for  $\rho$ , we can eliminate all the low-confident tags to reduce noise and ambiguity, and improve the calculation efficiency. In prac-

tice, we construct a document vector  $\mathbf{T}_i$  for user  $N_i$ , where each component represents the corresponding TF-IDF weight in the tag space. The content-based similarity matrix  $C'$ , with cosine similarity, is further defined as:

$$C'(i, j) = \text{sim}(\mathbf{T}_i, \mathbf{T}_j) = \frac{\mathbf{T}_i \cdot \mathbf{T}_j}{|\mathbf{T}_i||\mathbf{T}_j|}$$

We normalize  $C'$  in the same way as we normalize the shared-friend matrix ( $H'$ ). Finally, we have constructed the content view for ego network  $E_S$ , to capture the content-based similarities among the users.

**The Interaction Model.** Interactions of online social network users have different forms: reply on each other’s status or posted messages, “like” or “dislike” on the messages, retweet. etc. For each pair of nodes within an ego network, we consider three types of interactions: reply, retweet, and co-reply. For reply, we count both directions – the total number of replies from  $N_i$  to  $N_j$  and replies from  $N_j$  to  $N_i$ . Therefore, the reply matrix could be denoted as:

$$P(i, j) = |\{\vec{r}_{i,j}\}| + |\{\vec{r}_{j,i}\}|$$

We do the same for retweet, while co-reply is undirected. In this way, we generate three views, and normalize them as we do with  $H'$  and  $C'$ . As a result, we have constructed the reply view  $P$ , the re-tweet view  $T$ , and Co-reply view  $O$ .

Overall, we construct six views from personal networks: two from the structural perspective, one for content, and the other three from user interactions. Each view is represented as a matrix demonstrating similarities between each pair of users within an ego network. The next step is to integrate these views

---

**Operator 1:**  $LapEig(X, k) = Y$ 

---

**Input:**  $X \in \mathcal{R}^{n \times n}$  and  $k \in \mathcal{N}$ **Output:**  $Y \in \mathcal{R}^{n \times k}$ **Operation:**1: Compute diagonal matrix  $D$  with  $D_{(ii)} = \sum_{j=1}^n X_{(ij)}$ 2: Compute Laplacian  $L = D^{-1/2} X D^{-1/2}$ 3: Compute the top  $k$  eigenvectors of  $L$ , and store them in  $Y$  with each column as one eigenvector

---

to identify social circles.

## 5.3 Multi-View Clustering

### 5.3.1 Notations and Operators

We use capital letter to represent matrix, boldface to represent vector and lower-case to represent scalar. Subscript without parenthesis is used to indicate views, subscript with parenthesis is used to indicate elements in matrices or vectors, and superscript is used to indicate iteration number in an iterative algorithm. For example,  $X_{(m,n)}$  represents the element of matrix  $X$  on row  $m$  and column  $n$ , and  $X_j^i$  represents matrix  $X$  of view  $j$  in the  $i_{th}$  iteration.  $tr(S)$  is used to denote the trace of  $S$  matrix,  $A \circ B$  to denote the Hadamard product (element-wise product) between matrix  $A$  and matrix  $B$ , and  $1_E$  to denote the element-wise indicator function on  $E$ .

For convenience we define two operators in Operator 1 and Operator 2.

### 5.3.2 Co-trained Spectral Clustering: A Revisit

In this section we briefly review co-trained spectral clustering (CSC) [68], which is a clustering algorithm for multi-view data. In spectral clustering, it

---

**Operator 2:**  $Cls(X, k) = Y$ 

---

**Input:**  $X \in \mathcal{R}^{n \times k}$  and  $k \in \mathcal{N}$ **Output:**  $Y \in \mathcal{R}^{n \times n}$ **Operation:**1: Normalize each row of  $X$ 2: Run  $k$ -means on rows of  $X$  to obtain an  $n$  by  $n$  matrix  $Y$  such that  $Y_{(i,j)} = 1$  if user  $i$  and user  $j$  are in the same cluster, and  $Y_{(i,j)} = -1$  if the two users are not in the same cluster

---

has been shown that the eigenvectors of the graph Laplacian contains robust discriminative information about the cluster, and hence by applying standard clustering techniques on the eigenvectors may lead to a better clustering result. When multiple views of data are available, CSC alternately refines the graph Laplacian of one view based on the clustering result suggested by other views. The refinement is realized by projecting and reconstructing the Laplacian of one view onto the eigenvectors of the graph Laplacians of other views. This process iterates and glues the graph edges within a cluster and differs edges between clusters. The final clustering result is obtained by performing single-view spectral clustering on the refined Laplacians of dominant views.

CSC assumes the graph of each view is completely observed, and transfers the complete graph information across views. However, in our application, graphs of many views are partially observed (hence extremely sparse). For example, intimate users may communicate frequently by replying to each other, while ignoring retweeting messages. In this scenario, we hypothesize that enforcing a completely agreement between the retweet view and other views will mis-refine Laplacians and degenerate clustering performance. As we justified in experimental study, this is indeed a problem.

---

**Algorithm 1** Selective Co-trained Spectral

---

**Input:** Similarity matrix of two views:  $K_1, K_2$

**Output:** Cluster matrix  $C$

**Initialize:**  $U_j^0 = LapEig(S_j^i, k), C_j^0 = Cls(U_j^0, k), j \in I_v$

$$K_{all} = \{K_j\}_{j \in I_v}, C_{all}^0 = \{C_j^0\}_{j \in I_v}$$

**for**  $i = 1$  **to**  $iter$  **do**

**for**  $j = 1$  **to**  $views$  **do**

    1:  $R_j^i = SO(C_{all}^{i-1}, K_{all}, j)$

    2:  $S_j^i = R_j^i \circ K_j$

    3:  $U_j^i = LapEig(S_j^i, k)$

    4:  $C_j^i = Cls(U_j^i, k)$

**end for**

    5:  $C_{all}^i = \{C_j^i\}_{j \in I_v}$

**end for**

6: Choose the dominant view  $j$  and run  $Cls(U_j^i, k)$  to get the cluster matrix.

---

### 5.3.3 Selective Co-trained Spectral Clustering

In this section we propose the new multi-view clustering algorithm. We identify a graph as partial if the number of non-zero entries (edges) in the graph Laplacian is below a pre-specified threshold, and safely assume that only non-zero edges in partial graphs are observed. Our intuition is that, only clustering result on observed edges should be transferred from views with partial graphs.

The proposed Selective Co-trained Spectral Clustering (SCSC) multi-view clustering approach is presented in Algorithm 1. The major difference between SCSC and CSC are twofold: 1) CSC uses eigenvectors of the Laplacian of one view to refine Laplacians of other views, while SCSC uses the clustering result of one Laplacian to refine other Laplacians, which tends to be more precise; 2) CSC transfers the complete graph information across views, while SCSC selectively transfers graph information.

Operation  $SO(\cdot)$  realizes the selective process. Let  $\rho_j$  be defined as

$$\rho_j = \frac{\# \text{ zeros in } K_j}{\# \text{ all elements in } K_j}, \quad (5.1)$$

where  $K_j$  represents the similarity matrix of view  $j$ , and abbreviate  $SO(C_{all}^{i-1}, K_{all}, j)$  as  $SO(j)$ , we design

$$SO(j) = \exp \left( C_{j'} \circ \left( 1_{\{K_{j'} \neq 0\}} \right)^{1_{\{\rho_{j'} > \rho_{thre}\}}} \right), \quad (5.2)$$

where  $C_{j'}$  represents the clustering matrix of view  $j'$  such that if user  $p$  and user  $q$  are assigned to the same cluster in this view, then the  $p_{th}$  row and  $q_{th}$  column of  $C_{j'}$  is 1, otherwise it is zero. In addition,  $j' \in I_v, j' \neq j$  and  $\rho_{thre}$  is a pre-specified threshold. The intuitions behind  $SO(j)$  are as follows:

- For  $C_{j'}$ , if two users are assigned to the same cluster in view  $j'$ , then the corresponding element in  $C_{j'}$  is 1 and  $SO > 1$  so their Laplacian in other views will be boosted, and vice versa.
- For  $I_{\{K_{j'} \neq 0\}}$ , if two users have non-zero edges in the partial graph of view  $j'$ , then  $I = 1$  and their clustering result in view  $j'$  will be transferred to other views; otherwise  $SO = 1$  and their Laplacian in other views will not be affected.
- $I_{\{\rho_{j'} > \rho_{thre}\}}$  is used to identify which views have partial graphs based on threshold  $\rho_{thre}$ . Given a view with partial graph, we have  $I = 1$  and hence the selective component  $I_{\{K_{j'} \neq 0\}}$  will take effect; otherwise  $I = 0$  and all information of the graph is transferred.

