Understanding Social Media Users via Attributes and Links

by

Mohammad Ali Abbasi

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved November 2014 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Hasan Davulcu
Jieping Ye
Nitin Agarwal

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

With the rise of social media, hundreds of millions of people spend countless hours all over the globe on social media to connect, interact, share, and create user-generated data. This rich environment provides tremendous opportunities for many different players to easily and effectively reach out to people, interact with them, influence them, or get their opinions. There are two pieces of information that attract most attention on social media sites, including user preferences and interactions. Businesses and organizations use this information to better understand and therefore provide customized services to social media users. This data can be used for different purposes such as, targeted advertisement, product recommendation, or even opinion mining. Social media sites use this information to better serve their users.

Despite the importance of personal information, in many cases people do not reveal this information to the public. Predicting the hidden or missing information is a common response to this challenge. In this thesis, we address the problem of predicting user attributes and future or missing links using an egocentric approach. The current research proposes novel concepts and approaches to better understand social media users in twofold including, a) their attributes, preferences, and interest, and b) their future or missing connections and interactions. More specifically, the contributions of this dissertation are (1) proposing a framework to study social media users through their attributes and link information, (2) proposing a scalable algorithm to predict user preferences; and (3) proposing a novel approach to predict attributes and links with limited information. The proposed algorithms use an egocentric approach to improve the state of the art algorithms in two directions including improving the prediction accuracy, and increasing the scalability of the algorithms.

To my parents and family

ACKNOWLEDGMENTS

Foremost, I would like to express my deepest thanks to my supervisor Professor Huan Liu, who has been invaluable mentor guiding me in my research. His patience, encouragement, and immense knowledge were key motivations throughout my PhD. I am truly thankful for his steadfast integrity and selfless dedication to both my personal and academic development. I cannot think of a better supervisor to have. He is a mentor and friend, from whom I have learned the vital skill of disciplined critical thinking. I would also like to thank my committee members, Dr. Hasan Davulcu, Dr. Jieping Ye, and Dr. Nitin Agarwal, who have supported me further in my endeavors.

I would like to thank my colleagues at the Data Mining and Machine Learning Lab at Arizona State University for their constructive criticism and helpful suggestions regarding this work as well as for their support. I would particularly like to thank Salem Alelyani, Geoffrey Barbier, Ghazaleh Beigi, Huiji Gao, Pritam Gundecha, Xia (Ben) Hu, Tahora H. Nazer, Isaac Jones, Shamanth Kumar, Fred Morstatter, Sai Thejasvee Moturu, Ashwin Rajadesingan, Suhas Ranganath, Jiliang Tang, Lei Tang, Xufei Wang, Reza Zafarani, and Zheng Zhao. Without their incessant help and insightful discussions, this work would have not been possible.

This material is based upon work financially supported by, or in part by, the Office of Naval Research under grant numbers N000141110527, N000141010091, and N000141410095, Army Research Office under grant number 025071, and Air Force Office of Scientific Research under grant number FA9550-09-1-0261, for which I am grateful.

I would like to thank my friends who have provided me with their support. They have been a pillar of strength behind me through the years allowing me to focus and

achieve my goals. I look forward to further support from all of these individuals as I complete my thesis.

My special thanks are reserved for my parents, my brother, and my sisters who have been a continual source of support, strength and motivation and for that I am forever grateful.

Above all I would like to thank my lovely wife Zahra and my wonderful daughter Baran for their love and constant support, for all the late nights, early mornings, and weekends. I owe you everything.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Individuals extensively use online social networking sites to connect to each other, share content, express themselves, and benefit from the information provided by other users. Based on the degree of openness on their profiles, users publicly share some information and preferences including their *attributes* and *interactions*.

Many applications use this information to improve users' online experience or to monitor users' opinion and preferences. Online applications use these types of information to provide customized services to the users in many ways, such as recommending new products, friends, and content, or even providing better search results. Political campaigns, for example, monitor the political views of social media users to predict the outcome of the general elections, and to evaluate the effectiveness of their political strategies. News sites use user preferences to deliver customized news to every user, search engines can deliver personalized search results, or online advertisers can serve targeted ads. They are used by movie recommender systems to recommend movies of interest to their users. Social networking websites use this information to find people with similar interests and recommend them to the user. Having access to users' information is a key to success in both service and the product-oriented industries.

Despite the importance of this information for the service providers, many social media users prefer not to reveal their information to the public. Due to the importance of this information for the service providers, they use different approaches to predict user profile including attributes and connections. In this research, we study approaches which help us to better understand social media users from two perspec-

tives. First, predicting user attributes and second predicting future or missing links. Based on our proposed framework, shown in Figure 1.1, a combination of attributes and connections would help to better understand social media users. Based on the availability of data for each of these categories, we study two problems including large data and limited data. When we look at social media data from a global perspective, it is big data. There are millions of users who generate content and interact with each other. Every algorithm in this context should be able to handle this huge amount of data. Simultaneously, from a local point of view, many social media users do not generate enough data or hide their data due to privacy issues. As a result, algorithms have to deal with limited data problem. In this thesis, we propose solutions for both problems (large scale data and limited data) for attribute prediction and link prediction. First problem is to deal with large amount of data available in social networks. We propose *scalable learning of attributes* to deal with this problem for attribute prediction. We also propose *Egocentric link prediction algorithm* to handle big data for link prediction. The second problem is happening when we do not have access to enough data to perform prediction tasks. Despite the fact that social media data is large, in many cases we do not have access to enough data for specific social media users. We propose to use *network signals* to address lack of data in the attribute prediction problem. Our *event-based link prediction* is a solution for the limited data problem in link prediction.

1.1 Attribute Prediction in Social Networks

Despite the importance of social media information, many social media users do not reveal their personal information, attributes, and preferences such as geographic

	User Understanding	
	Attributes	**Links**
Large Scale Data	Scalable Learning	Egocentric Link Prediction
Limited Data	Network Signals	Event-based Link Prediction

Figure 1.1: In this thesis we propose to use attributes and links to study social media users. For each of the attributes and links, we address *large scale data* and *limited data* problems.

information, age, gender, political view, and interests [82, 39]. These users usually are challenged to manage privacy concerns and balance trade-offs between disclosing and withholding their personal information. As a result, attributes and preferences of only a small fraction of the social media users, is accessible. Kumar et al. [39], analyzing more than 100 million tweets, show that only less than 1% of the tweets are coming with geo-location information. Two common solutions to this problem are: (i) explicitly asking users to provide the information, and (ii) inferring missing attributes and preferences by using other sources of information [37]. Asking a random sample of people is a common approach that is usually used to collect a large population's preferences and information. Surveys are a common attempt to collect people's opinion in large scale. Survey and polling methodology, extensively developed through the 20th century, gives numerous tools and techniques to accomplish representative public opinion measurement such as their political views [37]. An alternative to surveys is to extract the missing attributes using other sources of information. These attempts have a root in experimental psychology which suggests a person may be understood by what happens around him. Predicting individual's interests and preferences based on various cues from the individual and his environment has a long history in social science [26]. Sam Gosling in [25] reveals methods that his team uses

to gather people's preferences and interest only by examining their work and living places.

With the popularity of social media and huge amount of publicly available user-generated data, we are able to investigate users' preferences by studying their online activities in social media [55, 2]. This information can be used in lieu of the explicit information that we expect the users to reveal about themselves. There are considerable amount of work which show the possibility and effectiveness of using publicly available user-generated content in social networking sites to infer users' missing attributes and preferences [51, 42, 48]. Specifically, in a community of users involved in political discussions, and with sufficient user-generated content, researchers predict users' political alignment with more than 85% accuracy [18, 50].

However, the majority of regular social media users are reluctant to talk about controversial topics such as politics or share their political views publicly [17]. Consequently, their profiles and the content generated by them do not reveal sufficient clues about their political views [38]. Therefore, usually political views of only a small fraction of the social media users are given explicitly or can be inferred from their profiles and their user-generated content. For example, in a sample of more than 5.8 million fan pages we have collected from Facebook, only less than 1.0% of them revealed their political views. In this situation, the content generated by the social media users can be used to infer political views of only a small fraction of the users, and the majority of the users can not be covered. For many applications such as opinion mining it is important to infer users' attributes and preferences. A general approach to address this problem is to utilize network information to predict users' missing attributes and preferences [47]. *Network-based* approaches, leverage users' friendship or interaction information to predict their attributes.

In the presence of content information, classical classification algorithms such as

support vector machines (SVMs) and logistic regression are commonly used to find patterns in a data set characterized by a collection of independent instances of a single relation. These patterns are used to predict preferences for unlabeled users. However, when we use network information, we have to deal with new challenges. In network-based approaches, the predictor uses one's connections in the network to infer his preferences. Based on the homophily effect, users are more likely connect to those who share common interests or preferences than to the random users [49]. Consequently, the data points in social networks are not independent and identically distributed. Naively applying classical statistical inferences algorithms such as SVM, which assume that instances are independent, can lead to inappropriate results [45, 30]. Dealing with this problem is one of the major challenges of preference predicting from linked data. Collective inference and relational learning are two common approaches to address the network-based inference problem. However, scalability of the algorithms to deal with millions of data points is the common challenge of the proposed algorithms in this area.

In Chapter 3, we address the scalability problems on predicting social media users' preferences, using a relational learning approach. We develop a network-based scheme to predict social media users' preferences taking into consideration the nature of network information (i.e., non-i.i.d. characteristics of network information). To this end, we design our solution based on the social influence theory, which indicates that a user's preferences are influenced by the influential users in his social circle [81]. This local pattern suggests that an influential user and those influenced by him, should share similar preferences. With this intuition, we use the influential users and their immediate neighbors or neighbors a few hops away to construct local social dimensions. Users in the same local social dimension are likely to share similar preferences. Then we use these local social dimensions as features to train classifiers and predict

users' preferences. Instead of using the entire network information, which is inefficient for real-world social networks, we propose an egocentric local social dimension, which leads to an efficient and a scalable solution. Further, the proposed solution, captures the network's global pattern and balances the trade-off between accuracy and efficiency while is highly scalable.

The second problem relates to silent users, whose information is not available or there is insufficient information about them. These users are not active or do not publicly share their online activities. For example, a report from Harvard Business Review[1] revealed that most Twitter users are inactive, with 10% of all users accounting for 90% of the overall number of tweets. According to this report, among the social media users about 1% of them are content generators, 10% are active, and the rest 89% are silent users. Specially, when it comes to controversial topics such as politics or religions, they become more conservative to either talk about or share their views publicly. Consequently, their profiles and the content generated by them do not reveal sufficient clues about their preferences. Therefore, usually personal preferences of only a small fraction of the social media users are given explicitly or can be extracted from their profiles and their user-generated content. More precisely, both content-based and network-based prediction algorithms require having access to user-generated content, and users' online activities, respectively. Hence, these approaches are not effective in predicting users' preferences whose information is not available or is not sufficient. Most of the prediction algorithms use user-generated content or users' network information to predict users' preferences. However, due to huge portion of the inactive users, most of the approaches are not able to predict these users' preferences [62].

In this research, we use the term "silent user" to refer the users without online

[1]http://mashable.com/2009/06/02/twitter-users-dont-tweet/

activities. From applications' point of view, there is no difference between a silent or an active user. All users including silent and active users are equally important for advertisers, politicians, and businesses. For example regardless of being active or not, users can vote and their political orientations are equally important for political campaigns. In the absence of user-generated content or online activities in which we refer to them as "user-generated signals", other sources of information in the network are available, and might be used to describe the silent users and their preferences. In this research, we refer to the signals generated by other social media users as *"network signals"*. Though this information might be used before, but to the best of our knowledge its effect never officially has been investigated. This information is signals generated by other users in the network. In this work, we investigate the effectiveness of *network signals* (especially those who are having interaction with the silent user) to better understand silent users and their preferences.

1.2 Link Prediction

With the advent and spread of social networking sites, computational analysis of social network structures becomes a common focus of many branches of network science, and huge efforts have been made to understand and model the evolution of social networks [44]. In a social network, nodes represent people or other entities embedded in a social context, and edges represent interaction and collaboration between the entities [43]. One of the most fundamental problems relevant to network analysis, is link prediction [44, 43, 69], which aims at estimating the likelihood of the existence of a link between two nodes based on observed links and the attributes of nodes [22]. Link prediction problem is the process of predicting the most likely

links to appear in the network in near future. The most common approach to predict future links is based on computing a measure of proximity or "similarity" between the nodes, relative to the network topology [44, 43]. All link prediction methods assign a connection weight $s(u_i, u_j)$ to pairs of nodes $< u_i, u_j >$ based on the input from graph $G(U, V)$, where U is the set of nodes and V is the set of edges. The algorithm then generates a ranked list in *decreasing* order of $s(u_i, u_j)$. These scores are considered as a measure of proximity or "similarity" between the nodes. After measuring the proximity between the nodes in the network, algorithms recommend the nodes with highest scores to connect each other.

Liben and Kleinberg in their seminal paper [43] show that "social proximity" is one of the most important factors that leads people connect each other. Social proximity can be measured using different metrics such as number of common friends, length of the path that connects two nodes, or similarity of their attributes or interests. Therefore, the node's position in the network becomes an important source for link prediction algorithms to predict future connections and interactions. For example, if two people have many common friends, they are more likely to connect in near future [8, 53] than those with small overlap among their neighbors. Most of the popular link prediction algorithms use this simple idea to find and rank future connections.

Despite the popularity of this approach, it has some limitations; it is also not consistent with our real-world experiences of finding and connecting to new friends or acquaintances. One limitation of proximity-based link prediction algorithms is that they treat all the connections in the network homogeneously without any differentiating among the connections. In real-world interactions, people connect to each other for different reasons. Due to the homophily effect, we are more likely interested in connecting other people that share similar interests or have common affiliations. For

example, some of our connections are our family members, some of them are our colleagues, and some of them are our classmates, and so on. Therefore, one's neighbors can be clustered into number of groups which are called *social dimensions* [67] or *social circles* [40]. In our daily life, when we connect with other people, we do not look for the most similar person to connect; we simply look for someone with common attributes, affiliations, or interests. For example, we connect to our colleagues because of our common affiliation; we connect to our classmates because of attending the same class; and we connect to our family members because we are part of the same family. In all of these examples, we might have very few common interests. In particular if we measure the structural proximity, in most cases we will end-up with comparatively low structural proximity between these pairs. However, most of the common link prediction algorithms do not consider this heterogeneity and try to find and recommend people based on the overall structural proximity such as number of mutual friends.

Although the link prediction problem has been extensively studied, the existing work did not consider the multidisciplinary aspect of the social media users to predict future links. In this study, we propose an egocentric approach for the link prediction problem. The proposed method uses the multidimensionality aspect of the nodes' connections. Due to the multidimensionality of social media users, they form "social circles" and their neighbors usually can be cluster into few social circles. We show that every new link, either is an extension to the existing social circles, or adds a new dimension to the existing egocentric network. In this work, we use an egocentric approach to cluster the local network for every user. We then use these egocentric local clusters to help predict the further expansions of the network. We address these two problems in Chapter 4. First, we propose *egocentric link prediction* algorithm, and use a local approach to address the large data problem for link prediction. Second,

we address limited data paradox by proposing *even-based link prediction* algorithm. In both of the algorithms, we use egocentric local clustering to cluster every node's network into social circles. Then, we use these social circles to find and recommend the best matches.

1.3 Dissertation Structure

In Chapters 2 and 3, we give a detailed description of the aforementioned solutions, and analyze their usefulness and drawbacks. Each of the Chapters 2 and 3 are self-contained, that is, they can be read in any order. In particular, Chapter 2 presents the attribute prediction problem in social networks, our solutions, experiments and the results. In this chapter we discuss three problems including, scalable learning of users' preferences in social networks, the preference prediction with limited data, and preference prediction in directed networks. Chapter 3 presents our solution for the link prediction problem. In this chapter, we discuss an egocentric approach for link prediction problem. In Chapter 4, we review the related work and the way we contribute in the area. Finally, we conclude the research and discuss open research problems for attribute prediction and link prediction in Chapter 5.

Chapter 2

EGOCENTRIC ATTRIBUTE PREDICTION IN SOCIAL NETWORKS

Inferring users' preferences such as political orientation is an important task for many online and offline systems. Political campaigns need to know people's political view to better plan their future actions. Ad placement algorithms use users' preferences to display the most relevant ads to the user. Recommender systems use this information to recommend products that users might like the most. Traditionally, such information has been collected explicitly by querying users, conducting surveys, or field studies. The popularity of social media has empowered people to generate and publish tremendous amounts of data. Experimental psychology suggests that a person may be understood by what happens around him and his behavior. Following this suggestion, researchers turn to the use of social media data as an alternative source of information for inferring users' preferences. In addition to the content information, which is predominantly used, the connections between social media users also provide a rich source of information for prediction. Recently, many researchers employ link information along with content to improve the accuracy of predictions.

There are two common approaches on using network information for prediction. Researchers either use nodes in a small neighborhood (e.g. k-nearest neighbor) or use the entire network. The first approach is efficient, but needs huge amounts of labeled data, which is not usually available. In the second approach, matrix factorization algorithms such as SVD are commonly used to extract latent attributes, and then classification algorithms are used to infer users' preferences. This approach usually leads to higher prediction accuracy, but is not efficient for large networks.

In this work, we study two problems with network-based prediction of users' preferences. The first problem is efficiency of network-based prediction algorithms. As connections in social media are multi-dimensional, it is common to use discriminative learning algorithms such as logistic regression to extract social dimensions and then use the social dimensions as features to predict users' preferences. Instead of using the entire network information, which is inefficient for real-world social networks, we use local approaches to construct the social dimensions. We use these social dimensions in lieu of latent social dimensions, and by using classical classification algorithms, we efficiently infer users' preferences. The second problem relates to silent users, whose information is not available or there is insufficient information about them. To address the silent users' problem, our approach is to investigate if we can gather additional information from users who are connected or have interaction with silent users, e.g., they like, follow, or tag silent users.

2.1 Scalable Learning of Social Media Users' Attributes

Users' personal information such as their political views is important for many applications such as targeted advertisements or real-time monitoring of political opinions. Huge amounts of data generated by social media users present opportunities and challenge to study these preferences in a large scale. In this research, we aim to infer social media users' political views when only network information is available. In particular, given personal preferences about some of the social media users, how can we infer the preferences of unobserved individuals in the same network? There are many existing solutions that address the problem of classification with networked data problem. However, networks in social media normally involve millions and even

hundreds of millions of nodes, which makes the scalability an important problem in inferring personal preferences in social media. Due to the size of real-world social networks, using the entire network information is inefficient and not practical in many cases. To address the scalability issue, we use social influence theory to construct new features based on a combination of local and global structures of the network. Then we use these features to train classifiers and predict users' preferences. By extracting local social dimensions, we present an efficient and scalable solution. Further, by capturing the network's global pattern, the proposed solution, balances the performance requirement between accuracy and efficiency.

2.1.1 Attribute Prediction Using Network Data

Network data is commonly used to model the relations between the entities of a system, such as relationship between social entities and paths between geographical locations. In such models, entities are represented by nodes whose labels give their attributes, and edges are relations between these entities. The task of inferring users' attributes is to recover the missing attributes of nodes based on the available information from other sources. In network-based approaches of predicting users' attributes, the predictor uses one's connections in the network to infer the missing attributes. The underlying assumption for these algorithms is the social correlation theories such as homophily and influence, which is observed in many social networks, including directed social networks [7]. Based on the homophily effect, users are more likely connected to those with common interests or preferences than to the random users [49]. As a result, structural information of the network can be leveraged to infer properties about users that tend to associate with one another. Due to the homophily and the influence effects, the data points in social networks are not independent and

identically distributed. Therefore, traditional classification algorithms such as SVM may not be directly applied in predicting users' labels or preferences. This is because those algorithms work based on i.i.d. assumptions of the input data. In this situation, the classification of a node may have an influence on the class membership of related nodes, and vice-versa [66]. To overcome this problem, different techniques are proposed. Among them, *collective classification* is a technique that is widely used. The idea of the collective classification is to simultaneously infer the class membership of the nodes in the network [45, 67, 20]. In addition to the collective classification, researchers propose other classification methods built upon the ideas of social science theories such as homophily and influence.

Based on the homophily, similar nodes connect to each other, and based on the influence, connected nodes become similar to their neighbors. Based on perspectives to exploit the social network, the vast majority of existing algorithms can be roughly divided into two groups - *local algorithms* and *global algorithms*. Local algorithms only use the ego-centric networks of users, i.e., users' immediate neighbors or a local view of the social network. The basic assumption behind these methods is that *nearby nodes are likely to have the same label or attribute*. Global algorithms, however, utilize the entire network (or a global view of the social network). Both of local and global algorithms are based on the same assumption that connected users in social networks are likely to share similar characteristics or similar interests, hence, social networks are homogeneous with regards to many personal or behavioral characteristics [49, 79].

Local algorithms usually are easy to implement, fast, and with enough labeled data produce accurate predictions [46]. As a results, in many studies, they are used as baseline solutions [47]. We describe and use weighted-vote relational neighbor (wvRN) algorithm [46] as a representative of the local algorithms and use it as a baseline solution. Global algorithms split the network into clusters of users. Then by

using the information from user's association to the clusters, the solutions infer user's preferences. The clustering assumption is similar to the idea of using dimensionality reduction algorithms such as singular value decomposition (SVD) or matrix factorization, since the central idea of these algorithms is to construct low-dimensional feature set preserving both local and global structure of the network [76]. In the class of global algorithms, we choose a framework based on *social dimensions* [67] for which its superiority over representative relational learning solutions is empirically verified.

Notations Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users where n is the number of users. We use $\mathbf{X} \in \mathbb{R}^{n \times n}$ to denote the social network among these n users where $\mathbf{X}_{ij} = 1$ if u_i has a direct link to u_j, and zero otherwise. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ be the set of class labels where m is the number of labels. $\mathbf{Y} \in \mathbb{R}^{n \times m}$ is the label indicator matrix. $\mathbf{Y}_{ij} = 1$ if u_i is in the j-th class c_j, and zero otherwise. Assume that there are $N \leq n$ labeled users in the network, which indicates that there are only N non-zero entities in $\mathbf{Y}$ and the remaining $n - N$ rows of $\mathbf{Y}$ are zero.

Weighted-vote relational neighbor (wvRN)

Local prediction algorithms, infer user u_i's attributes via using attribute values observed from his "local" edges directly. A user's local edges are the edges which directly connect to him. However, in some cases we also might consider neighbors of up to 2 hops away from the node, as local connections. In this section, we use weighted-vote relational neighbor (wvRN) algorithm [47] to predict users' preferences. This algorithm performs relational classification via a weighted average of the class membership scores of the node's neighbors. The classifier works by making two strong, yet reasonable, assumptions: a) in the given network, some nodes' class labels are known, b) the network exhibits homophily effect. Both of these assumptions hold

for the problem that we try to solve. The algorithm estimates the class membership probability of a node u_i belonging to class $c_j \in \mathcal{C}$.

$$P(c_j|u_i) = \frac{1}{|\mathcal{N}(u_i)|} \sum_{u_k \in \mathcal{N}(u_i)} w_{ik} \mathbf{Y}_{kj}, \tag{2.1}$$

where w_{ik} is the weight of the link between nodes u_i and u_j, and $\mathcal{N}(u_i)$ is the immediate neighbors of u_i, which is formally defined as

$$\mathcal{N}(u_i) = \{u_k | u_k \in \mathcal{U} \wedge \mathbf{X}_{ik} = 1\} \tag{2.2}$$

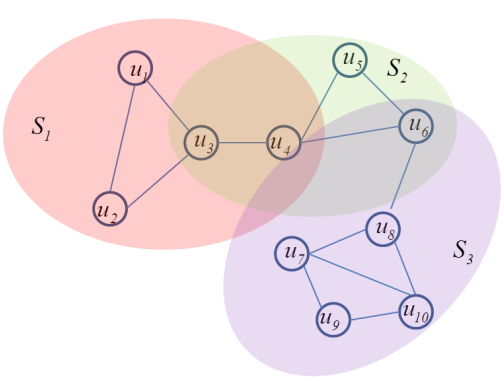
The label of the user u_i is predicted as

$$c_j = \arg\max_{c_j \in \mathcal{C}} P(c_j|u_i) \tag{2.3}$$

Local prediction algorithms are easy to implement. However, as they only use local information to predict users' preference, they are expected to achieve lower accuracy than the global algorithms.

Social Dimensions

Global prediction algorithms, infer a user's attributes using entire network information. Therefore, it is expected that these algorithms infer user's attributes more accurately than the local approaches. Algorithms based on social dimensions are the state-of-the-art approaches to infer users' preferences by utilizing the network information [67]. Social dimension based techniques are usually composed of two steps - (1) extracting social dimensions and representing nodes with social dimensions, and (2) training a classifier on the user presentation by social dimensions. Social dimensions are extracted based on global network information to capture the potential affiliations of users. Then these social dimensions can be treated as features of users for the subsequent classifier learning process. Users in the same social dimension are



	S_1	S_2	S_3	....
u_1	1	0	0	.....
u_2	1	0	0	.....
u_3	1	1	0	.....
u_4	1	1	0	.....
u_5	0	1	0	.....
u_6	0	1	1	.....
u_7	0	0	1	.....
u_8	0	0	1	.....
u_9	0	0	1	.....
u_{10}	0	0	1	.....

(a) Social Dimensions (b) Representation with Social Dimensions

Figure 2.1: Social dimension based methods

likely to interact with each other more frequently. Hence to infer social dimensions, we need to find out a group of people who interact with each other more frequently than randomly chosen pairs of users, which boils down to a classical community detection problem. A typical example of a social dimension based technique is illustrated in Figure 2.1. A community detection algorithm is employed to extract social dimensions such as $\{S_1, S_2, S_3\}$ in Figure 2.1(a), and then users will be presented by social dimensions as shown in Figure 2.1(b). Finally a classifier will be trained based on the new representation.

We use the following optimization problem to implement social dimension algorithm,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{W}} \quad & \|\mathbf{X} - \mathbf{U}\mathbf{V}\mathbf{U}^\top\|_F^2 + \lambda\|\mathbf{H}(\mathbf{U}\mathbf{W}^\top - \mathbf{Y})\|_F^2 \\ & + \alpha(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \beta\|\mathbf{W}\|_F^2 \end{aligned} \tag{2.4}$$

where $\mathbf{U} \in \mathcal{R}^{n\times d}$ captures the latent social dimension structure and d is the number of social dimensions. $\mathbf{W} \in \mathbb{R}^{m\times d}$ is a linear classifier, which is trained on the new representation of users $\mathbf{U}$ based on social dimensions. $\mathbf{H}$ is a diagonal matrix where

$\mathbf{H}_{ii} = 1$ if u_i is labeled, and zero otherwise. The terms of $\alpha(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \beta\|\mathbf{W}\|_F^2$ are added to avoid overfitting.

Since it is difficult to provide a direct closed-form solution for the optimization problem in Equation 2.1.1, we use a gradient decent approach to find local optimal solutions for the variables $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$. Following formulation is the optimization problem:

$$\mathcal{L} = tr[(X^T - UV^TU^T)(X - UVU^T) + \lambda(WU^TH - Y^T)(HUW^T - Y) + \alpha U^TU + \alpha U^TU + \beta W^TW], \tag{2.5}$$

$$\begin{aligned}\mathcal{L} = tr[&(X^TX - 2UV^TU^TX + UV^TU^TVU^T) + \\ &\lambda(WU^THHHUW^T - 2WU^THY + Y^TY) \\ &+2U^TU + \alpha U^TU + \beta W^TW,\end{aligned} \tag{2.6}$$

By taking the derivative of $\mathcal{L}$ with respect to $\mathbf{U}$, we obtain:

$$\frac{\partial \mathcal{L}}{\partial U} = -2XUV^T - 2X^TUV + 2UV^TU^TUV + 2UVU^TUV^T + 2\lambda HHUW^TW - 2\lambda HYW + 2\alpha U \tag{2.7}$$

By taking the derivative of $\mathcal{L}$ with respect to $\mathbf{V}$, we obtain:

$$\frac{\partial \mathcal{L}}{\partial V} = -2U^TXU + 2U^TUU^TUV + 2UVU^TUV^T + 2\lambda HHUW^TW - 2\lambda HYW + 2\alpha U \tag{2.8}$$

By taking the derivative of $\mathcal{L}$ with respect to $\mathbf{W}$, we obtain:

$$\frac{\partial \mathcal{L}}{\partial W} = 2\lambda W U^T H H U - 2 X Y^T H U + 2\alpha H \tag{2.9}$$

We can update U, V, and W as follows:

$$\begin{aligned} U^{t+1} &= U^t - \lambda_u \frac{\partial \mathcal{L}}{\partial U} \\ V^{t+1} &= V^t - \lambda_v \frac{\partial \mathcal{L}}{\partial V} \\ W^{t+1} &= W^t - \lambda_u \frac{\partial \mathcal{L}}{\partial W} \end{aligned} \tag{2.10}$$

After learning the classifier $\mathbf{W}$ and the new representation $\mathbf{U}$ of users based on social dimensions, the label $\mathbf{y}_i$ of an unlabeled user u_i is predicted as

$$\mathbf{y}_i = \mathbf{u}_i \mathbf{W}^\top \tag{2.11}$$

where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$ and is the new representation of u_i based on social dimensions.