It is worthy to elaborate more on the convergence property of our algorithm.

Consider an example of clustering three users with two views. From one view we obtain the clustering result

$$C_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}, \quad (5.3)$$

which implies user 1 and user 3 are similar while user 2 and user 3 are not. For view 2 we have the similarity matrix (not the Laplacian)

$$K_2 = \begin{bmatrix} 1 & 0.8 & 0.1 \\ 0.8 & 1 & 0.6 \\ 0.1 & 0.6 & 1 \end{bmatrix}, \quad (5.4)$$

which implies user 2 and user 3 are similar while user 1 and user 3 are not. Apparently there is certain consistency between views. After refining  $K_2$  by  $C_1$  based on Algorithm 1, we have the updated similarity  $K'_2$  such that

$$K'_2 = \begin{bmatrix} 2.7 & 2.2 & 0.3 \\ 2.2 & 2.7 & 0.2 \\ 0.3 & 0.2 & 2.7 \end{bmatrix}. \quad (5.5)$$

It can be seen that after refinement, we manage to adjust user 2 and 3 in view 2, so that their updated (low) similarity becomes consistent with the clustering result of view 1. However, for user 1 and 3, due to their strong evidence of dissimilarity in view two, they remain dissimilar after refinement.

The above example tells that, our algorithm can effectively adjust and converge on users whose similarity are “uncertain” in some views, but does not

enforce agreement and converge on users with strong evidence on two views that are against each other. This is the same issue with existing co-trained spectral clustering, as evidenced by its empirical performance on real-world data set that contain view-inconsistent (noisy) data. Moreover, in a view with partial graph, we may have massive strong evidence of dissimilarity between users, which are largely against their similarity in other views. In this case, our selective algorithm is expected to gain much faster convergence rate than non-selective clustering algorithms, as will be seen in our experimental study. In addition, notice that since update matrix  $C$  is symmetric, the refined similarity matrix  $K$  remain symmetric and corresponding Laplacian remains positive semi-definite, which is guaranteed to have real positive eigenvalue.

To extend  $SCSC$  to multi-view setting, we let  $j = \{1, 2, \dots, \ell\}$  and define  $SO(j)$  as:

$$SO(j) = \exp \left( \sum_{\substack{j' \neq j \\ j'=1}}^{\ell} C_{j'} \circ \left( I_{\{K_{j'} \neq 0\}} \right)^{I_{\{\rho_{j'} > \rho_{thre}\}}} \right). \quad (5.6)$$

The summation in (5.6) follows the majority voting principle: if two users are grouped in more than half of the other views, then their similarity in the current view should be boosted; otherwise their similarity should be decreased. More interestingly, for  $C_{j'}$  if half of the views grouped two users while the other half separated them, then the summation equals zero and  $SO(j) = 1$ , in which case we maintain the similarity of the current view.



## Chapter 6

# Evaluation of Multi-view Clustering for Social Circle Detection

### 6.1 Evaluation Metrics

To evaluate the quality of our clustering result is quite challenging since no ground truth is provided and no feature matrix is available (in fact, not even defined) in most views. This prevents the use of standard external evaluation metrics such as random index or F-measure or internal evaluation metrics such as Davies-Bouldin index and Dunn index. Hence we propose a new internal metric that requires only the similarity matrix. We believe that better clustering should group users that are not only structurally cohesive (more friendship relations among them), but also interact more frequently and post similar content. Based on this, our evaluation is tripartite.

We first propose the normalized similarity ratio to evaluate the performance of clustering result for each view. Our design follows the same idea as Fisher ratio [11], and consists of three parts, i.e., within-cluster similarity, between-

cluster similarity and sparse degree. To prevent the result from being dominated by extra-large clusters, we normalize each similarity by the size of assigned clusters. Consider an arbitrary view, let  $d_i$  denote the size of the cluster (number of users in that cluster) assigned to user  $i$ , and recall that  $K$  is the similarity matrix and  $C$  is the cluster matrix. We define the within-cluster similarity as

$$S_{wc} = \frac{1}{N_w} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} K_{(i,j)} I_{\{C_{(i,j)} > 0\}} \quad (6.1)$$

and the between-cluster similarity as

$$S_{bc} = \frac{1}{N_b} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} K_{(i,j)} I_{\{C_{(i,j)} < 0\}}, \quad (6.2)$$

where  $N_w, N_b$  are the normalizers that respectively count the number of positive and negative elements in  $C$ .

By definition,  $S_{wc}$  denotes the average similarity between users in the same cluster, and  $S_{bc}$  denotes the average similarity between users in different clusters. For high quality clusters, it is quite natural to expect similarity within groups are larger and similarity between groups are smaller. Hence we define the *Normalized Similarity Ratio* as

$$NSR = \frac{S_{wc}}{S_{bc} + \alpha}, \quad (6.3)$$

where  $\alpha$  is a small constant in case  $S_{bc} = 0$ , which happens frequently on sparse views.

It is worthy to note that, we do not directly penalize imbalance cluster results, since in applications some social circles, such as family, are indeed smaller

than other circles, such as friends. However, our metric will lower the score when a super-large cluster appears.

To evaluate the performance over all views, we define the total similarity ratio. Let  $S_{wc[j]}$  and  $S_{bc[j]}$  respectively denote the within-cluster and between-cluster similarity ratio of view  $j$ , where  $j \in 1, 2, \dots, \ell$ , we define the total similarity ratio on one data set as

$$NSR_T = \frac{\sum_{j=1}^{\ell} S_{wc[j]}}{\sum_{j=1}^{\ell} S_{bc[j]}} \quad (6.4)$$

## 6.2 Data Collection and View Construction

At present, Facebook and Twitter are two most popular social networking sites, judging by number of active users and daily traffic. Since Facebook users mostly use real identities, it enforces constraints that prevent us from collecting large amount of data. In this research, we collected our data sets from Twitter, which is most recognizable for the “tweet” function – the microblog service.

We have implemented a crawler to collect Twitter data using its API, which allows us to get a Twitter user’s profile, follower/following lists, and tweet messages. We start with a random user as the seed, and crawl all his/her information (profile, follower/following lists and most recent 2,000 tweets). The intersection of the *follower list* and the *following list* are regarded as *friends*. We crawl the same set of information from the seed user’s friends. All the collected data about a seed user and all his/her friends is considered as one *data set*. For each user, we attempt to collect the following information: user name, screen name, user id, profile create time, description (a personal statement), list of followers, list of followings, location, time zone. Meanwhile, for each tweet, we

collect the following: tweet id, post time, tweet location, in-reply-to user id, in-reply-to status id, list of re-tweets (user id and tweet id), tweet content. The attribute list is shown in Table 6.1. Note that not all the attributes are available and accurate for all the users. For example, user location in user profiles is self-generated textual description, where we have seen “Worldwide”, and “Coming Soon Everywhere” etc. Meanwhile, tweet locations are accurate latitudes and longitudes, but they are missing from most of the tweets.

**Table 6.1.** Attributes of user profiles and tweets data.

<b>Profile</b>	
<i>Name</i>	user name
<i>Screen name</i>	displayed screen name
<i>ID</i>	user id
<i>Created at</i>	create time
<i>Description</i>	a personal statement
<i># of followers</i>	number of followers
<i># of following</i>	number of followings
<i>Location</i>	user location
<i>Timezone</i>	user timezone
<b>Tweet</b>	
<i>ID</i>	tweet id
<i>Create Time</i>	tweet post time
<i>Location</i>	tweet location
<i>In reply to (user id)</i>	replied user id
<i>In reply to (status id)</i>	replied tweet id
<i># of retweeted</i>	number of retweets
<i>Retweet (user id)</i>	retweeted user id
<i>Retweet (tweet id)</i>	retweeted tweet id
<i>Content</i>	tweet content

Twitter has enforced mandatory limits on the crawling rate, especially for crawling account-specific information. We have collected 92 data sets – 92 seed users and all their friends. In our data set, each seed user has 245 friends on average. In total, we have collected information of more than 22K users, with

approximately 3 million friendship links, and more than 27 million tweet messages.