2.1.2 The Proposed Scalable Algorithm (LSocDim)

Social influence theory suggests that a user's preference is likely to be influenced by influential users in his social networks. Therefore, wvRN directly uses the preferences of a user's neighbors to infer his preference and avoid accessing the global network information in social dimension methods. Therefore, they are computationally efficient. However, they do not take into account the global structure patterns of network information, and need huge amount of labeled data to generate comparable results with social dimension methods. Social dimension based methods can access the whole

network and extract global pattern (i.e, social dimensions) to represent users. Social dimensions can capture user preference dependence in the social network. Therefore with the social dimension representation, traditional powerful classifiers such as SVM can be trained. They can achieve high accuracy with a small fraction of labeled data. However, finding global structure patterns are very expensive and time consuming.

Local Social Dimensions (LSocDim)

Social influence theory indicates that a user's preference is influenced by influential users in his social networks [68]. This local pattern suggests that an influential user and users influenced by this user should share similar preferences. With this intuition, we define an influential user and his immediate neighbors a few hops away as a local social dimension. Similar to global social dimensions, users in the same local social dimensions are more likely to share similar preferences.

To extract K local social dimensions, we first find K influential users such that the number of users influenced by these K influential users is maximum, and then form a local social dimension with each influential user and users in his immediate neighbors or neighbors a few hop away. Therefore, extracting local social dimensions boils down to finding K influential users. We now formally define the problem of identifying K influential users as follow: *given a social network* $G = (U, E)$ *and a positive integer* $K \leq |\mathbf{U}|$*, identify a set of users* $\mathbf{U}'$ *such that a subset* $\mathbf{U}' \subseteq \mathbf{U}$*,* $|\mathbf{U}'| \leq K$*, and the number of users influenced by* $\mathbf{U}'$ *is maximum.*

The problem of finding K-influential users is tantamount to the maximization of a non-negative, non-decreasing, sub-modular function with a cardinality constraint. A greedy method gives a $(1-1/e)-$approximation for the maximization problem [34, 52]. The proposed algorithm to extract K local social dimensions is shown in Algorithm 1. It first, starts with an empty output set $\mathbf{U}'$, and adds one element from users set $\mathbf{U}$ to

the output set that provides the largest marginal increase in the coverage; it repeats the previous steps until all the users are processed or the maximum cardinality bound K is reached. Then K local social dimensions are formed based on K influential users and their neighbors.

Algorithm 1 Local social dimension extraction

Input: The network information $\mathbf{X}$, K

Output: K Local Social Dimensions

1: Initialize $\mathbf{U}' \leftarrow \phi$
2: **while** $(|\mathbf{U}'| \leq K)$ and $(|\mathbf{U}'| \neq n)$ **do**
3: Find $u_i \in \mathbf{U}$ such that difference between the amount of influenced users from $\mathbf{U}$ and $\mathbf{U}'$ is maximum.
4: Update $\mathbf{U}' \leftarrow \mathbf{U}' \cup u_i$
5: Update $\mathbf{U} \leftarrow \mathbf{U} - u_i$
6: **end while**
7: **for** Each Influential Users u_i in $\mathbf{U}'$ **do**
8: Form a local social dimensions with u_i and immediate or a few hop neighbors
9: **end for**

An example of extracting local social dimensions from the network in Figure 2.1 is shown in Figure 2.2 where $\{u_3, u_6, u_7\}$ are three influential users and $\{LS_1, LS_2, LS_3\}$ are three local social dimensions formed by $\{u_3, u_6, u_7\}$.

Attribute Prediction with Local Social Dimensions

Local social dimensions can capture user preference dependency as social dimensions. Then users can be represented by local social dimensions and traditional powerful classifiers can be trained based on the new representation to facilitate the user pref-

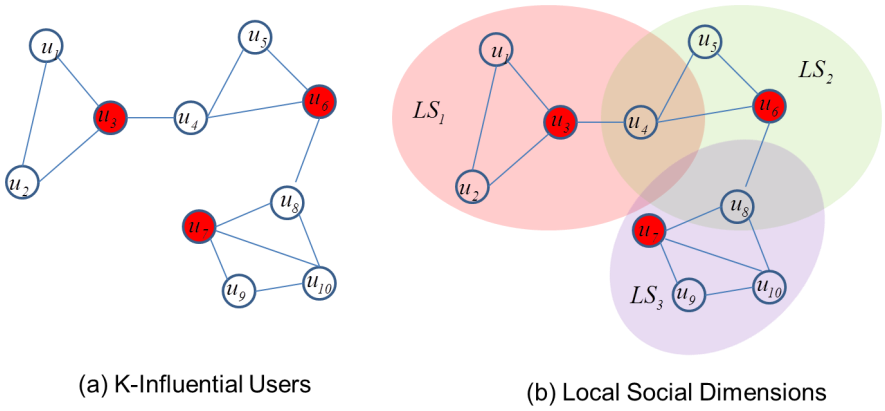


Figure 2.2: Extracting local social dimension

erence prediction problem, which leads to a novel framework LSocDim as shown in Algorithm 2. Next, we briefly review the algorithm. In line 1, we use Algorithm 1 to extract K local social dimensions. In line 2, similar to processes in [67], we treat local social dimensions as new features for users and represent users by local social dimensions. In line 3, we train a classifier based on the new representation.

Algorithm 2 The proposed local social dimension based method

Input: The network information $\mathbf{X}$, K

Output: A Classifier

1: Extract K local social dimensions by Algorithm 1

2: Represent users by local social dimensions

3: Train a classifier based on the new representation of users based on local social dimensions

Complexity Analysis

Due to the size of social networks, time complexity of the algorithm is an important parameter. In this section, we analyze the time complexity of the proposed LSocDim

algorithm and compare it with two baseline algorithms, wvRN and SocDim. In the next section, we present the running time of the three algorithms. *Weighted Vote Relational Neighbors (wvRN) algorithm* needs only to look at the nodes' neighbors; therefore its execution time depends only on the size of the node's neighborhood. A theoretical analysis of time complexity shows that complexity of the local algorithm is $O(n)$, which n is the number of users in the network. *Social Dimensions* algorithm [67], uses a SVD approach to extract social dimensions from the network. The best algorithms for SVD computation of an $n \times n$ matrix take time that is proportional to $O(n^3)$ [1]. Then, the algorithm utilizes the social dimensions as features to train the classifier and predicts the missing labels. In this algorithm, extracting the social dimensions is the bottleneck and has the highest effect on the complexity of the algorithm. Overall, the algorithm has the time complexity proportional to $O(Kn^3)$, where K is number of social dimensions. In *Local social dimensions* algorithm extracting K local social dimensions needs $O(KnN^2)$ operations [34, 52]. We treat local social dimensions as features. Since, the number of social dimensions is much smaller than the number of users, $K \ll n$, the new representation is much denser than the original representation. Most of the popular classifiers can be trained with less than $O(KnN^2)$ operations. Hence, the overall time complexity of Algorithm 2 is $O(KnN^2)$

2.1.3 Evaluation

In this section, we run experiments to evaluate the proposed algorithm. First, we evaluate the efficiency of the proposed algorithm against the two representative algorithms: 1) weighted vote relational neighbor (wvRN) [46] and 2) social dimension

[1] http://rakaposhi.eas.asu.edu/s01-cse494-mailarchive/msg00028.html

(SocDim) [67]. We also study how the performance varies with the size of the labeled data. To evaluate the scalability of the algorithm, we use datasets with different sizes and run the experiments. For every experiment, we randomly sample a portion of nodes as labeled and report the average performance of 10 runs. The section starts with a quick introduction about the baseline algorithms. Then the dataset is described, and finally the results are presented.

Dataset

We use a directed dataset from Facebook. The nodes are Facebook fan-pages and the links are formed by *like* relation between the pages. Each page can *like* or *be liked* by other pages. In our settings, if page u_i *like* page u_j, we consider u_j as u_i's *followee* and u_i as u_j's *follower*. In the network structure, each page is a node and liking another page creates a link from the follower to the followee. There is no limitation on the number of users who can like a Facebook page. The number of likes is a public property of the page, and is further considered as a measure of popularity in our experiments. Table 2.1 shows the statistics about the dataset.

Table 2.1: Our Facebook fan-pages dataset

Number of nodes	5,856,000
Number of links	19,646,000
Size of labeled data	25,129 (0.43%)

Data collection process The dataset is collected by crawling Facebook pages through the site's public web interface. We start with a small set of seeds from the United States politicians, whose pages are publicly available on Facebook. We follow a *breadth first search (BFS)* algorithm to expand the nodes to the pages that are liked

by the current page. Thus, after we crawl all of the seed pages, we continue with the pages being liked by the seeds, and this process is continued until all of the pages are collected. For every page, we collect the following publicly available attributes: *title of the page*, *number of likes*, *political view*, *political party*, *category*, *gender*, and *list of liked pages*.

Table 2.2: Political views of self-reported Facebook fan-pages in our dataset

Political view	Distribution
Conservative	23%
Moderate	19%
Liberal	18%
Very Liberal	7%
Libertarian	7%
Very Conservative	4%
Apathetic	2%
Other	21%

In our experiments, labels are pages' political views. Table 2.2 shows political views and their distribution in the dataset. Among more than 5.8 million Facebook pages in our dataset, only 25,129 of them revealed their political views or parties which is about 0.43% of all pages. We use *page category* information to filter users affiliated with political issues. *Page category* is a public attribute of Facebook fan-pages, which is usually chosen from a dropdown list of predefined values. In our dataset, *Community* with 16.8%, *Musician/Band* with 7.4%, *Non-Profit Organizations* with 4.1%, and *Public figure* with 3.8% popularity are the most popular categories. Our target pages are chosen from categories related including *Public Figures*, *Politicians*, *Political Organizations*, and *Political Parties*.

Performance Analysis

Accuracy of the prediction or precision is the most important factor in evaluating the performance of prediction algorithms. We define precision as the fraction of the preferences correctly predicted as follows,

$$\frac{\sum_{c_i \in C} tp_{c_i}}{\sum_{c_i \in C} (tp_{c_i} + fn_{c_i})} \tag{2.12}$$

where C is the set of all class labels, tp_{c_i} is *true positive*, number of accurately classified nodes, and fn_{c_i} is *false negative*, number of misclassified nodes for the given cluster.

Figure 2.3 shows the accuracy of all three algorithms (wvRN, SocDim, and LSocDim). The second bar in the graph represents our proposed algorithm, Local Social Dimensions (LSocDim). It can be seen from the figure that the results of LSocDim algorithm are promising and are comparable with SocDim (the state-of-the-art prediction algorithm for networked data). Among the three algorithms, wvRN uses the minimum network information (only information from node's neighbors), and SocDim uses the maximum network information. LSocDim, however, uses local information, but also considers the global patterns of the network. In all of the experiments the local algorithm, wvRN, takes the third place in prediction accuracy among the three algorithms. Though the highest prediction accuracy of 45.6% belongs to SocDim method, according to Table 2.3 LSocDim outperforms SocDim for all of the experiments with less than 10% labeled data. These results confirm the effectiveness of LSocDim when only a small fraction of the data is labeled. When a small fraction of the data is labeled, nodes tend to associate with other like-minded nodes in local communities. In this situation, techniques that use the entire network information to extract the clusters are not as effective as those focus more on local information but consider the global signals as well. LSocDim is an example of a local approach with global patterns. Recall our goal is to design a scalable algorithm where its accuracy is

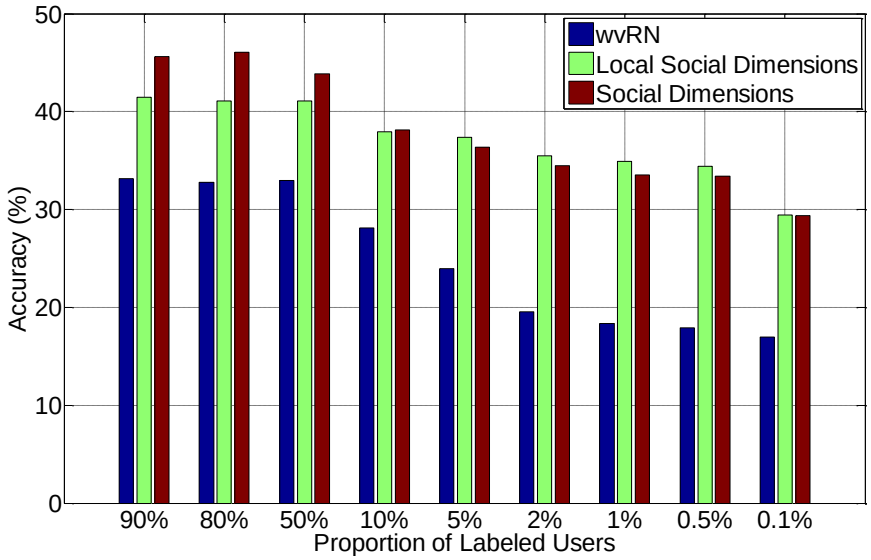


Figure 2.3: Accuracy of Local Social Dimensions comparing with wvRN [46] and Social Dimensions. The x-axis shows the fraction of the nodes that are labeled.

comparable with other state-of-the-art algorithms.

Table 2.3: Accuracy of the three algorithms, with respect to the size of labeled data.