We construct six views, as introduced in Section 5.2: content ( $V_1$ ), friendship ( $V_2$ ), common friends ( $V_3$ ), reply ( $V_4$ ), re-tweet ( $V_5$ ), co-reply ( $V_6$ ). For each data set, we have  $n$  users in total, and use  $V_{k,[i,j]}$  to denote the similarity between node  $N_i$  and node  $N_j$  in view  $V_k$ . In particular, for the content view, we have set a threshold of  $\rho = 0.2$  to remove 80% of the low-confidence tags (Pareto principle, a.k.a. 80-20 rule). Meanwhile, all the matrices are normalized by dividing every element by the maximum value in the matrix. Then all diagonal elements are set to one, indicating that self-similarity is always the highest among all. Among all views, we observe that interaction matrices are very sparse in general, as can be seen in Table 6.2. This is largely because of the nature of micro-blogs. Meanwhile, the fact that the microblogs are completely open access also somehow prevented users from explicit interactions.

**Table 6.2.** Sparse Degree of Six Views. The SD is calculated as the ratio between number of zeros in a matrix and the number of elements in that matrix.

Category	View	1-SD
Content	Tag	97.31%
Structure	Friend	2.94%
Structure	Common Friend	51.92%
Interaction	Reply	0.37%
Interaction	Retweet	0.49%
Interaction	Co-Reply	0.06%

### 6.3 Experiment Design

We first implement the Selective Co-trained Spectral Clustering (*SCSC*) algorithm with six views, as described in Section 5.3.3. After the update iterations

in the algorithm, we concatenate  $U_j$ 's of the most informative views to obtain matrix  $V$ , and run  $k$ -means on rows of  $V$  to obtain the final clustering result. In particular, we concatenate the spectral matrices  $U_j$  of the content and structure views to obtain

$$V = [U_{content}, U_{friend}, U_{commonfriend}]. \quad (6.5)$$

We choose these views because they are denser in information. It is worthy to point out that, we did not concatenate interaction views since they may be too sparse to provide accurate information for all users. However, their information have already been (selectively) transferred to the content and structure views through the multi-view algorithms, i.e. CSC and SCSC.

**Baseline approaches.** For comparison, we also employ three baseline approaches on our data: SCAN, SC, CSC.

**SCAN:** Structural Clustering Algorithm for Networks, proposed in [144], is based purely on friendship information of social networks. We run SCAN on the friendship view and compare the results with SCSC.

**Spectral Clustering:** SC utilizes eigenvectors and eigenvalues of similarity matrices (or derived matrices), to find the membership for each vertex. We run spectral clustering on each view separately to obtain eigenvectors  $U_j$ 's. We then column-wise concatenate  $U_j$ 's of the most informative views to obtain matrix  $V$ , and run  $k$ -means on rows of  $V$ .

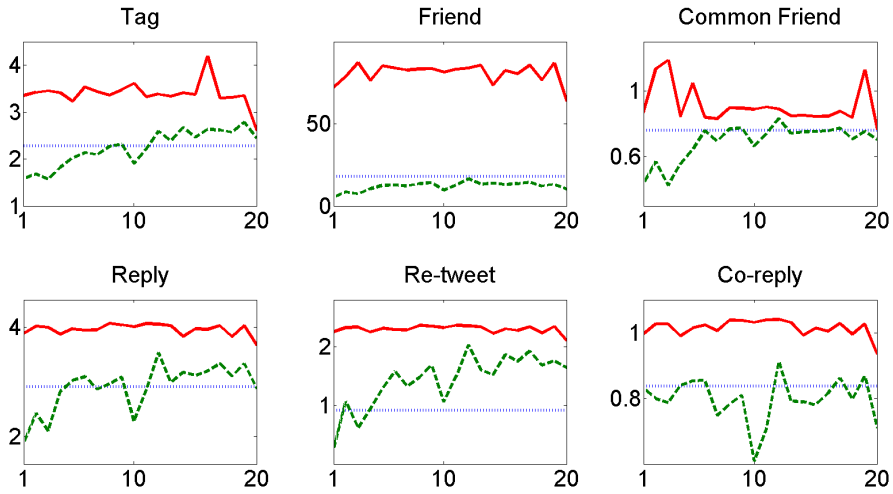
**CSC (Co-trained Spectral Clustering:)** Special type of spectral clustering that exploits multiple sources of information, as mentioned in Section 5.3. We first run CSC on six views. After the update iterations are done, we concatenate  $U_j$ 's of the most informative views to obtain matrix  $V$ . Run  $k$ -means on rows of  $V$ .

In the experiments, we observed similar trends of all approaches when changing the number of clusters  $k$  from 3 to 10. Hence we set  $k = 5$  for all approaches except SCAN. We use default parameters for SCAN.

## 6.4 Results and Performance Analysis

We first examine the performance of *SC*, *CSC* and *SCSC* approaches on six views of one data set, which contains 386 users. We iterate *CSC* and *SCSC* for 20 times and report their normalized similarity ratio (NSR) on each view in Figure 6.1. We see a general trend that *CSC* improves its performance as more iterations are done. This coincides with the spirit of co-trained style algorithms. However, the convergence rate is relatively slow and improvements are not very significant on Tag and Reply views. On the Friend and Co-reply views, *CSC* does not improve the performance of single-view clustering. As we explained before, this may be due to the ignorance of *CSC* on inconsistency between views, especially sparse views. On the other hand, our *SCSC* approach efficiently and significantly boosts the performance after just one or two iterations. On Common Friend view, we observe a degeneration of *SCSC*, which may be because this view has lower correlation to other views.

We further examine the *balance* of the output clusters by each algorithm at their best iterations. An iteration is called the *best iteration of an algorithm* if the algorithm reaches the highest total similarity ratio across all iterations. In Table 6.3 we summarize the size of each group generated by one algorithm. It can be seen that *CSC* encourages more balanced clusters, and both *SC* and *SCSC* output one big cluster. From the algorithm point of view, this may be because *CSC* enforces stronger consistency across views and hence making



**Figure 6.1.** Normalized Similarity Ratio on Six Views. In each figure, the y-axis represents NSR and x-axis represents the number of update iterations. Blue dot curve represents *SC* approach, green dash curve represents *CSC* and red solid curve represents our *SCSC* approach.

**Table 6.3.** Size of Each Cluster in the Clustering Result. *std* stands for standard deviation of all group sizes.

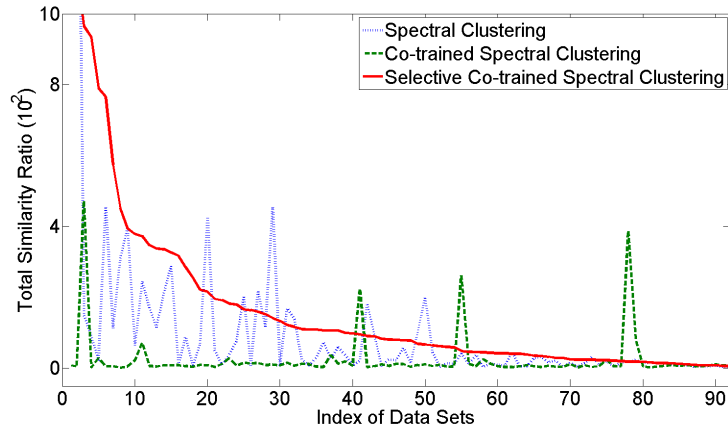
Cluster	1	2	3	4	5	std
<i>SC</i>	8	12	13	25	44	14.6
<i>CSC</i>	16	17	20	24	25	4.04
<i>SCSC</i>	10	10	13	15	54	18.9

the similarity matrix of each view smoother than before. In practice, we think imbalance clusters are acceptable in many applications. For example, a family circle is usually much smaller than a friend circle.

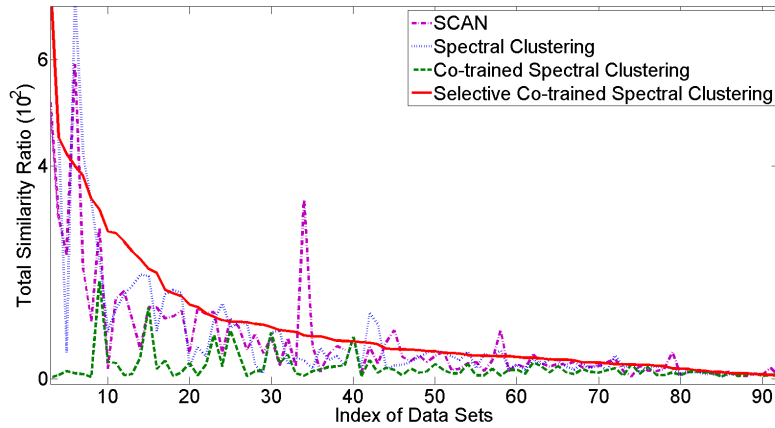
Next we evaluate the performance of all approaches on 92 data sets. The total similarity ratio of each data set is shown in Figure 6.2 (data sets are ordered by the total similarity ratio (TSR) from *SCSC* approach). It is clear that *SCSC* outperforms single-view spectral clustering (*SC*), while *CSC* performs the worst. This coincides with our observation in Figure 6.1, as well as our anal-



ysis of the limitation of *CSC*: enforcing the complete similarity information to transfer from one view to another may contaminate other views and worse the performance. Finally, the Mean TSR (MTSR) for SC is 108.8, MTSR for CSC is 24.8, and MTSR for SCSC is 187.4.



**Figure 6.2.** Total similarity ratio (TSR) of all data sets.



**Figure 6.3.** Normalized Similarity Ratio of all Seed Users on Friend View.

Last, we compare all approaches with SCAN, which is designed for structure-based clustering. Since SCAN filters out *outliers*, we evaluate all approaches only on the non-outlier users, to be fair. Total similarity ratio of all data sets

are shown in Figure 6.3. In particular, the ATSR for SCAN is 71.9, while the updated ATSR for SC is 79.5, ATSR for CSC is 20.9, and ATSR for SCSC is 100.6. It is clear that the performance of *SCAN* is worse than either *SC* or *SCSC*, but better than *CSC*.

## 6.5 Manual Evaluation

Ultimately, the quality of the discovered social circles must be assessed by users. To include users in the loop, we launch a manual evaluation for boundary nodes. As it is impractical to manually examine all users, we attempt to evaluate the nodes that are most doubtful in the clustering process. A boundary node  $N$  represents a user who is clustered by SCSC into cluster  $C_i$ , but is far away from the centroid of the cluster. In particular, we select the boundary node with the largest distance (i.e., least similarity) from each data set. For each selected  $N$ , we identify the cluster  $C_j$ , which is (on average) the closest to  $N$  other than  $C_i$ . We ask users to evaluate if  $N$  should be clustered into  $C_i$  or  $C_j$ .

In the evaluation, we randomly select 5 nodes from cluster  $i$  and  $j$ , respectively. For each selected node  $n_k$ , we display it with  $N$  to an external evaluator, and ask the evaluator to answer the question “Do you think  $N$  should be in the same social circle as  $n_k$ ?” In particular, each evaluator marks the node pair  $(N, n_k)$  with a score from 1 to 5: 5: strongly agree – they belong to the same circle; 4: somewhat agree; 3: neutral; 2: somewhat disagree; and 1: strongly disagree – they do not belong to the same circle. Please note that the evaluation is *blind*. That is, the evaluators do not know whether the pair of nodes are clustered into the same circle or not. In the experiment, we asked 5 external evaluators

(not the authors) to evaluate 60 boundary nodes, which means examining 600 node pairs. As a result, node pairs from the same cluster, as identified by SCSC, earned an average score of 2.63, while node pairs from different circles earned an average score of 2.52.

From the experiment results, we can conclude that our multi-view clustering approach is effective in clustering users’ ego networks into circles. Although the margin appears to be very small, however, we would like to emphasize that we have selected the boundary nodes ( $N$ ) that SCSC is least confident with in the evaluation. Therefore, the result appears to be acceptable.

## 6.6 Keyword Extraction for Clusters

To have a direct perception on the content of the circles, we attempt to find the most unique tags for each cluster. To do so, we calculate the probability of “representativeness” for each tag in each cluster. Intuitively, a tag with larger bias towards a cluster better represents the content of the cluster. Formally, the probability of tag  $t$  in cluster  $C$ , denoted by  $P(t|C)$  can be defined as:

$$P(t|C) = \frac{\sum_{i \in C} tf_{norm}(i, t)}{|C|}$$

$$tf_{norm}(i, t) = \frac{tf(i, t)}{\max\{tf(i, t) | t \in \mathbf{T}_i\}}$$

$tf(i, t)$  is the frequency of tag  $t$  in user  $i$ ’s content, and the  $\max$  function returns the largest frequency for all tags in  $i$ ’s content. To find the most representative tags in a certain cluster, we propose to utilize *Kullback-Leibler Divergence* (KL-divergence). In particular, we first construct 2 discrete probability distributions

$P_t(i)$  and  $Q_t(i)$  as.

$$P_t(i) = \frac{P(t|C_i)}{\sum_{C_i \in \mathbb{C}} P(t|C_i)}$$

$$Q_t(i) = \frac{1}{|\mathbb{C}|}$$

We further calculate the bias of tags:

$$Bias_{KL}(t) = \sum_i (P_t(i) \ln \frac{P_t(i)}{Q_t(i)})$$

For each cluster  $C_i$ , we can find the tags with largest  $Bias_{KL}$ , and having  $\max P(t|C)$  in  $C_i$  as the representative tags for  $C_i$ . The top 3 tags for 5 clusters of a randomly selected data set are shown in Table 6.4. For clusters having less than 3 representative tags, we just show all of them. From this example, we can see different groups have different topics. For instance, group 2 is leaning towards entertainment, group 4 seems to be interested in health care information, while group 5 is quite technical. The extracted content has been confirmed by our manual examination of the circles. As a result, we can actually perceive the separations of different circles in the ego network.

**Table 6.4.** Representative tags for clusters of a seed

Cluster	Representative Tags
$C_1$	Human,Sleep
$C_2$	Valentine's Day,Dance,Sport
$C_3$	Ireland,Beer,Coffee
$C_4$	Social media,Health,Cancer
$C_5$	Yahoo!,WHATS'On (Software),Android

## 6.7 Discussions

**Computational complexity of SCSC.** Compared with CSC, the SCSC algorithm introduces an extra  $k$ -means clustering procedure (step 4 in Algorithm 2) when updating the similarity matrices. Theoretically, the dominant computational complexity of  $k$ -means is  $O(nk^2)$ , where  $n$  is the number of friends of a seed user, and  $k$  is the number of clusters expected. In application of personalized social network, we may expect that both  $n$  and  $k$  are not too large: a typical seed user we crawled has around 200 friends, and it is quite unlikely for one to create and maintain more than 10 circles of his or her friends. For more information about the efficiency of  $k$ -means, readers are referred to [5]. In practice, we observe an average time around one minute to run *SCSC* with 20 update iterations for one seed user. Moreover, the extra computational burden brought by *SCSC* can be released from multiple aspects. As shown by our empirical study, *SCSC* converges much faster than *CSC*. Hence only a few number of extra  $k$ -means clustering will be introduced. We may also update all views in parallel within one iteration, freeing the extra computational burden from being accumulated over views. As evidenced in our empirical study, by trading certain efficiency for adaptiveness, *SCSC* significantly improved the clustering performance.

**Non-overlapping vs. overlapping circles.** In the ego network  $E_S$  of seed  $S$ , if we allow any user  $N_i$  to belong to multiple circles, it is regarded as *overlapping circles*. Meanwhile, if each user  $N_i$  is allowed in exactly one circle, it is *non-overlapping circles*. In the literature, both types of circles have been used. In this dissertation, we select non-overlapping circles for two major reasons. First, our approach is primarily motivated by privacy protection and information bound-

ary enforcement in social networks. When two social circles in the ego network overlaps, the overlapping users observe information from both circles. Such users may also easily violate the boundaries by moving information from its origin circle to the other overlapping circles. This is the online version of “social gossip”. On the other hand, in theory, overlapping and non-overlapping circles are essentially equivalent. That is, two overlapping circles  $A$  and  $B$  could be converted to three non-overlapping circles  $A \cap B; A \setminus B; B \setminus A$ . Contained circles  $A \subset B$  could be converted to two non-overlapping circles  $A; B \setminus A$ .

In reality, there exist users who have close connections with multiple circles (for now, we consider two circles). They actually cause difficulties in our clustering approach. When we manually examine the clustering results, we discover different outcomes: (1) in most cases, such nodes are assigned into the one circle, with which he/she has stronger connections; (2) when there are many nodes connecting to two circles and the nodes strongly connect to each other, we may end up identifying three circles; (3) it is also possible that we discover one combined circle, due to the existence of multiple users in the overlap.

**Applications of social circles.** As suggested in [125,126], social circles are used to protect information privacy, by delivering messages to designated circles and enforcing circle boundaries. Automatically clustered circles are presented to users, so that they could further re-organize and configure such circles. In socialization, messages are posted to the selected circles. Meanwhile, social circle enforcement becomes particularly challenging when some social networking sites allows breaches in privacy protection (e.g., when users are allowed to “re-share” private posts of their friends). However, those issues are outside of the scope of this dissertation.