Labels	90%	80%	50%	10%	5%	2%	1%	0.5%	0.1%	RND
LSocDim	41.5	41.1	41.1	37.9	**37.4**	**35.5**	**34.9**	**34.4**	**29.5**	18.6%
wvRN	33.1	31.8	35.0	28.1	23.9	19.6	18.4	17.9	17.0	18.6%
SocDim	**45.6**	**46.1**	**43.9**	**38.1**	36.4	34.5	33.6	33.4	29.4	18.6%

Efficiency Analysis

It is always desired to design algorithms to be both efficient and accurate. However, usually there is a trade-off between performance and efficiency of prediction algorithms. Generally, simple learning algorithms such as lazy learners are efficient, but their performance is not as good as state-of-the-art algorithms. More complicated algorithms are needed to achieve higher prediction accuracy. However, these algo-

rithms are computationally more complex, compared to simple learning algorithms. Efficiency and scalability of algorithms become more important when we use them in large networks or real-time applications. As a result, usually there is a trade-off between the efficiency and performance of the prediction algorithms.

In this experiment, we use the running time of each algorithm to measure the efficiency. Table 2.4 shows the running time of the three algorithms. Among the three methods, wvRN is the fastest one with only 0.9 second for every experiment. On average, LSocDim algorithm needs 39.9 seconds to predict labels in a network with 26,240 nodes and 457,597 edges. SocDim algorithm however, needs much longer time to predict the labels with an average of 1107 seconds for each iteration. In both SocDim and LSocDim algorithms, running time positively correlates with the size of labeled data. The reason is that when size of the labeled data decreases, the training set size decreases. Therefore, the algorithms converge faster comparing to the case where we use larger training set.

Figure 2.4 provides a comprehensive comparison between *LSocDim* and *SocDim* (global) algorithms. The experiments show that LSocDim is 23 to 34 times faster than SocDim, which is a huge improvement on the efficiency of the algorithm. However, there is a huge gap between LSocDim and wvRN algorithm. From efficiency point of view, LSocDim can be considered as a bridge between wvRN and SocDim algorithms.

Scalability Analysis

In this section, we study the scalability of the proposed algorithm, i.e., how the computational time of the algorithm (when running on a Core i7-4770 CPU and 16GB memory desktop) varies with the number of nodes in the network. To evaluate the scalability of the algorithm, we construct four smaller samples of the original network. Table 2.5 shows statistics about the networks.

Table 2.4: Running time of the three algorithms, *wvRN*, *LSocDim*, and *SocDim* (all numbers are in Seconds). The first row shows the fraction of the labeled data. In *SocDim* and *LSocDim*, running time has a direct correlation with the fraction of revealed labels. In wvRN algorithm, running time is a constant value.

Labels	90%	80%	50%	10%	5%	2%	1%	0.5%	0.1%
Local	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
LSocDim	47.2	46.2	42.2	38	38	38	37.8	38	37.6
Global	1237	1255	1260	1308	1209	1278	878	888	876

Labeled Users	Accuracy (%)			Running time (Seconds)		
	SocDim	LSocDim	LSocDim vs SocDim	SocDim	LSocDim	LSocDim vs SocDim
90.0%	45.6%	41.5%	-9.1%	1237	47.2	3.8%
80.0%	46.1%	41.1%	-10.8%	1255	46.2	3.7%
50.0%	43.9%	41.1%	-6.4%	1260	42.2	3.3%
10.0%	38.1%	37.9%	-0.5%	1308	38.0	2.9%
5.0%	36.4%	37.4%	2.8%	1209	38.0	3.1%
2.0%	34.5%	35.5%	3.0%	1278	38.0	3.0%
1.0%	33.6%	34.9%	4.1%	878	37.8	4.3%
0.5%	33.4%	34.4%	2.9%	888	38.0	4.3%
0.1%	29.5%	29.5%	0.1%	876	37.6	4.3%

Figure 2.4: From performance point of view, LSocDim is comparable with SocDim, and from efficiency point of view LSocDim is comparable with wvRN. Therefore, LSocDim can be considered as a bridge between wvRN and SocDim algorithms.

We run the experiments and measure the running time of each algorithm for all four networks. Table 2.6 shows the running time for the three algorithms with respect to different network sizes.

Figure 2.5 shows how the running time increases when we use networks with larger sizes. In the figure, x-axis shows the increase of the network size, and y-axis shows the running time increase. Among the three algorithms, LSocDim has the minimum slope of increasing the running time, and SocDim has the maximum slope. The theoretical analysis and the empirical study show the scalability of the proposed local

Table 2.5: Smaller networks to test the scalability of the algorithms

	N_1	N_2	N_3	N_4
Nodes	529	900	1715	2498
Edges	2879	7507	10939	24201
Number of Clusters	137	313	492	1301

Table 2.6: Scalability analysis. The table shows the number of seconds each algorithm needs to predict labels for the given network.

	N_1	N_2	N_3	N_4
wvRN	0.011	0.025	0.042	0.079
SocDim	0.960	2.140	7.291	13.861
LSocDim	0.026	0.037	0.056	0.121

social dimension algorithm. Incorporating the results from performance analysis and efficiency analysis, it is evident that the proposed algorithm is capable of handling the prediction task.

Sensitivity Analysis

In both supervised and semi-supervised learning algorithms, size of the labeled data is an important parameter that affects the prediction accuracy [77], which is also observed in the experiments of the previous sections. Table 2.3 shows the accuracy of the prediction algorithms with respect to the size of labeled data. The first column on the left hand side of the figure is the accuracy, when 90% of the users are labeled, and the algorithm needs to predict the remaining 10% non-labeled users. As the table suggests, for all of the three algorithms, there is a direct correlation between the size of labeled data and the accuracy of prediction. Further, the table and Figure 2.3 shows two trends. First, the prediction accuracy of both SocDim and LSocDim smoothly

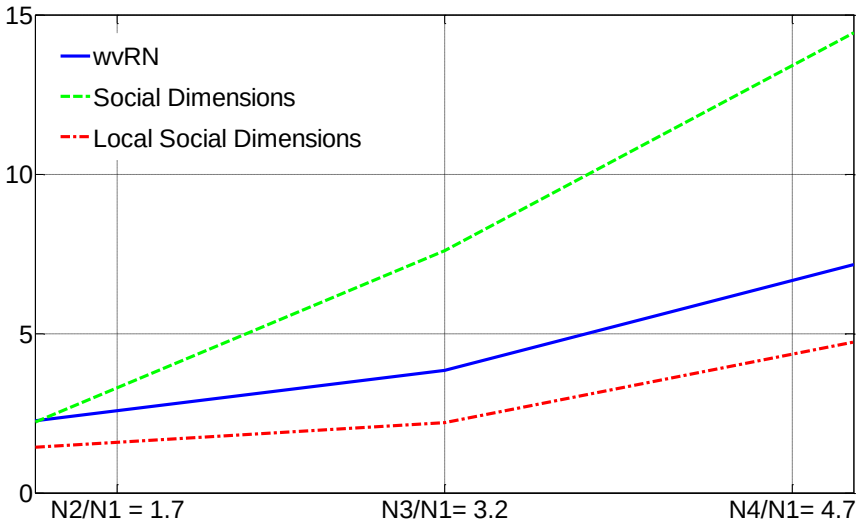


Figure 2.5: Scalability of the algorithms; x-axis shows the growth of the network size, and y-axis shows the running time increase.

decreases when the size of labeled data decreases from 90% down to only 5%. It shows that these methods are more prone to the changes of the size of the labeled data. Even when only 5% of the data is labeled, the prediction accuracy of these algorithms is relatively high. Second, the results show that wvRN is highly sensitive to the size of the labeled data. In particular, when the fraction of the labeled data goes under 10%, the accuracy of wvRN decreases sharply, and the results become similar to the random prediction. In local approaches, algorithms usually use the nodes' immediate neighbors to make prediction. However, due to the size of nodes' neighbors and small fraction of the labeled nodes, most likely algorithms are not able to find a labeled node in the neighborhood. The need for a huge amount of labeled data can be considered as one of the main drawbacks of local algorithms, as the labeled data is not always available. Unlike the local algorithms, SocDim and LSocDim use entire network information. This means that they use the entire network structure to extract the clusters, train a model, and then make prediction.

Consequently, even small fraction of labeled nodes can be effectively used to achieve high prediction accuracy.

Table 2.3 shows that even when only 1% of the data points are labeled, the prediction accuracy of the global approaches are higher than the accuracy of local algorithm with 90% labeled data. Surprisingly, when the size of the labeled data decreases from 90% to 1%, the accuracy of SocDim and LSocDim only drops to 30%. These results show the stability of the global approaches against the size of the labeled data. Among the three algorithms, LSocDim is more efficient and reliable when smaller sizes of labeled data are available. The experiments show that when less than 10% of the users are labeled, LSocDim always outperform SocDim in prediction accuracy. Labeling the data points is an expensive and time consuming process, and in many cases it may be impossible to label enough data points. Therefore, it is important that the algorithm has the capability of predicting with reasonable accuracy even if a small set of labeled data is available. In this regard, the experimental results prove that LSocDim is preferred over the other two algorithms.

In the literature, many of the proposed algorithms are evaluated with more than 10% available labeled data points. This is far from many real world cases where the available labeled data can be as low as (or less than) 1%. One line of our future work is to evaluate the performance of the existing algorithms with respect to the size of labeled data.

2.1.4 Discussion

In this chapter we studied the network-based approach of inferring users' personal preferences. We categorized the network-based algorithms into *local* and *global* algorithms. Local algorithms use only users' immediate neighbors to predict their

preferences, while the global approaches use the entire network information to predict user's preferences. Our experimental results show that local algorithms are fast and scalable; however they need large amount of labeled data to achieve reasonable prediction accuracy. Further their prediction accuracy is always less than the accuracy of global algorithms. *Global* algorithms, in contrast, are computationally expensive, but perform well even in cases where only a very small fraction of the data is labeled. We proposed a new algorithm called *LSocDim* based on social influence theory to bridge the efficiency of local algorithms and the accuracy of global algorithms. The experiments show the efficiency and the effectiveness of the proposed algorithm. In particular, we show that LSocDim achieves prediction accuracy near to that of the state-of-the-art global algorithm, SocDim, while improving the running time by up to 40 times. The proposed algorithm compromise the accuracy of global approaches based on social dimensions in order to run faster. Another advantage of this algorithm over those based on latent attributes such as singular value decomposition (SVD) or random projection [41, 58, 59] is its simplicity to understand the features and implement the algorithm. The proposed algorithm in some aspects, including performance and scalability, is similar to random projection method, and we propose to have a comprehensive comparison between these two algorithms and assess their efficiency and effectiveness on text and network analysis as future extension to this work.

By performing a sensitivity analysis with respect to the size of labeled data, we show that SocDim performs better than the baseline local algorithm, wvRN, when the available labeled data is limited. We also show that the proposed algorithm, LSocDim, performs better than SocDim, when less than 10% of the data is labeled. This is an important result, considering that labeling the data points is expensive and time consuming process, and in many cases it is even impossible to label enough data points. We also evaluate the scalability of the algorithms. The theoretical

and experimental results show that the proposed algorithm is computationally less expensive than SocDim algorithm, the baseline global algorithm. The scalability analysis also shows a promising result. As the networks in social media are normally involving millions and even hundreds of millions of nodes, it is important for prediction algorithms to be fast and scalable.

2.2 Attribute Prediction with Limited Data

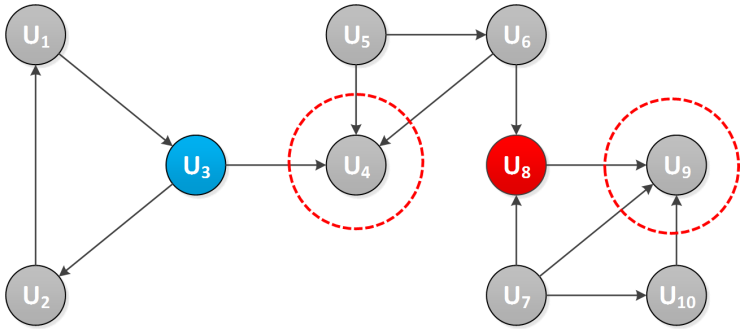
Successful online services perform the best by having access to users' attributes and preferences. With having access to this information, the quality of service for recommendation, advertisement, and marketing systems improves significantly. It has been shown that content generated by users and their interactions with other users provide a reliable source of information to predict their preferences. However, this argument is not valid for *"silent users"*, whose information is not available to be used by predictive systems. In this research, we investigate methods to predict missing information and personal attributes of *silent* users. In particular, we study the effectiveness of using network signals, those generated by users other than our target user, to predict the attributes of silent users. Experimental results on Facebook dataset show the effectiveness of the proposed approach on predicting silent users' preferences.

2.2.1 Introduction

Individuals extensively use online social networking sites to connect each other, share content, express themselves, and benefit from the information provided by other users. Based on the degree of openness on their profiles, users publicly share some

of their information and preferences such as demographic information, age, gender, political view, and interests. Online applications use this information to provide customized services to users in many ways, such as recommending new products [56, 6], friends [60], content [61], or even providing better search results [63]. This information also is used by the agencies to monitor users' opinion on different topics. Political campaigns, for example, monitor the political views of social media users to predict the outcome of the general elections, and to evaluate the effectiveness of their political strategies [71, 72]. This information offers an alternative opportunity for first responders and disaster relief organizations to collect information about the disaster, victims, and their needs [3]. Gathering information of social media users is very important for both businesses and organizations. Despite the importance of this information, many social media users do not reveal their personal information and preferences [82]. These users usually are challenged to manage privacy concerns and balance trade-offs between disclosing and withholding their personal information. With the popularity of social media and huge amount of publicly available user-generated and network data, we are able to investigate users' preferences by studying their online activities in social media [55, 2, 5]. There are considerable amount of work showing the possibility and effectiveness of using publicly available information in social networking sites to infer users' missing attributes and preferences [51, 42, 51, 48].

The existing approaches use information provided by the user in the form of either *content information* or *interactions with other users* to predict users' attributes. There are different types of content in social media such as *status update*, *review*, *comment*, and *blog post*. Content-based preference prediction approaches are rooted in social psychology, which suggests a person may be understand by his belongings and behaviors [26]. Predicting individual's interests and preferences based on various cues left from the individual and his environment has a long history in social



	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
U1			1							
U2	1									
U3		1		1						
U4										
U5				1		1				
U6				1				1		
U7								1	1	1
U8									1	
U9										
U10									1	

Figure 2.6: In the figure on left hand side, users u_4 and u_9 are silent users. The figure on the right hand side shows the adjacency matrix of the given network. Based on this matrix, users u_4 and u_9 do not have any activity in the network

science. Sam Gosling in [25] shows how accurately people's personality, preferences and interests can be inferred only by examining their work and living places. The network-based approaches, however, mostly have originated from social correlation theories such as homophily and influence, implying the similarity between connected people [49].These approaches might use either friendship or interaction networks. Examples of interaction networks include *like*, *tag*, and *retweet* networks. Both content- and the network-based approaches are designed based on the users' online behavior including the content generation behavior and interactions with other users, and perform well only if users have generated sufficient amount of online activities.

However, many of the regular social media users are not active or do not publicly

share their online activities. For example, a report from Harvard Business Review[2] revealed that most Twitter users are inactive, with 10% of all users accounting for 90% of the overall number of tweets. Specially, when it comes to controversial topics such as politics or religions, they become more conservative to either talk about or share their views publicly. Consequently, their profiles and the content generated by them do not reveal sufficient clues about their preferences such as their political views. Therefore, usually personal preferences of only a small fraction of the social media users are explicitly given or can be extracted from their profiles and based on their user-generated content. For example, in a sample of more than 5.8 million fan-pages we have collected from Facebook, only 0.43% of the users revealed their political views.

Most of the prediction algorithms use user-generated content or users' network information to predict users' preferences. However, due to huge portion of the inactive users, most of the approaches are not able to predict these users' preferences [62]. In this research, we use the term "silent user" to refer the users without online activities. From applications' point of view, there is no difference between a silent or active user. All users including silent and active users are equally important for advertisers, politicians, and businesses. For example regardless of being active or not, users can vote and their political orientations are equally important for political campaigns.

In the absence of user-generated content or online activities in which we refer to them as "user-generated signals", other sources of information in the network are available, and might be used to describe the silent users and their preferences. In this research, we refer to the signals generated by other social media users as *"network signals"*. Though this information might be used before, but to the best of our knowledge its effect has never been investigated. This information is signals gener-

[2]http://mashable.com/2009/06/02/twitter-users-dont-tweet/

ated by other users in the network. Social psychology studies show the effectiveness of user-generated activities (content and interactions) on understanding the user and his preferences. In this work, we investigate the effectiveness of *network signals* (especially those who are related to the silent user, e.g., had interaction with the silent user) to understand the silent user and his preferences.

In this research, we investigate the issues of predicting silent users' preferences in social media as illustrated in Figure 2.6. By utilizing the network signals to infer silent users' preferences, we aim to answer two research questions: 1) How effective are network signals on predicting silent users' preferences? 2) For regular users, which sources of information are more effective, network signals or user-generated signals?

Problem Statement

Given the network $G(U, E)$, where U is the set of users and E is the set of connections between the users. Matrix $\mathbf{X}$ denotes the adjacency matrix for the given graph $G(U, E)$. In this research, we use users' *like behavior*, to build the adjacency matrix. Each directed link $e_{ij} \in E$ forms when $u_i \in V$ like $u_j \in U$.

Now we formally define social media silent user preference prediction problem as: *Given the network $G(U, E)$ with the adjacency matrix of $\mathbf{X}$, the task is to predict silent user $U_i \in U_S$'s preferences where $X_i = \mathbf{0}$.*

Silent Users

There are different approaches on categorizing social media users based on their level of activities. Among them, dividing the users into *active users* and *inactive users* is more common. For example according to Page 44 of Facebook's prospectus[3] a user is considered active if he or she *"logged in and visited Facebook through our website or*

[3]http://www.foxbusiness.com/technology/2012/02/01/full-text-facebooks-ipo-prospectus/

a mobile device". In this definition, there is no need for a user to have an action on the site to be considered as active user. There are other sources that categorize social media users based on their level of activities in the network. For example, Aimia[4] has identified 6 social media personas in the United States. Among these 6 social media personas, three groups with a distribution of only 28% are actively interact with other users and share information. Another 16% of the users, infrequently post on social media. The remaining 56% of the users barely or almost never posts anything on social media.

Regardless of which definition we use to define active and inactive users, a huge amount of social media users hardly post or interact with other users on social media. In this research, we refer to these users as *"silent users"*. *Silent users are social media users whose information is not available or there is not sufficient information about them.* These users also might be referred as "passive users"[5] or "inactive users"[6] in the literature.

2.2.2 Network Signals

To address the "silent users" problem, we investigate methods that lead to better understanding of silent users and predicting their personal preferences. Our approach is to use signals from other social media users who are connected to, or have interactions with silent users. Examples of such interactions are *liking*, *tagging*, or *following* silent users. In this work, we investigate whether a user can be understood by what

[4]http://www.digitalstrategyconsulting.com/intelligence /2012/06/six_types_of_social_media_user.php

[5]http://www.huffingtonpost.com/belle-beth-cooper/10-surprising-social-medi_b_4325088.html

[6]http://www.jeffbullas.com/2013/07/04/5-insights-into-the-latest-social-media-facts-figures-and-statistics/

is happening around him, but not initiated by himself. Particularly, we investigate the effectiveness of signals generated by other users connected to the current user. The main idea behind this work is that, though silent users are inactive and did not interact with other users, other users' interactions with silent users might help to understand silent users. From this perspective, we categorize the other social media users into two categories and form our hypothesis based on the sources we use to study silent users. 1) A silent user can be understood through those directly have communication with him, such as his immediate neighbors. This is a local approach on studying silent users. According to Figure 2.6, u_4 is a silent user without any network activity. However, other users (u_3, u_5, and u_6) in the network had interaction with u_3. In this approach we use these users to study u_4. 2) A silent user can be understood by assigning him to some clusters based on passive interactions with them. These approaches use entire network information to cluster the network and predict preferences.

Notations Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of n users. We use $\mathbf{X} \in \mathbb{R}^{n \times n}$ to denote the directed social network among these n users where $\mathbf{X}_{ij} = 1$ if u_i has a directed link to u_j, and zero otherwise. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ be the set of class labels where m is the number of labels. $\mathbf{Y} \in \mathbb{R}^{n \times m}$ is the label indicator matrix where $\mathbf{Y}_{ij} = 1$ if u_i is in the j-th class c_j, and zero otherwise. Assume that there are $N \leq n$ users who are labeled, which indicates that there are only N non-zero entities in $\mathbf{Y}$ and $n - N$ rows in $\mathbf{Y}$ are zeros.

Extracting the Preferences Locally

In this approach, we employ those social media users who directly interact with silent users. Examples of the interactions with silent users are *following silent users*, *tagging*

silent users, or *liking silent users*. In all of these activities the silent user does not perform any action, and we use the actions of other users to study the silent user. In this approach, we assume that users, who interacted with silent users, might find some sort of similarity with the silent user. Then we use this as an interpretation for existing the homophily effect between the silent user and the node interacted with the silent user.

For local prediction, we use weighted-vote relational neighbor (wvRN) algorithm [46] to predict silent users' preferences. This algorithm performs relational classification via a weighted average of the class membership scores of the node's neighbors. The classifier works by making two strong, yet reasonable, assumptions: a) in the given network, some nodes' class labels are known, and b) the network exhibits the homophily effect. The algorithm estimates the class membership probability of a node u_i belonging to class $c_j \in \mathcal{C}$.

$$P(c_j|u_i) = \frac{1}{|\mathcal{N}(u_i)|} \sum_{u_k \in \mathcal{N}(u_i)} w_{ik} \mathbf{Y}_{kj}, \tag{2.13}$$

where w_{ik} is the weight of the link between nodes u_i and u_k, and $\mathcal{N}(u_i)$ is the immediate neighbors of u_i, which is formally defined as

$$\mathcal{N}(u_i) = \{u_k | u_k \in \mathcal{U} \wedge \mathbf{X}_{ik} = 1\} \tag{2.14}$$

The label of an unlabeled user u_i is predicted as

$$c_j = \arg\max_{c_j \in \mathcal{C}} P(c_j|u_i) \tag{2.15}$$

Next, we describe the global approach on predicting silent users' preferences.

Extracting the Preferences Globally

In this section, we propose a method based on social dimensions algorithm [67] that captures the global structure of the network, and clusters the network based on the

connections between the users. Then we use users' membership on these clusters (communities) as features to predict silent users' preferences. This prediction method is based on an observation in social networks in which users tend to form communities with other like-minded users [46]. The algorithm includes two parts: 1) extracting latent social dimensions based on the network structure, and 2) training a classifier on the user presentation by social dimensions. We formulate our social dimension-based algorithm as follows

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{U}\mathbf{V}\mathbf{U}^{\top}\|_F^2+\lambda\|\mathbf{H}(\mathbf{U}\mathbf{W}^{\top}-\mathbf{Y})\|_F^2 \\ & +\alpha(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)+\beta\|\mathbf{W}\|_F^2 \end{aligned} \tag{2.16}$$

where $\mathbf{U}\in\mathcal{R}^{n\times d}$ captures the latent social dimension structure and d is the number of social dimensions. $\mathbf{W}\in\mathbb{R}^{m\times d}$ is a linear classifier, which is trained on the new representation of users $\mathbf{U}$ based on social dimensions. $\mathbf{H}$ is a diagonal matrix, where $\mathbf{H}_{ii}=1$ if u_i is labeled, and zero otherwise. The terms of $\alpha(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)+\beta\|\mathbf{W}\|_F^2$ are added to avoid overfitting. Since it is difficult to provide a direct closed-form solution for the optimization problem in Equation (4), we use a gradient decent approach to find local optimal solutions for the variables $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$. Below is the optimization problem

$$\begin{aligned}\mathcal{L} = tr[&(X^T-UV^TU^T)(X-UVU^T) \\ &+\lambda(WU^TH-Y^T)(HUW^T-Y) \\ &+\alpha U^TU+\alpha U^TU+\beta W^TW]. \end{aligned} \tag{2.17}$$

Using this approach, we solve the problem and learn the classifier $\mathbf{W}$ and the new representation $\mathbf{U}$ of the users based on social dimensions. The label $\mathbf{y}_i$ of an unlabeled user u_i is predicted as

$$\mathbf{y}_i=\mathbf{u}_i\mathbf{W}^{\top} \tag{2.18}$$

where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$ and is the new representation of u_i based on social dimensions.

2.2.3 Evaluation

In this section, first we evaluate the effectiveness of using network signals on predicting silent users' preferences. Then, we compare the effectiveness of network signals with user-generated signals. For both of the experiments, we use two algorithms 1) weighted vote relational neighbor (wvRN) [46] and 2) social dimension algorithm (SocDim) [67]. The former algorithm captures the local structure and the later algorithm captures the global structure of the network.

Dataset

In this study, we use *Facebook fan-pages* to construct the directed social network. On Facebook, users can create regular user accounts or fan-pages. For regular user accounts, the connections between users are undirected, but in fan-pages the connections are directed. To connect to a page, users have to like the target page. This is similar to *following* behavior on Twitter. Both regular Facebook users and fan-pages can *like* fan-pages, and each page can *like* other pages or *be liked* by other pages.

In the network structure, each page is a node and liking another page creates a link from the source to the target node. There is no limitation on the number of users that can like a Facebook fan-page. The number of likes is a public property of the page and in our experiments is used to measure of popularity of the pages. Table 2.7 represents the statistics about our Facebook dataset. This dataset is collected by crawling Facebook through the site's public web interface.

Table 2.7: Facebook fan-pages dataset

Total number of pages	5,856,000
Number of personal pages	764 K
Number of links	19,646,000
Revealed political orientation	25,129 (0.43%)

Experimental Setup

We construct a directed network based on liking behavior on Facebook. For each experiment, political views of $x\%$ of the users are exposed, and the task of the classifier is to predict the remaining $(1-x)\%$, where x is varied as $\{0.1, 0.5, 1, 2, 5, 10, 50, 80, 90\}$ In this research. For every experiment, we randomly sample a portion of nodes as labeled and report the average accuracy of 10 runs.

Evaluating the Effectiveness of Network Signals

Prediction using local information In this section, we report the experiments that we conduct to evaluate the effectiveness of local network signals on predicting silent users' preferences. In the directed network constructed by like behavior on Facebook, local signals come from those who liked silent users. The experiment results are shown in Figure 2.7.(a). In this figure, x-axis shows the percentage of labeled data and y-axis shows the accuracy of prediction. The baseline is the random prediction of users' political orientation. The results show that local network signals are highly effective on predicting users' preferences. For example, with using 90% of the data labeled, the accuracy is 38.04% while the random predictor achieves only 16.79% accuracy. However, the prediction results are not consistent when the size of labeled data decreases. For example, with 10% and 2% labeled data, the average accuracy is 25.85% and 19.99% respectively. Though the approach is highly sensitive

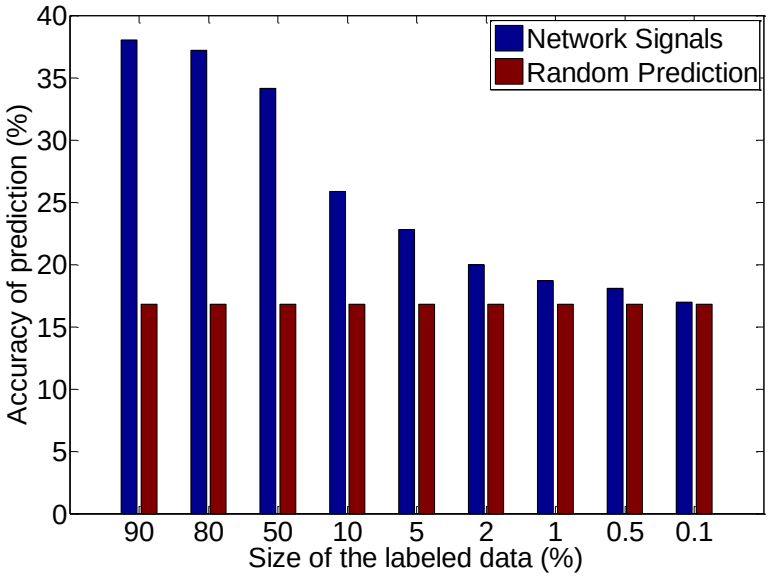
to the size of labeled data, it confirms our original hypothesis on the effectiveness of network signals on predicting silent users' preferences.

Prediction using global information By using social dimension algorithm, we are able to capture the global structure of the network, and use the information to predict silent users' preferences. We conduct similar experiments as we described for the previous section, and report the results in Figure 2.7.(b). In this method, we use entire network information to cluster the network, and use these clusters to predict silent users' political views. The prediction accuracy of these experiments is more promising than using only local information. Another observation is the stability of the method when we use different portions of the data to train the model. In this approach, even when only 1% of the users are labeled, the accuracy of the method is more than 40%, which is higher than the accuracy of neighbor-based approach, with 90% labeled nodes.