On the other hand, the discovered social circles could be used to improve the efficiency of ad delivery, targeted advertising, and opinion mining in social groups. Social circles could also be used to study users' socialization behavior and social network information flow. If temporal information is added to the data, we can extend our model to further study the development of social circles and evolution of ego networks.

## Chapter 7

# Automatic Circle Recommendation and Proof-of-Concept Implementation

In this chapter, we will present our proof-of-concept implementation of automatic social circle discovery and message recommendation system to demonstrate the practicality of using social circles for privacy protection.

### 7.1 Automatic Circle Recommendation

In our system, we implement the automatic circle recommendation functionality: when users enter a message, it will recommend a circle to post the message. But the question is how to make such recommendation. In this section, we first propose 3 theoretical approaches for circle recommendation to fulfill this function, and continue to present our implementation in details in the next section.



Intuitively, social circle recommendation could be based on: (1) user perception: infer what the user would do with each message (2) attributes: extract attributes from each message and manage attributes in each circle (3) content: calculate content similarities between a message and circles and select the circle with the highest similarity. Here we briefly discuss all 3 approaches.

### 7.1.1 User-perception-based Circle Recommendation

Ultimately, we expect the circle recommendation mechanism to be consistent with user perception. That is, when a user enters a message, the circle recommendation mechanism attempts to guess which circle the user would choose and make the suggestion accordingly. As an example, if a soccer fan wants to post a message: "Germany vs Argentina, the world cup final would be great!!!", he/she may more likely choose a circle composed of other soccer fans or sport lovers. So for this approach, the system aims to reproduce user perceptions precisely.

However, it's extremely difficult to capture and model user perception, to do so we have to: (1) fully understand the message content, (2) fully understand the user perception, and (3) fully understand the circles. All these tasks require machine learning or artificial intelligence techniques which are still open problems and challenging to solve. In this project, we use an approximation. We assume that a user wants to discuss a topic with a group of friends that he/she has discussed similar topics with. That is, for a new message  $m$ , a user is more likely to select a circle where he/she has already posted messages similar to  $m$ .

In addition, in this approach, we always allow users to override the system

suggestions, which further ensures user perception.

### 7.1.2 Attribute-based Circle Recommendation

For this approach, we assume that attackers can extract attributes from message. We have already claimed in Chapter 3, attributes need to be protected, and intuitively, we can use attributes to suggest circles for newly inputted messages while preserving their privacy. A message  $m$  could be delivered to a circle  $C$  without compromising user privacy, if: (1) the attributes covered in  $m$  is already in  $C$ ; or (2) the user explicitly authorizes the attributes to be delivered to  $C$ .

The main idea of this approach is to identify privacy related attributes from unstructured messages (in other words, free text, such as Status, Diaries, etc.), such as Education, Age, Job, Location, etc. As an example, if a user writes a message, "The traffic jam of Los Angeles highway is too bad to go to work on time!!!," we can know several privacy attributes: Location="Los Angeles", HavingJob="true". Another example is from the message, "Today I graduated from High School," we can probably get Education="High School", AgeGroup="15-20".

Users can specify which attributes can be disseminated to which circles. For example, Age attributes are available to be sent to "Family" circles. When unspecified attributes are intended to be disclosed to a circle, the system will warn users. When attributes are already visible within a circle, we allow them to be distributed to the same circle again (e.g., when the user has already disclosed that he/she has a new born baby in a circle, he/she could send more messages about the baby to the same circle). When users attempt to disclose new

attributes to the circle, we firstly identify the attributes from user input and based on some analysis tools such as machine learning models to learn from user specifications, and make decisions on whether to disclose the attributes to a circle.

However, to implement the attribute-based circle recommendation is extremely difficult. The attribution (also called “attribute discovery” or “attribute mining”) is still a difficult and unsolved problem. To extract attributes from unstructured user input, we can use natural language processing techniques based on machine learning or probabilistic models, but this is extremely hard to obtain and still an open problem in the research community. As a result, this approach is beyond the scope of this dissertation, and we choose not to implement based on this method.

### 7.1.3 Content-based Circle Recommendation

As we have discussed, user-perception-based and attribute-based circle recommendation approaches are theoretically sound. However, practically they are both extremely challenging to implement. As a compromise, we developed content-based circle suggestion, which is based on a relatively fuzzy concept, “content”. In this approach, if a new message is similar to the content of a circle, we assume that it could be delivered to this circle. In practice, we use tag-based and term-distribution-based similarity.

In our implementation, we use a tag-based Bayesian-probability method. Our goal is to calculate  $P(c_i|m)$ , i.e. the probability of a message  $m$  belonging to a circle  $c_i$ . The tag-based idea is to firstly tag the message using TagMe and obtain several tags with their  $\rho$  (tag goodness indicator). Then we can calculate

$P(c_i|m)$  as:

$$P(c_i|m) = \sum_j P(c_i|t_j)\rho_j$$

which means that for each tag, multiply  $\rho$  with its probability belonging to  $c_i$  and sum the values for all the tags. For  $P(c_i|t_j)$ , we can use Bayesian theory to calculate it:

$$P(c_i|t_j) = \frac{P(t_j|c_i)P(c_i)}{P(t_j)}$$

where

$$P(t_j) = \sum_{c_i \in \mathbb{C}} P(t_j|c_i)P(c_i)$$

and we can use the equation in 6.6 to calculate  $P(t_j|c_i)$ . If we assume the probability for each circle is the same,

$$P(c_i) = \frac{1}{|\mathbb{C}|}$$

where  $\mathbb{C}$  represents the number of circles for a seed user. Then we can calculate  $P(c_i|t_j)$  and  $P(c_i|m)$ , so as to find the circle with the highest probability for circle suggestion.

## 7.2 Implementation

In this section, we will present the details of our proof-of-concept implementation of the social circle model.

### 7.2.1 System Architecture

The architecture of our system is shown in Figure 7.1. It is a 2 layer client server connection architecture. After users enter the starting page, they input

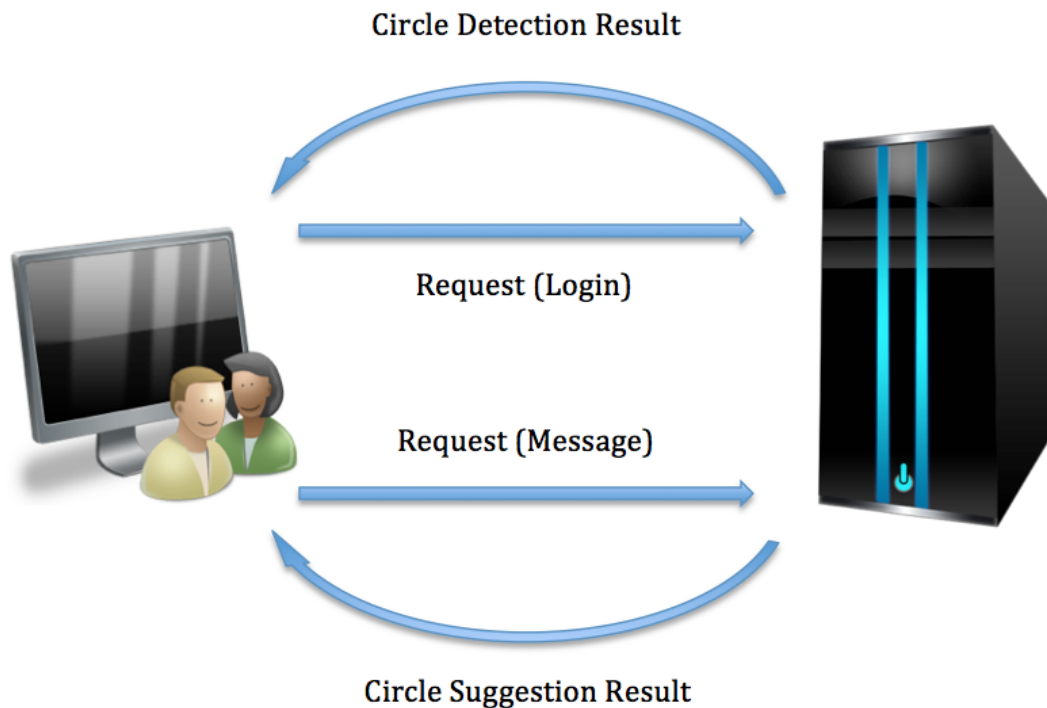
their login information (since for Twitter users, we don't actually need their login information to get their data, for convenience, we use user names to represent their login information) and send a request to the server for automatic circle detection results. The server will receive user requests with their login information (names), and render the result web page using JSP techniques. The created result page will be send to users, and when users receive the result page, their web browsers will display the result and execute the JavaScript code accordingly. Similarly, when users are on the circle detection result page, they can enter a message and request for the automatic circle suggestion result. The server will receive this request with the message, and create a web page with the suggestion result. The created page will be sent to users, and their browsers will display the automatic circle suggestion result as instructed.