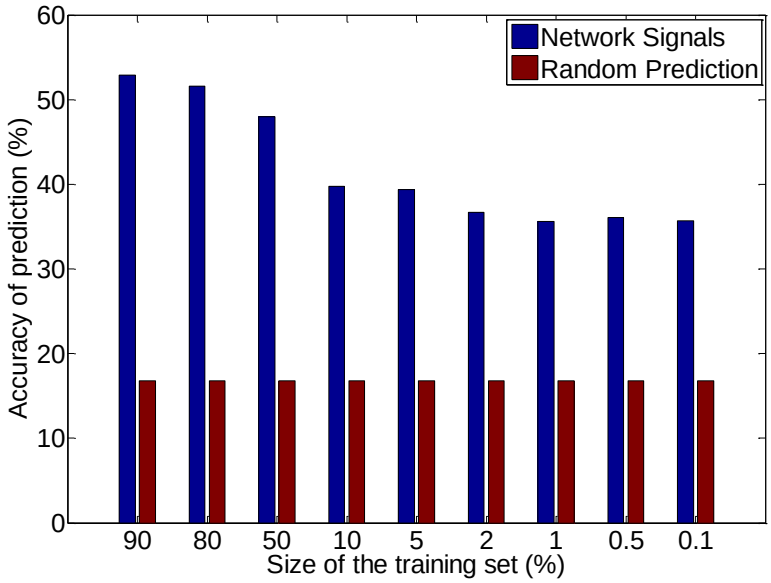
Evaluating the Effectiveness of Network Signals vs. User-generated Signals

In this section, we evaluate the comparative effectiveness of the user-generated signals and network signals on predicting users' preferences. We try to answer the following question: *when we have access to both user-generated signals and network signals, what source(s) of information is more promising to predict users' preferences?* For example, consider the network in Figure 2.8. The task is to predict preferences and compare the results for the given user u_i when we can use user-generated signals and network signals.

Figures 2.9 and 2.10 shows the accuracy of predicting users' political orientation using three sources of information, network signals, user-generated signals, and combination of network and user-generated signals. These experiments show the ef-



(a) Prediction using local information



(b) Prediction using global information

Figure 2.7: Predicting silent users' political orientation using a) local information and b) global information

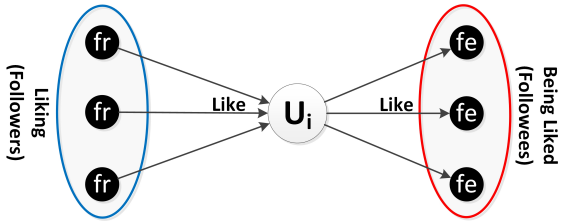


Figure 2.8: On social networking sites, *like* is a behavior with a source and a target user. The behavior of the user who liked, is active behavior and the one's whom being liked is passive behavior.

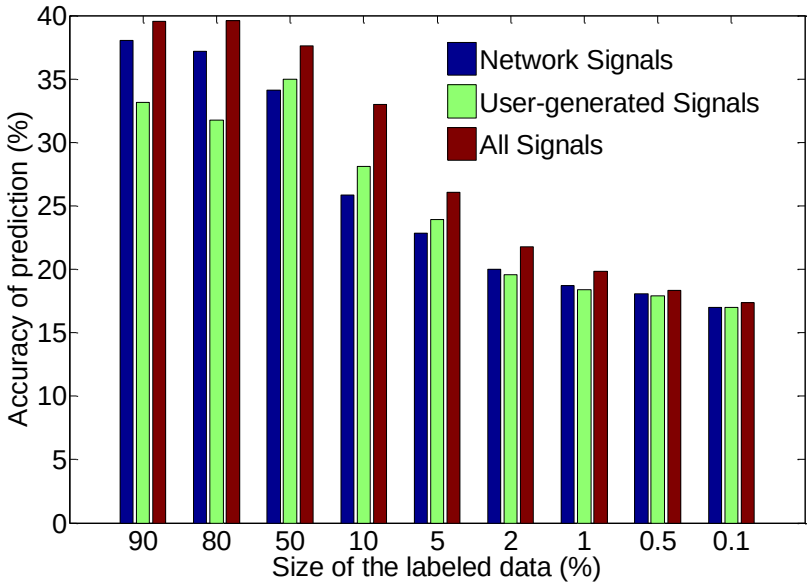


Figure 2.9: Preference prediction using local information, comparing the effectiveness of network signals with user-generated signals.

fectiveness of network signals comparing with the signals generated by the users on predicting their preferences. Following are our observations from this experiment: 1) network signals are usually as effective ad user-generated signals on predicting users' attributes, and 2) a combination of network signals and user-generated signals might increases the prediction accuracy when we use local information, however, it decreases the prediction accuracy when we use entire network information.

Figure 2.9 shows that in most of the experiments, network signals generates higher

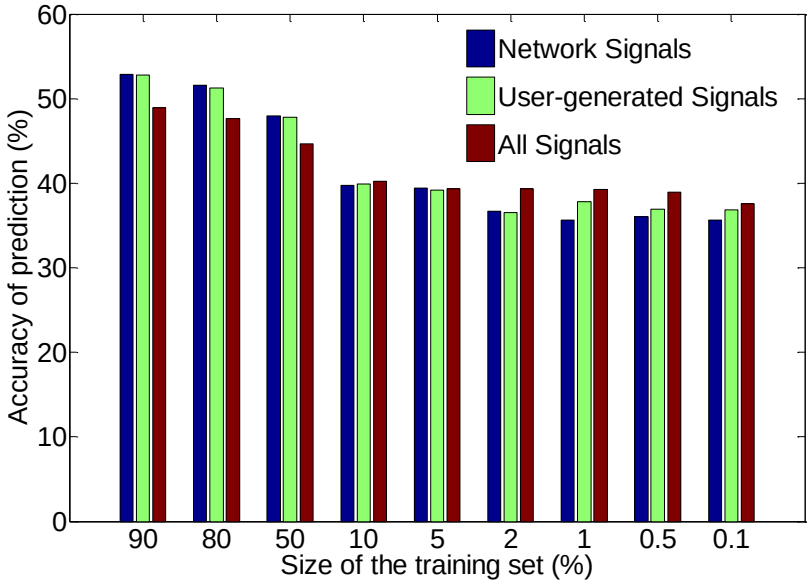


Figure 2.10: Preference prediction using global information, comparing the effectiveness of network signals with user-generated signals.

prediction accuracy than user-generated signals, which is surprising. One explanation for these results is that in our dataset most of the users have more in-links than out-links. Therefore, when we use network signals, the algorithm has access to more information than when we use user-generated signals. In global approach where the predictor uses the entire network information, the results of prediction accuracy from network signals and user-generated signals are similar.

From the previous set of the experiments (Figure 2.7), we observe that using more labeled data leads to higher prediction accuracy. With the same intuition of using more sources of information to receive higher accuracy, we run a new experiment and use the maximum available information including network signals and user-generated signals to predict users' political views. Surprisingly, in global approach (Figure 2.10) using "all signals" does not lead to higher accuracy comparing with using only network signals and only user-generated signals. However, in local approach, using "all signals" always leads to higher accuracy. This might be due to the limited num-

ber of accessible labeled nodes in local approach. Therefore, using more sources of information increases the accuracy of prediction. However, in the global approach, we use the entire network's information, using a combination of network signals and user-generated signals might increase the inconsistency and decrease the accuracy of prediction.

This experiment shows that in cases that we use local information to predict preferences, the network signals lead to higher accuracy than user-generated signals. In global information approach, however, network signals and user-generated signals perform similarly on predicting users' preferences.

Evaluating the Effect of Popularity

To evaluate the effect of source popularity on predicting accuracy, we categorize users based on their popularity. In this experiment we use users' number of likes to measure their popularity. We calculate the similarity of each group of users regarding their popularity level and plot the results in Figure 2.11. As shown in this figure, overall, followers are better predictor for users than their followees, although there are some exceptions. When popularity of the neighbors increases, followers and followees show opposite behavior; when popularity of followers increases, their prediction power increases. Followees, on the other hand, become less similar to the user who followed them when they become more popular. *Preferential attachment* provides an insight to analyze this behavior. Based on preferential attachment, a random user more likely would choose to connect to a popular user rather than a non-popular user [12]. Therefore, following a popular user is more likely due to the preferential attachment property rather than holding similar interests or opinions. Considering the fact that popular users usually have a limited number of carefully chosen followees, we expect to observe higher degree of similarity between a popular user and his followees. These

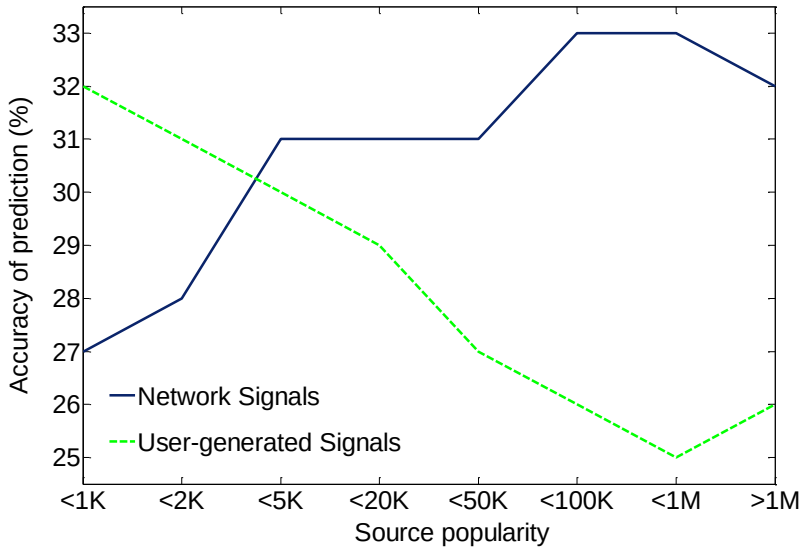


Figure 2.11: The figure shows the effectiveness of source popularity on predicting users' political views. In this figure, the x-axis indicates user popularity and the y-axis represents the accuracy of prediction using wvRN algorithm

users have too many followers and many of them might have other reasons to follow than holding the same preferences. Non-popular users, on the other hand, eagerly follow other users, hoping those users follow them back [15]. These users usually follow a wide range of other users; therefore, there is a lower chance of selecting their followees from those who hold similar preferences. Their small set of followers, on the other hand, should have a good reason to follow a non-popular user. Therefore, there is higher chance that a non-popular user and his followers have similar interests or similar attributes, such as political orientation.

2.2.4 Discussion

We used two approaches on predicting silent users' preferences including a local and a global approach. We first, show that it is possible to use network signals to predict silent users' preferences. In many cases, network signals exhibit higher predic-

tion accuracy than user-generated signals in predicting users' political orientation. By conducting another set of experiments, we show the effectiveness of network signals comparing with user-generated signals. These results suggest that even in the presence of user-generated signals, we can use network signals to achieve higher prediction accuracy.

Privacy Issues Previous work on inferring missing information in social networks shows that it is possible to predict users' interests and personal preferences by exploiting their own online behavior [36]. In this study, we show that it is possible to predict a user's missing attributes by using his neighbors' online activities, such as their following (or liking) behavior. The experiments show that network signals are as good as and in some cases even better sources of information on inferring users' preferences than those activities generated by users. This finding raises privacy issues that the online users must be aware of. According to these findings, if the social media users are mindful about their private information, not only they have to control their own online behavior, but also should care about all of the other users who are connected to or have interaction with.

2.3 Attribute Prediction in Directed Networks

In this chapter we employ collective inference to learn preferences of nodes in a social network. Despite the popularity of this approach in learning preferences in networks, it is usually employed for undirected networks [31], and to the best of our knowledge there is not a comprehensive study on evaluating the effectiveness of this approach in directed networks. Collective inference exploits the characteristics of relational data in which the value of an attribute for connected instances are highly

correlated. This also can be explained by homophily in social science, which is the theory behind the formation of social ties between individuals with similar characteristics or interests. Based on homophily in a social network, it is expected to observe a higher degree of homogeneity among connected than disconnected people. In collaborative inference we use this simple yet effective principal to infer users' missing information and interests based on the information provided by their neighbors. In this section, we study homophily and its effect on collective inference and preference prediction in directed networks. To study this problem in a directed network, we analyze if a user's personal preferences can be inferred from those of users who are connected to him. In our experiments we use a directed network which connections can be further divided into followers and followees. Our goal is to evaluate the effectiveness of each of these two groups of neighbors on predicting one's preferences.

2.3.1 Introduction

Individuals extensively use online social networks to connect to other users, share information, express themselves, and benefit from the information provided by other users. In social networks, users often connect to those who have similar characteristics or similar interests. As a result, social networks are homogeneous with regards to many personal or behavioral characteristics [49]. *Homophily* is the tendency of similar individuals to form connections. The effect of this phenomena is a network in which connected users are more likely to share similar attributes and interests than disconnected users [81]. Homophily is rooted in undirected social networks, in which the two sides of the interaction are equally responsible to create and maintain the relation. Forming a real-world friendship is an example for this type of behavior. There is another type of connection that mostly appears in traditional mass media

as well as online directed social networks. In this type of relation, only one party is responsible for the creation of the connection. Becoming a fan of an author or following a user on Twitter are examples of directed relations. In the example of an author and the group of his fans, the connection between the fans and the author is different from a regular friendship. The author has no control over these connections or does not even know many of his fans. Though the fans find themselves similar to the author, it cannot be concluded that the author also will find himself similar to his fans. In this example, the relation is formed and maintained solely by one of the parties involved in the relation. How can we measure the homophily effect in a directed social network? If the fans find themselves similar to the author, does this imply that the author will also reach the same conclusion?

A similar situation can be observed in many online social networks. In many networks, such as Facebook, the relation is bidirectional, where two connected users have to show their willingness to form the relation. For instance, to form friendships on Facebook, one should initiate a friend request and the other user should accept it. However, in many social networks, the relations are directed. A directed relation, such as *following* on Twitter or *liking* on Facebook, is the result of only one user's action, and in many cases even the second user is not aware of such connection (Figure 2.12).

In this work, we study homophily in directed social networks. To analyze homophily in directed networks, we investigate if a user's personal preferences can be inferred from his neighbors. Our goal is to determine which group (followers or followees) is more effective on inferring users' personal preferences.

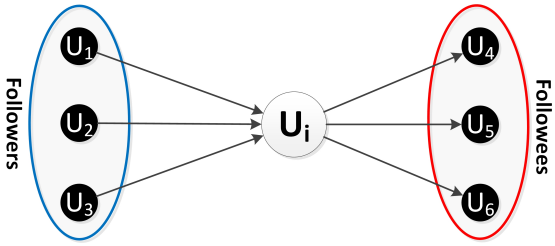


Figure 2.12: In directed networks, the connections can be divided into two groups, and every user only has control over one group of the connections. For example, on Twitter a user selects his friends (or followees), but she does not have any control on selecting his followers. Similarly on Facebook a user chooses what pages or contents to like, but she cannot control or force other users to like his content or page.

2.3.2 Homophily-based Prediction of User's Attributes

In this section, we introduce our homophily-based approach for predicting user's preferences. We evaluate the predictive power of followers and followees for predicting users' profile attributes. We follow a two-step approach: first, we determine the level of homophily between users and their followers and users and their followees. Then we use followers and followees as independent sources to predict users' profile attributes.

Measuring Homophily

To measure homophily, one requires a method to compute homogeneity between users and their followees and followers. We employ a similar measure to the one outlined by Mislove et al. [50] to calculate the homophily among the users. Let a_i denote the value for attribute a for user u_i. We calculate the similarity among the user u_i and his neighbors $u_j \in N(u_i)$ on attribute a as

$$S_a = \frac{\sum_{u_j \in N(u_i)} \sigma(a_i, a_j)}{|N(u_i)|} \tag{2.19}$$

where $N(u_i)$ is the set of u_i's neighbors, and $\sigma(a_i, a_j)$ is the Kronecker delta function

that returns 1 if the value of attribute a is equal for the two users and 0, otherwise.

$$\sigma(a_i, a_j) = \begin{cases} 1 & \text{if } a_i = a_j \\ 0 & \text{otherwise,} \end{cases}$$

$N(u_i)$ can be either u_i's followers or followees. For every user, we run the algorithm twice; first we use followers and then we use followees. In Equation 2.19, the value of S_a represents the fraction of the nodes with similar attribute values for the given attribute a. To measure the statistical significance of S_a, we divide S_a by the expected value E_a when two users are chosen at random. Assume that attribute a can take k attribute values. Let A_i, denote the number of users that take the ith, $1 \leq i \leq k$ possible value for attribute a. Let $U = \sum_{i=1}^{k} A_i$ denote the total number of users. Then E_a can be computed as

$$E_a = \frac{\sum_{i=1}^{k} A_i(A_i - 1)}{|U|(|U| - 1)} \tag{2.20}$$

Let $H_a = \frac{S_i}{E_i}$ denote the degree of homophily between the user and his neighbors. When H_a is 1, there is no correlation between the attribute values. When it is less than 1, there is a negative correlation, and when it is greater than 1, it indicates a positive correlation between the attribute a's value of the user and the neighbors. Higher H_a indicates higher correlation between the attribute values of the user and that of the neighbors.

Predicting the Profile Attribute Values

The algorithm infers the given node's missing information by using the node's neighbors as the source of information. In this study, we use weighted majority vote to infer the user's attributes. To predict the value of attribute a for user u_i, we take the majority vote from u_i's neighbors regarding this attribute and assign the value with the highest number of votes.

2.3.3 Evaluation

As we described earlier, in directed networks connections can be divided into two groups including followers and followees. Every user has control over one group of the connections. For example, on Twitter user selects his friends (or followees), but does not have any control on selecting his followers. On Facebook a user chooses what pages or contents to like, but she can not control or force other users to like his content or page. Therefore, we expect a higher degree of similarity between users and their followees than the users and their followers. We conduct two sets of experiments to evaluate the effect of homophily in directed social networks.

- *Observing the homophily*, to investigate the existence of homophily in directed networks. We measure and compare similarity between the user and his followees, and the user and his followers.

- *Investigating the prediction power of followers and followees*, we predict user's attributes, by using their followers and their followees and compare the results of the two sources.

Dataset We conduct our experiments by using a set of more than 5 million Facebook fan-pages. In this dataset which is crawled from Facebook during July 2013 to January 2014, we have a complete user profile and their network. Figure 2.13 shows the popularity distribution of the users in the dataset.

Homophily in Directed Networks

Our goal in this experiment is to show whether a user is more similar to his followees or his followers and to verify if there is any significant difference between these two. We use the technique described in previous section to measure homophily and to

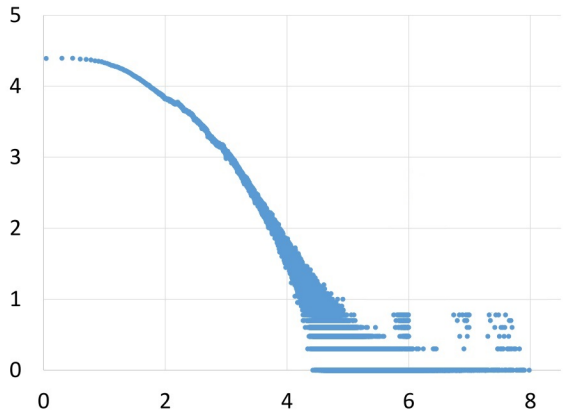


Figure 2.13: User popularity distribution, x-axis is popularity (logarithm of number of page likes) and y-axis is the frequency of pages holding that popularity.

evaluate the results. We use *political orientation* and *page category*, as the attributes for measuring homophily.

The experiments show that in more than 72% of cases, users have similar political orientation with their immediate neighbors, including followees and followers. In our dataset, the probability of holding the same political orientation for randomly chosen pairs of users is 25%. Next, we cluster the neighbors into two groups, including followers and followees. We observe a similarity of 73.5% between users and their followers, which is slightly higher than 74% similarity with their followees. There is a possibility that users' popularity influences our results. To investigate this possibility, we divide the users into two groups based on their popularity. A user is considered popular if she has more than 10,000 likes and non-popular if she has less than 1,000 likes. As we can see in Table 2.8, popular users are more politically aligned with their followees than non-popular users. In contrast, non-popular users are more likely to hold the same political orientation that their followers hold. One explanation for this observation is that popular and non-popular users' liking behaviors are different. Popular users, often have a small number of followees that are chosen very carefully. Therefore, we expect to observe a higher degree of similarity between a popular user

Table 2.8: Political orientation consistency between users, their followees, and their followers with respect to different levels of user popularity

Neighbors	All	$\leq 1K$	$\geq 10K$	≤ 100	$\geq 1M$
Followees	74%	75%	75%	73%	73%
Followers	73.5%	73%	74%	76%	74%
Fe + Fr	72%	72%	73%	73%	72%

and his followees. Popular users have too many followers and these followers might have variety reasons other than holding the same political orientation for following the popular user. Non-popular users, on the other hand, are more eager to attract more followers; therefore they follow other users hoping that these users would follow them back. Therefore, non-popular users are less likely to share similar attributes with their followees. On the other hand, the small set of users who follow non-popular users should have a good reason to follow them. Therefore, there is a good chance that a non-popular user and his followers share similar interests or attributes, such as a political orientation.

Page Category We run the same set of experiments for measuring homophily, but instead of *political orientation* attribute, we use the *page category*. Table 2.9 shows the results. On average, 35% of the connected users belong to the same category. Users in 39% of the cases have a similar category with their followees and in 37% of the cases with share similar category with their followers, which in both cases is higher than using a combination of followers and followees.

The Effect of Popularity on Homophily *Popularity* is an attribute that is correlated with the page's number of likes. As popularity follows a power-law distribution, we compute the logarithm of the number of user's likes, $\log(\text{likes}(u_i))$, to discretize

values of the attribute into 8 categories. Figure 2.13 shows the popularity distribution in our dataset. To evaluate the effect of page popularity on users' following behavior, we measure the relative popularity of each user and popularity of his followers and popularity of his followees. Results indicate that in 49.5% of the cases, users like more popular users. In 12.5% of cases, users like users with the same level of popularity, and in 38% of cases users like less popular users. This result matches with our expectation that users usually follow those who are more popular than themselves. Figure 2.14 shows this behavior with respect to different popularity levels. As we can see in this figure, 24% of extremely popular users like users that are not as popular as themselves. Users with more than 2,000 and less than 10,000 likes are the most balanced group of users with respect to following and being followed by users with the same popularity level.

To evaluate the effect of users' popularity on their following (liking) behavior, we measure the homophily of each group of users with respect to their popularity level. The results show that in general, followees better match with users than their followers, although there are some exceptions. Users with less than 100 likes highly match with those who followed them. When the popularity increases, we observe a higher homophily effect between users and those they like (follow). The maximum homogeneity belongs to users with about 100K likes. Beyond that, the trend changes

Table 2.9: Category Consistency between Users, Their Followees, and Their Followers with respect to Different Levels of User Popularity

Neighbors	All	$\leq 1K$	$\geq 10K$	≤ 100	$\geq 1M$
Followees	39%	44%	35%	30%	30%
Followers	37%	39%	34%	33%	26%
Fe + Fr	35%	40%	33%	31%	26%

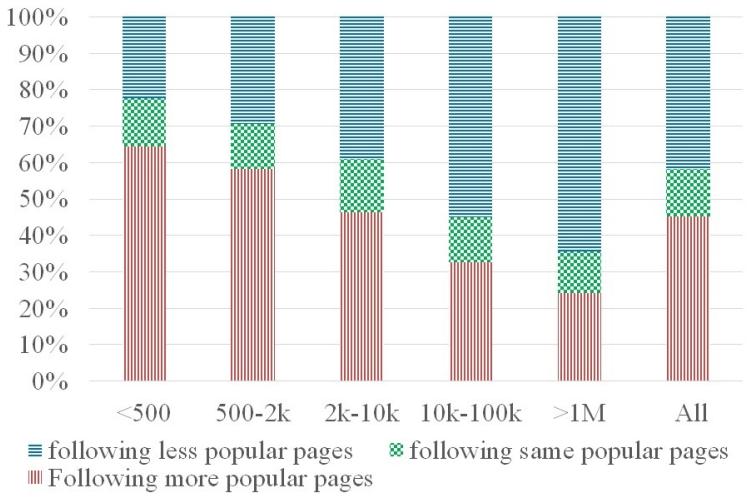


Figure 2.14: More than 62% of Facebook pages like pages that are more popular than or equally popular as the page is.

and the curve touches its minimum level of similarity, which belongs to celebrities. Celebrities, usually have a non-uniform liking behavior. They follow users from different categories and different popularities, which decreases the homogeneity between the user and his followees. The same effect occurs with those users who follow celebrities. A celebrity has followers from a variety of categories and interests, which decreases the similarity between the celebrity user and his followers.

Neighbors Diversity

In this section, we investigate the effect of neighbors' diversity on homophily. We use entropy to measure the diversity among followers and followees as follows,

$$e_i = -\sum_k P(A_k) log P(A_k) \tag{2.21}$$

where A_i is the number of users that take the ith, $1 \leq i \leq k$ possible value for attribute a and e_i is the entropy of user u_i's neighbors with respect to attribute a. Higher entropy indicates the higher diversity among one's neighbors. We calculate the

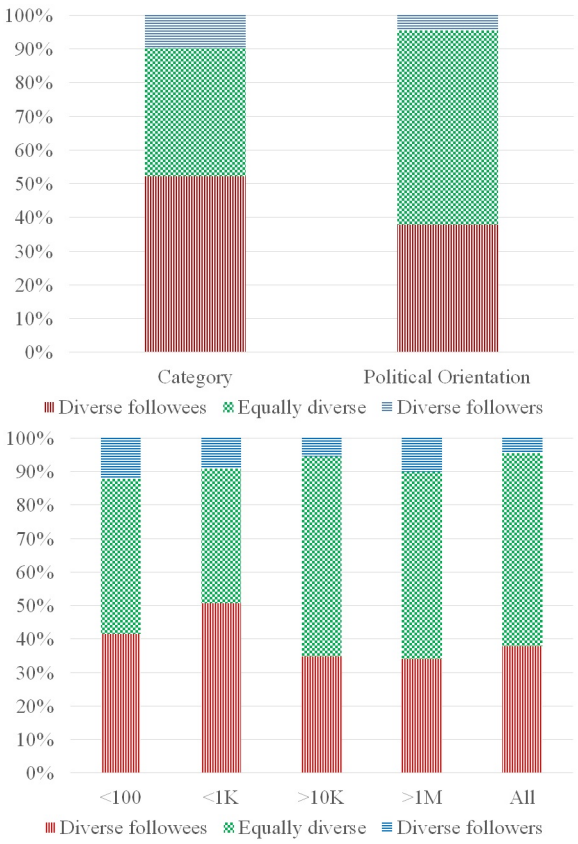


Figure 2.15: Neighbor diversity among followers and followees. For both of the attributes including page category and political orientation, followees are more diverse than followers. The figure on bottom shows the political orientation diversity for pages with different popularity level.

entropy for followers e_{i_r} and followees e_{i_e}. We summarize the results in Figure 2.15 considering the following possible scenario: $e_{i_r} \approx e_{i_e}$, $e_{i_r} > e_{i_e}$, or $e_{i_r} < e_{i_e}$.

Each bar in Figure 2.15 shows three values. The blue bar shows the percentage of users who have more diverse followees than followers; the red bar shows the percentage of users who have as diverse followers as followees, and the green bar the percentage of the users who have more diverse followers than followees. For both of the attributes, *political orientation* and *page category*, followees are more diverse than followers. Looking at this problem from a user popularity point of view, users with less than

1,000 likes follow the most diverse group of users. Diversity among the followers and followees is a measure that can be used to decide which source should be used to infer users' missing information.

Neighbors' Prediction Power

In this section, we investigate the neighbors' prediction power. We use followees, followers, and the combination of followees and followers to predict users' missing information. As previously mentioned, we use weighted majority vote to infer users' missing information. Similar to the previous section, we use followees and followers to predict users' *political orientation* and *category* .

Predicting Political Orientation In these sets of experiments, we used immediate neighbors to predict users' missing information. The results show that if we use all of the users' neighbors, including followees and followers, by using majority vote algorithm, we can achieve 75% accuracy in predicting users' political orientation. If we limit the neighbors to only the followees, the accuracy increases to 77%. By using one's followers to predict his information we are able to achieve 73% accuracy, which is less than followees and a combination of followees and followers. Table 2.10 shows the detailed results with respect to different levels of user popularity. The results show that in all different experiments, followees are better sources to predict users' political orientation. Though followers are not as good as followees, they can correctly predict political orientation in more than 73% of cases. Similar to the results from Section 2.3.3 using a combination of followers and followees does not improve the accuracy compared to just using the followees.