For this implementation, we mainly used web-based techniques such as: HTML and HTML CSS, JSP, JavaScript and DOM, JQuery and JQuery UI. HTML and HTML CSS are used to build up the web pages and add effects to them. JSP is used to combine Java with web pages. JavaScript and DOM are used to dynamically change web page elements, and JQuery and JQuery UI are used to realize special effects such as input text auto-complete and accordion lists.

### **7.2.2 Search Interface**

The search interface is the starting point of our system, where users can enter their names (simulated as the login process). As shown in Figure 7.2. Users can input names in the text field after "Name".

By JQuery UI auto-complete function, we can suggest possible names in our system to users, as shown in Figure 7.3. The back-end suggestion lists are



**Figure 7.1.** System Architecture

extracted from all the seed names in our data sets.

### 7.2.3 Automatic Circle Detection Result Presentation

After users click the “Enter” button on the keyboard, the circle detection result obtained by our multi-view clustering method will be presented on the next page, as show in Figure 7.4. The system will read the corresponded seed user’s circle detection result file using JSP.

On this page, the profile image of the logged user is shown and also its name and Twitter ID. Each circle is listed in an accordion group titled as circle id with its corresponded key words extracted using techniques presented in 6.6. JSP is used to read the key word file, and by the JQuery UI accordion,



**Figure 7.2.** System Interface



**Figure 7.3.** Name Suggestion

when users click on one of the circle titles, detailed information is faded in, as presented in Figure 7.5.



**Figure 7.4.** Result Presentation



**Figure 7.5.** Circle Detail

This frame demonstrates all the users' profile images from this circle. And when you move the mouse over each image, it will fade in the corresponded



user's name and Twitter ID, as shown in Figure 7.6.

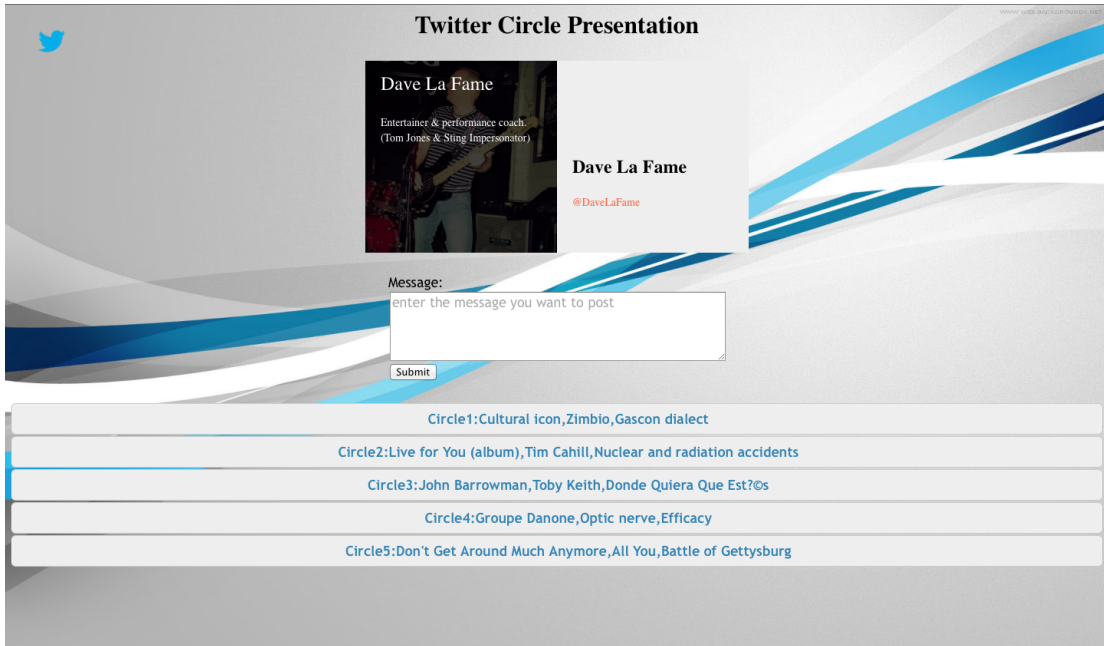


**Figure 7.6.** Detailed Circle User Info

Similarly, when you move the mouse over the seed user name text field, the detailed information of the seed user is faded in with its name and description, as shown in Figure 7.7.

All the images are inputted by including their source file urls into HTML, which are obtained by Twitter4J Java libraries.

As shown in these figures, there is also an input message field on this page, where users can enter messages they want to post, and when the "Submit" button is clicked, the automatically designated circle will be highlighted on the next page.



**Figure 7.7.** Seed User Info

## 7.2.4 Automatic Message Designation

After users enter a message and click the submit button, as shown in Figure 7.8, the message designation page will be directed. The directed JSP file will run the Java code implementing the circle suggestion technique presented in Section 7.1.3, which will firstly connect to TagMe in order to tag the input message and then run the Bayesian Probability method to identify the appropriate circle. The system recommended circle is highlighted with transparent deep sky blue, as shown in Figure 7.9.



Figure 7.8. User Input Message



Figure 7.9. Designation Result

# Chapter 8

## Conclusion

We have proposed thorough research about user privacy online. In this dissertation, we first study the vulnerabilities of user attributes and contents, in particular, the identifiability of the users when the adversary learns a small piece of information about the target. We further employ an information theory based approach to quantitatively evaluate the threats of attribute-based re-identification. We have shown that large portions of users with online presence are highly identifiable.

The notion of privacy as control and information boundary has been introduced by the user-oriented privacy research community, and partly adopted in commercial social networking platforms. However, such functions are not widely accepted by the users, mainly because it is tedious and labor-intensive to manually assign friends into such circles. To tackle this problem, we introduce a social circle discovery approach using multi-view clustering. The multi-view clustering technique is based on 3 observations: users belonging to the same circle are very likely to: (1) be friends and share many common friends, (2) be interested in similar content, and (3) have more interactions with each

other. From these 3 observations, we model 6 views and propose a selective co-trained spectral clustering technique to combine these different aspects of information together. Our experiment is performed on real-world Twitter data, including approximately 3 million friendship links, and more than 27 million tweet messages. From the presented result, multi-view clustering renders more accurate circle detection than single-view clustering, and our method gains significantly higher similarity ratio than the original co-trained spectral clustering technique. As a result, by utilizing our automatic social circle detection technique, users can get natural “information boundaries” for privacy control, while not losing the practicality and convenience of the online social network environment.

At last, we build an automatic social circle detection and suggestion proof-of-concept system, using multiple state-of-the-art web techniques, such as JSP, JavaScript, JQuery, etc. In the goal of making a user-friendly demonstration system, users can log in and obtain their circle detection results, and after they enter a message, the system can suggest a social circle to post the message. From this demonstration, we can show the usefulness of our techniques based on the social circle model. Users can get relatively accurate grouping of their friends, which is created automatically by the system. They can post messages to appropriate social circles, which are also automatically suggested by the system. By this function, users can control the accessibility of their private information, while still preserving the ability to socialize normally.

In conclusion, in this dissertation, we have studied the vulnerability of users’ private information online, proposed an automatic social circle detection method to solve this problem, and presented a proof-of-concept implementation, all in

the goal of better understanding online social network user privacy and helping build a secure socializing environment for them.

# References

- [1] M. S. Ackerman and L. Cranor. Privacy critics: Ui components to safeguard users' privacy. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 258–259, New York, NY, USA, 1999. ACM.
- [2] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2003.
- [3] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.
- [4] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, , and A. Zhu. k-anonymity: Algorithms and hardness. Technical report, Stanford University, 2004.
- [5] K. Alsabti, S. Ranka, and V. Singh. An efficient k-means clustering algorithm. 1997.
- [6] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of ACM international conference on World Wide Web*, pages 181–190, 2007.

- [7] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. *SIGCOMM Comput. Commun. Rev.*, 39(4):135–146, 2009.
- [8] M. Balduzzi, C. Platzner, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In S. Jha, R. Sommer, and C. Kreibich, editors, *Recent Advances in Intrusion Detection*, volume 6307 of *Lecture Notes in Computer Science*, pages 422–441. Springer Berlin / Heidelberg, 2010.
- [9] S. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2007.
- [10] S. B. Barnes. A privacy paradox: Social networking in the united states. *First Monday [Online]*, 11(9), 2006.
- [11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.
- [12] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.
- [13] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web, WWW '09*, pages 551–560, New York, NY, USA, 2009. ACM.



- [14] G. Blosser and J. Zhan. Privacy Preserving Collaborative Social Network. In *International Conference on Information Security and Assurance (ISA)*, pages 543 – 548, April 2008.
- [15] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [16] A. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Ubicomp*, 2010.
- [17] S. Byers, L. F. Cranor, and D. Kormann. Automated analysis of p3p-enabled web sites. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 326–338, New York, NY, USA, 2003. ACM.
- [18] K. Caine, L. G. Kisselburgh, and L. Lareau. Audience visualization influences disclosures in online social networks. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems, CHI EA '11*, pages 1663–1668, New York, NY, USA, 2011. ACM.
- [19] P. Cashmore. Privacy is dead, and social media hold smoking gun. CNN, October 2009.
- [20] J. Caverlee and S. Webb. A large-scale study of myspace: Observations and implications for online social networks. In *Proceedings of the International Conference on Weblogs and Social Media*, 2008.

- [21] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee. @phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *ASONAM*, 2012.
- [22] J. Chang, J. Boyd-Graber, and D. M. Blei. Connections between the lines: augmenting social networks with text. *KDD'09*, pages 169–178, 2009.
- [23] T. Cheng, X. Yan, and K. C.-C. Chang. Entityrank: searching entities directly and holistically. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 387–398. VLDB Endowment, 2007.
- [24] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *ACM CIKM*, pages 759–768, 2010.
- [25] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
- [26] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90, New York, NY, USA, 2005. ACM.
- [27] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification. Technical report, World Wide Web Consortium Recommendation, 2002. Available: <http://www.w3.org/TR/P3P/>.

- [28] L. F. Cranor, P. Guduru, and M. Arjula. User interfaces for privacy agents. *ACM Trans. Comput.-Hum. Interact.*, 13(2):135–178, 2006.
- [29] L. F. Cranor and L. Lessig. *Web Privacy with P3p*. O’Reilly & Associates, Inc., Sebastopol, CA, USA, 2002.
- [30] M. Culnan and J. Bies. Consumer privacy: Balancing economic and justice considerations. *Journal of Social Issues*, 59(2), 2003.
- [31] d. m. boyd and N. B. Ellison. Social network sites: definition, history and scholarship. *Journal of Computer-Mediated Communication*, 13:210–230, 2008.
- [32] danah m. boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2008.
- [33] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *USENIX Security Symposium*, 2004.
- [34] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering with multi-layer graphs: A spectral perspective. 2011.
- [35] B. Dubow. Confessions of ‘Facebook stalkers’. USA Today, March 2007.
- [36] E. Elmacioglu and D. Lee. On six degrees of separation in dblp-db and more. *SIGMOD Rec.*, 34(2):33–40, 2005.
- [37] M. Erdmann, T. Takeyoshi, G. Hattori, and C. Ono. Extraction and annotation of personal cliques from social networks. In *Applications and the*

*Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on*, pages 172–177. IEEE, 2012.

- [38] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91 – 134, 2005.
- [39] G. Eysenbach and J. E. Till. Ethical issues in qualitative research on internet communities. *BMJ*, 323:1103–1105, 2001.
- [40] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *International World Wide Web conference(WWW)*, 2010.
- [41] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360. ACM, 2010.
- [42] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *Software, IEEE*, 29(1):70–75, 2012.
- [43] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, 2000.
- [44] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

- [45] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *MobiSys*, 2008.
- [46] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1), 1997.
- [47] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HYPertext '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 225–234, New York, NY, USA, 1998. ACM.
- [48] E. Gilbert and K. Karahalios. Predicting tie strength with social media. *CHI'09*, pages 211–220, 2009.
- [49] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 99:7821–7826, June 2002.
- [50] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Commun. ACM*, 42(2):39–41, 1999.
- [51] P. Golle. Revisiting the uniqueness of simple demographics in the us population. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80, New York, NY, USA, 2006. ACM.
- [52] D. Gross. Social networks and kids: How young is too young? CNN, November 2009.

- [53] R. Gross, A. Acquisti, and I. H. John Heinz. Information revelation and privacy in online social networks (the facebook case). In *Proceedings of ACM workshop on Privacy in the electronic society*, pages 71–80, 2005.
- [54] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, 2008.
- [55] J. He and W. W. Chu. Protecting private information in online social networks. In *Intelligence and Security Informatics*, pages 249–273, 2008.
- [56] J. He, W. W. Chu, and Z. Liu. Inferring privacy information from social networks. In *IEEE International Conference on Intelligence and Security Informatics*, pages 154–165, 2006.
- [57] J. M. Hoffman and C. H. Wiggins. A bayesian approach to network modularity. *Phys. Rev. Lett.*, 100:258701, 2008.
- [58] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189, New York, NY, USA, 2004. ACM.
- [59] B. A. Huberman, E. Adar, and L. R. Fine. Valuating privacy. *IEEE Security and Privacy*, 3(5):22–25, 2005.
- [60] M. Irvine. Social network users overlook privacy pitfalls. *USA Today*, April 2008.
- [61] S. Jones and E. O’Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *Proceedings of the Sixth*

*Symposium on Usable Privacy and Security, SOUPS '10*, pages 9:1–9:13, New York, NY, USA, 2010. ACM.

- [62] S. Jones and E. O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. *SUPS*, July 2010.
- [63] A. Keen. Is social networking bad for our children? *Telegraph.co.uk*, October 2009.
- [64] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49:291–307, 1970.
- [65] J. Kirk. Developer finds major coding errors in facebook, myspace. *Computer World*, November 2009.
- [66] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. In *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks*, pages 7–12, New York, NY, USA, 2009. ACM.
- [67] J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, Aug. 2009.
- [68] A. Kumar and H. Daumé III. A co-training approach for multi-view spectral clustering. In *28th International Conference on Machine Learning, Bellevue, Washington*, pages 393–400, 2011.
- [69] A. Kumar, P. Rai, and H. D. Iii. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

- [70] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, 2006.
- [71] C. Lampe, N. Ellison, and C. Steinfield. A face(book) in the crowd: social searching vs. social browsing. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 167–170, New York, NY, USA, 2006. ACM.
- [72] Y. G. Le Gall, A. J. Lee, and A. Kapadia. Plexc: a policy language for exposure control. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies, SACMAT '12*, pages 219–228. ACM, 2012.
- [73] S. Lederer, J. I. Hong, A. K. Dey, and J. A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454, 2004.
- [74] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 915–924, 2008.
- [75] K. Lewis, J. Kaufman, and N. Christakis. The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication*, 14(1):79–100, 2008.
- [76] F. Li, J. Y. Chen, X. Zou, and P. Liu. New privacy threats in healthcare informatics: When medical records join the web. In *Proceedings of the 9th*



*International Workshop on Data Mining in Bioinformatics (BIOKDD)*, July 2010.

- [77] H. Li, Z. Nie, W.-C. Lee, L. Giles, and J.-R. Wen. Scalable community discovery on textual data with relations. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1203–1212. ACM, 2008.
- [78] N. Li, T. Li, and S. Venkatasubramanian.  $t$ -closeness: Privacy beyond  $k$ -anonymity and  $l$ -diversity. In *Proceedings of the 23rd International Conference on Data Engineering*, pages 106–115, 2007.
- [79] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *CIKM*, 2011.
- [80] Z. Li, D. Zhou, Y.-F. Juan, and J. Han. Keyword extraction for social snippets. In *Proceedings of the 19th international conference on World wide web*, pages 1143–1144. ACM, 2010.
- [81] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community detection in incomplete information networks. *WWW'12*, April 2012.
- [82] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 685–694, 2008.
- [83] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD*, pages 93–106, 2008.

- [84] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preserving in social networks against sensitive edge disclosure. Technical Report CMIDA-HiPSCCS 006-08, University of Kentucky, 2008.
- [85] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, IMC '11*, pages 61–70, New York, NY, USA, 2011. ACM.
- [86] Y. Liu, A. N. Mizil, and W. Gryc. Topic-link lda: Joint models of topic and author community. *ICML'09*, 382:665–672, 2009.
- [87] B. Luo and D. Lee. On protecting private information in social networks: A proposal. In *Workshop on Modeling, Managing, and Mining of Evolving Social Networks (M3SN) - in conjunction with IEEE ICDE*, 2009.
- [88] A. Macfarlane. *History, anthropology and the study of communities*. 1977.
- [89] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3, 2007.
- [90] M. Madejski, M. Johnson, and S. M. Bellovin. The failure of online social network privacy settings. Technical Report CUCS-010-11, Columbia University, 2011.
- [91] H. Mao, X. Shuai, and A. Kapadia. Loose tweets: an analysis of privacy leaks on twitter. In *Proceedings of the 10th annual ACM workshop on Privacy*

*in the electronic society*, WPES '11, pages 1–12, New York, NY, USA, 2011. ACM.

- [92] A. Mazzia, K. LeFevre, and E. Adar. The pviz comprehension tool for social network privacy settings. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 13:1–13:12, New York, NY, USA, 2012. ACM.
- [93] J. McAuley and J. Leskovec. Discovering social circles in ego networks. *TKDD'13*, 2012.
- [94] C. McCarty. Structure in personal networks. *Journal of Social Structure*, 3, 2002.
- [95] C. McCarty. Structure in personal networks. *Journal of Social Structure*, 3(1):Online, 2002.
- [96] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 251–260. ACM, 2010.
- [97] S. Motahari, C. Manikopoulos, R. Hiltz, and Q. Jones. Seven privacy worries in ubiquitous social computing. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 171–172, New York, NY, USA, 2007. ACM.
- [98] G. Nagesh. Social networking sites a treasure trove for identity thieves. NextGov, November 2009.

- [99] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [100] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proc Natl Acad Sci USA*, 104:9564–9569, June 2007.
- [101] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, Feb 2004.
- [102] X. Ni, X. Quan, Z. Lu, L. Wenyin, and B. Hua. Short text clustering by finding core terms. *Knowledge and information systems*, 27(3):345–365, 2011.
- [103] F. H. Norris. Epidemiology of trauma: frequency and impact of different potentially traumatic events on different demographic groups. *Journal of consulting and clinical psychology*, 60(3):409, 1992.
- [104] P. Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review*, 57(6):1701 – 1777, 2010.
- [105] K. E. Pettigrew. Waiting for chiropody: contextual results from an ethnographic study of the information behaviour among attendees at community clinics. *Information Processing & Management*, 35(6):801–817, 1999.
- [106] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM, 2008.
- [107] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper. Topic discovery based on text mining techniques. *Information processing & management*, 43(3):752–768, 2007.

- [108] S. Preibusch, B. Hoser, S. Gürses, and B. Berendt. Ubiquitous social networks - opportunities and challenges for privacy-aware user modelling. In *Proceedings of Workshop on Data Mining for User Modeling*, 2007.
- [109] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83:066114, June 2011.
- [110] H. Qian and C. R. Scott. Anonymity and self-disclosure on weblogs. *Journal of Computer-Mediated Communication*, 12(4), 2007.
- [111] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998.
- [112] B. Rolan. 18-year-old used trickery to solicit girls on facebook, police say. *Times-Georgian*, November 2009.
- [113] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. *WWW'13*, pages 1089–1098, May 2013.
- [114] M. Sachan, D. Contractor, T. A. Faruque, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. *WWW'12*, pages 331–340, April 2012.
- [115] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao. Understanding and capturing people’s privacy policies in a mobile social networking application. *Personal Ubiquitous Comput.*, 13(6):401–412, 2009.
- [116] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

- [117] R. Schlegel, A. Kapadia, and A. J. Lee. Eyeing your exposure: quantifying and controlling information sharing for improved privacy. In *Proceedings of the Seventh Symposium on Usable Privacy and Security, SOUPS '11*, pages 14:1–14:14. ACM, 2011.
- [118] J. Scripps and P.-N. Tan. Clustering in the presence of bridge-nodes. In *Proceedings of the 6th Siam International Conference on Data Mining*, volume 124, page 270. Siam, 2006.
- [119] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8), 2000.
- [120] P. Shi, H. Xu, and Y. Chen. Using contextual integrity to examine interpersonal information boundary on social network sites. In *CHI*.
- [121] K. Singh, S. Bhola, and W. Lee. xBook: Redesigning Privacy Control in Social Networking Platforms. In *Proceedings of 18th USENIX Security Symposium*, August 2009.
- [122] L. Singh and J. Zhan. Measuring topological anonymity in social networks. In *GRC '07: Proceedings of the 2007 IEEE International Conference on Granular Computing*, page 770, Washington, DC, USA, 2007. IEEE Computer Society.
- [123] P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 655–664, New York, NY, USA, 2008. ACM.

- [124] M. M. Skeels and J. Grudin. When social networks cross boundaries: a case study of workplace use of facebook and linkedin. *GROUP'09*, May 2009.
- [125] A. Squicciarini, S. Karumanchi, D. Lin, and N. DeSisto. Identifying hidden social circles for advanced privacy configuration. *Computers & Security*, 2013.
- [126] A. Squicciarini, D. Lin, S. Karumanchi, and N. DeSisto. Automatic social group organization and privacy management. In *CollaborateCom*, 2012.
- [127] A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede. A3p: adaptive policy prediction for shared images over popular content sharing sites. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 261–270, New York, NY, USA, 2011. ACM.
- [128] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842. ACM, 2010.
- [129] A. Sun. Short text classification using very few words. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1145–1146. ACM, 2012.
- [130] L. Sweeney. Uniqueness of simple demographics in the u.s. population, 2000.

- [131] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
- [132] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.
- [133] L. Tang, X. Wang, and H. Liu. Community detection in multi-dimensional networks. Technical report, DTIC Document, 2010.
- [134] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 1016–1021. IEEE, 2009.
- [135] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM, 2007.
- [136] H. Tavani. Philosophical theories of privacy: Implications for an adequate online privacy policy. *Metaphilosophy*, 38(1), 2007.
- [137] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. 2013.
- [138] X. Wang, H. Liu, and W. Fan. Connectiong users with similar interests via tag network inference. *CIKM'11*, pages 1019–1024, October 2011.
- [139] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor. “i regretted the minute i pressed share”: a qualitative study of regrets



- on facebook. In *Proceedings of the Seventh Symposium on Usable Privacy and Security, SOUPS '11*, pages 10:1–10:16, New York, NY, USA, 2011. ACM.
- [140] F. Wei, W. Qian, C. Wang, and A. Zhou. Detecting overlapping community structures in networks. *World Wide Web*, 12(2):235–261, 2009.
- [141] A. F. Westin. Social and political dimensions of privacy. *Journal of social issues*, 59(2):431–453, 2003.
- [142] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 223–238. IEEE, 2010.
- [143] Q. Xiao, H. H. Aung, and K.-L. Tan. Towards ad-hoc circles in social networking sites. In *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks, DBSocial '12*, pages 19–24, New York, NY, USA, 2012. ACM.
- [144] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833. ACM, 2007.
- [145] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.

- [146] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. *KDD'09*, pages 927–936, 2009.
- [147] Y. Yang, J. Lutes, F. Li, B. Luo, and P. Liu. Stalking online: on user privacy in social networks. In *ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2012.
- [148] H. Yildiz and C. Kruegel. Detecting social cliques for automated privacy control in online social networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 353–359, march 2012.
- [149] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *SIAM International Conference on Data Mining (SDM)*, 2008.
- [150] A. Yuksel, M. Yuksel, and A. Zaim. An approach for protecting privacy on social networks. In *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, pages 154–159, aug. 2010.
- [151] J. Zhan, G. Blosser, C. Yang, and L. Singh. Privacy-Preserving Collaborative Social Networks. In *Pacific Asia Workshop on Intelligence and Security Informatics (PAISI)*, 2008.
- [152] Y. Zhang and D.-Y. Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. *KDD*, August 2012.
- [153] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *International Workshop on Privacy, Security, and Trust in KDD (PinKDD)*, pages 153–171, 2008.

- [154] E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *18th International World Wide Web conference (WWW)*, April 2009. Earlier version appears as CS-TR-4926.
- [155] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 24th International Conference on Data Engineering (ICDE)*, April 2008.
- [156] B. Zhou, J. Pei, and W. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, 2008.
- [157] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. *WWW'06*, pages 173–182, 2006.
- [158] W. Zhou, H. Jin, and Y. Liu. Community discovery and profiling with social messages. *KDD'12*, pages 388–396, August 2012.
- [159] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous. *KDD'13*, August 2013.