Table 2.10: Predicting political orientation using users' neighbors

Neighbors	All	$\leq 1K$	$\geq 10K$	≤ 100	$\geq 1M$
Followees	77%	78%	76%	78%	78%
Followers	73%	72%	72%	73%	72%
Fe + Fr	74%	74%	74%	74%	74%

Table 2.11: Page category prediction using users' neighbors

Neighbors	All	$\leq 1K$	$\geq 10K$	≤ 100	$\geq 1M$
Followees	45%	47%	40%	29%	31%
Followers	43%	36%	38%	32%	26%
Fe + Fr	43%	36%	39%	33%	28%

Predicting Category Similar to predicting political orientation, we used neighbors to predict users' category. Using all neighbors generates 43% accuracy which is less than followees with 45% accuracy and is similar to followers with 43% accuracy. Prediction results with respect to different levels of users' are reported in Table 2.11.

2.3.4 Discussion

Our goal in this research was to study homophily in directed social networks. We investigate whether one can use the neighbors in directed networks to infer users' preferences. We use a dataset of 5 million Facebook fan-pages and form a directed network to conduct experiments. We divided every user's neighbors into followers and followees, and use them to infer users' personal preferences. The experiments revealed one's followees can be used to predict his preferences with 74% accuracy. With a similar setting, followers predict users' preference with 73.5% accuracy. The results show the effectiveness of both followers and followees on predicting one's preferences. Previous work on inferring missing attributes in social networks show that it is possible

to predict users' personal preferences by using their own online behavior. In this study, we show that not only users' own online behavior, but also users' neighbors' behavior can be used to predict their missing attributes and preferences.

2.4 Summary

In this chapter, we studied preference prediction problem in social networks. In the first problem we proposed a scalable algorithm to predict users' preferences using collective inference. We categorize the learning algorithms into local and global algorithms. Our experimental results show that local algorithms are fast and scalable; however they need large amount of labeled data to achieve reasonable prediction accuracy. Global algorithms, in contrast, are computationally expensive, do not need much labeled data, and generate higher accuracy than local algorithms. We proposed a new algorithm called *LSocDim* to bridge the efficiency of local algorithms and the accuracy of global algorithms. The experiments show the efficiency and the effectiveness of the proposed algorithm. In particular, we show that LSocDim achieves prediction accuracy near to that of the state-of-the-art global algorithm, SocDim, while decreasing the running time by up to 40 times.

The second problem is preference prediction with limited data which is a common problem in social networks. To address this challenge, we propose to use network signals as a source to predict user's preferences. We first, show that it is possible to use network signals to predict silent users' preferences. Interestingly, in many cases network signals exhibit higher prediction accuracy than user-generated signals in predicting users' political orientation.

In the third problem, we studied preference prediction in directed networks. In this

problem, we divided the network into two groups including followers and followees. Followees are those whom the current user decided to follow them, and followers are those who decided to follow the current user. We show that similar to undirected networks, we can use collective inference approach to predict users' preferences in directed networks. The results show that followers are as effective as followees on prediction users' preferences. We further studied these two groups based on their popularity and the correlation between their popularity and their prediction power.

Chapter 3

EGOCENTRIC LINK PREDICTION IN SOCIAL NETWORKS

We consider the problem of *link prediction* in social networks: given the structure of the network at a certain time, we seek to predict links between nodes that will form in the future. A common approach for link prediction is to compute a measure of proximity between nodes based on the network structure, such as the number of common neighbors between nodes, which is then used to rank pairs of nodes in terms of predicted likelihood of the link appearing in the future. We approach the link prediction problem from an *egocentric* perspective, where we seek to predict the most likely links an ego node will form in the future. From analyzing egocentric network data, we discover that connections to egocentric *social circles* and *times of link formations* play an important role in the selection of nodes an ego connects to. We then propose an approach for link prediction that estimates social circles using egocentric clusters and incorporates link formation times in order to improve upon existing proximity measures for link prediction, improving accuracy by 10% on average on a Facebook and a Google+ data set.

3.1 Introduction

With the advent and spread of social networking sites, computational analysis of social network structures has become a common focus of network science, and significant efforts have been made to understand and model the evolution of social networks [44]. In a social network, nodes represent people or other entities embed-

ded in a social context, and edges represent relations between the entities including friendship, interaction, and collaboration [43].

One of the most fundamental problems relevant to network analysis is *link prediction* [44, 43, 69], which aims to estimate the likelihood that a link between two nodes will be formed in the future based on the structure of the network and the attributes of the nodes [22]. The link prediction problem has many interesting applications. In a social network in which links connect users, predicting future links may correspond to predicting future friendships or interactions between the users. Predicted links may also be recommended to users to help them grow their network.

Due to the complexity and dynamic nature of social networks, identifying the mechanisms by which they evolve is a fundamental research question that helps us to better design accurate link prediction algorithms. In a friendship network there are many reasons why two people connect or interact with each other, and many researchers try to answer this question.

The most common approach to predict future links is based on computing a measure of proximity or "similarity" between the nodes, relative to the network topology [44, 43]. All link prediction methods assign a connection weight $s(u_i, u_j)$ to pairs of nodes $< u_i, u_j >$ based on the input from graph $G(U, V)$ where U is the set of nodes and V is the set of edges. The algorithm then generates a ranked list in *decreasing* order of $s(u_i, u_j)$. These scores are considered as a measure of proximity or "similarity" between the nodes. After measuring the proximity between the nodes in the network, algorithms recommend the nodes with highest scores to connect each other.

Social proximity, $s(u_i, u_j)$, can be measured using different metrics such as number of common friends ($s(u_i, u_j) = |\Gamma(u_i) \cap \Gamma(u_j)|$), length of the path that connects two nodes ($s(u_i, u_j) = 1/sp(u_i, u_j)$), or similarity of the nodes' attributes and interests

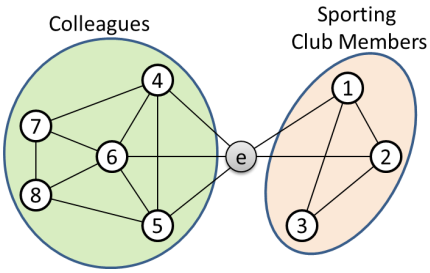


Figure 3.1: User e's neighbors belong to two social circles, colleagues and sporting club members. Nodes with many neighbors in common with e belonging to a particular social circle are assigned high link prediction scores.

$(s(u_i, u_j) = sim(u_i, u_j))$[1]. Therefore, the node's position in the network and his neighbors become an important source for link prediction algorithms to predict future connections. For example, if two individuals have many common friends, they are more likely to connect in the near future [8, 53] than those with small overlaps amongst their friends. This is a common approach for predicting future links in networks.

Despite the popularity of this approach, it is an oversimplified model for how people find and connect to new friends or acquaintances. Specifically they treat all the connections in the network homogeneously without any differentiating among the connections. However, this is not what we practice in our life when we form new connections. We usually do not connect to someone just because she is the most similar person to us. Instead we may connect to someone for a variety of different reasons; for example, she may be a classmate, a colleague from work, a family member, or someone with similar interests. These multiple "social dimensions" are not usually considered in link prediction algorithms.

To better understand the problem with link prediction algorithms that do not dif-

[1]In these equations, $\Gamma(u_i)$ is the set of u_i's neighbors, $sp(u_i, u_j)$ is length of the shortest path between u_i and u_j, and $sim(u_i, u_j)$ is the similarity score between u_i and u_j based on their common attributes or interests.

ferentiate the connections, let us look at the toy example demonstrated in Figure 3.1. User e has two social circles. She is connected to users 1 and 2 because they attend the same sports club. She is also connected to her colleagues, users 4, 5, and 6. Given this information, what is the best approach to predict new links for e? Should we consider e's entire neighborhood to predict new links or should we primarily use links to sporting club members to predict new links to other sporting club members, and likewise, primarily use links to colleagues to predict new links to other colleagues? What if e recently started a new job and began forming links with many colleagues in a short period of time? Should e's entire neighborhood be used to predict future links to other colleagues she may form connections with in the future? We propose a modification of neighborhood-based proximity measures, including common neighbors, that estimates an ego's social circles and detects bursts of link formations to predict future links. We find that our proposed approach is superior in link prediction accuracy to existing approaches that utilize all of an ego's neighbors.

Our main contributions are as follows:

1. From analyzing egocentric network data from Facebook and Google+, we discover that social circles and times of link formation play an important role in the choice of neighbors an ego connects to, with key implications for link prediction from an egocentric perspective.

    (a) We find that most neighbors connected to an ego belong to a single social circle, and the majority of links from neighbors are also connected to a single social circle. Since social circles are generally not known, we estimate them by clustering the egocentric network and find that the majority of links from neighbors also belong to a single cluster. These findings suggest that neighbors' cluster memberships should be taken into account when

predicting future links.

(b) We find that an ego typically expands his egocentric network at a steady rate; however, there are also periods where an ego adds many neighbors in short bursts of time. Furthermore we discover that neighbors added to an egocentric network in bursts often have very low similarity to existing neighbors of an ego, suggesting that typical proximity-based approaches that utilize entire neighborhoods would not be good predictors of these neighbors.

2. We propose an approach for local link prediction that takes advantage of the above discoveries by using egocentric clusters formed using the structure of the egocentric network and the times of link formations in the egocentric network. We demonstrate that incorporating these egocentric clusters into common link prediction proximity measures such as common neighbors improves prediction accuracy by about 10% on a Facebook and a Google+ data set.

In this chapter, we first study link formation in social networks. We then propose egocentric link prediction algorithm for two cases of expanding existing social circles and creation of new social circles.

Problem definition Social media users usually connect to others due to some common interests or affiliations. Observations show that individual's networks consist of more than one social circle, and each social circle can be mapped to a specific affiliation or interest. Figure 3.2 shows a sample egocentric network from Facebook. In the figure, the neighbors are clustered into social circles with different sizes, mostly representing the node's affiliations and interests. Three of these social circles are relatively large and covers a large portion of the neighbors.

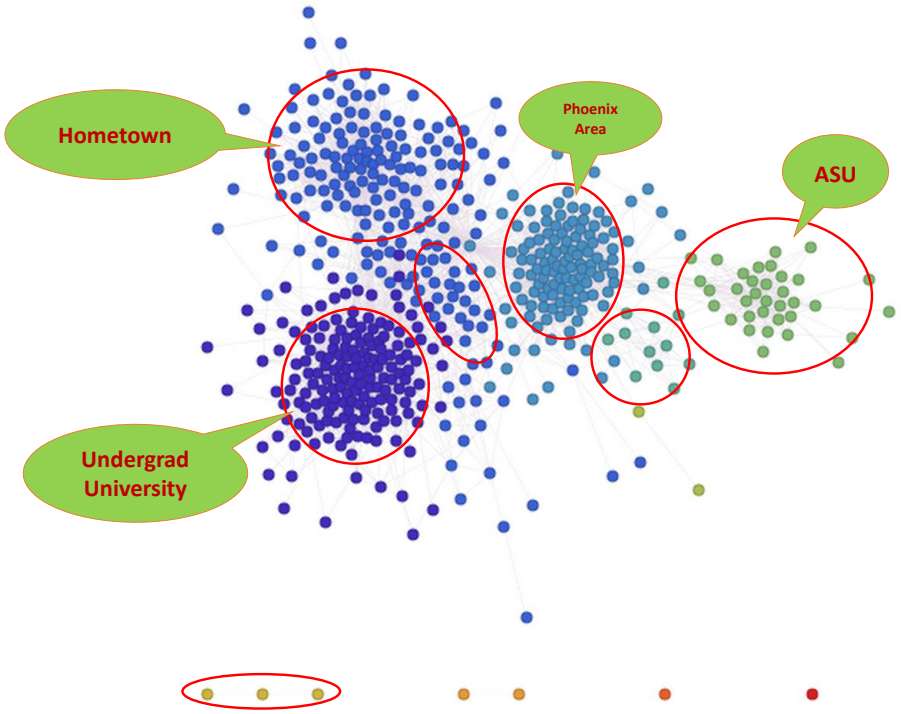


Figure 3.2: A Facebook user's egocentric network with its social circles.

The natural link formation approach which we practice in our everyday communications is to expand each of these social circles. For example, when we connect to a new classmate we add him to our existing social circle of classmates. Or our new colleague becomes part of our existing social circle of colleagues. Similarly, we find a new friend in our sports club, or find and connect to a neighbor in our neighborhood. In all of these examples, we expand only one social circle, regardless of its connections with other social circles, or regardless of the overall proximity between us and the target person whom we are about to connect to. A new colleague is expected to be similar and be connected to other colleagues. But we do not expect him to be similar to our other social circles such as our family circle or sports club circle.

3.2 Egocentric Link Formation

To better predict the future links, we need to understand the dynamic and the underlying causes of creation of links in social networks. Using network models is an effective approach on studying the link formation problem. We can design network models that generate, on a smaller scale, graphs similar to real-world networks. Under the assumption that these models simulate properties observed in real-world networks well, the analysis of real-world networks boils down to a cost-efficient measuring of different properties of simulated networks. In addition, these models allow for a) a better understanding of phenomena observed in real-world networks by providing concrete mathematical explanations, and b) allow for controlled experiments on synthetic networks when real-world networks are not available. *Random graphs* [21], *small-world network models* [78], and *preferential attachment* [12] are the most popular network models that describe how networks form and evolve. Preferential attachment and small-world network models are adapted the most, to explain the formation of links in social networks. According to preferential attachment model, chance of creation of new links between two nodes is directly correlated with the degree centrality of each of those nodes. Based on small-world network mode, nodes that are positioned closer have a higher chance to connect each other. In common neighbors [53] and Adamic Adar [8] algorithms, number of common neighbors are the most important indicators for predicting future links. These algorithms, usually, do not consider the heterogeneous aspect of social media users' connections, and treat all of the connections equally in which in many cases resulted in poor recommendation. As an example, consider the network in Figure 3.2 which is a real Facebook egocentric network. In this network, three of the social circles are dominant and cover majority of the neighbors. If we use network overlap approaches, such as *mutual friends*, to find similar

nodes, we give a higher chance to larger social circles to play dominant role. In this situation, smaller clusters will not get a change to be effective and play a role on link recommendation task. Consequently, the algorithm keeps recommending nodes that are connected to the node's largest social circle, resulting a growing network in a single dimension. In the given example, most of the recommendations made by Facebook, is related to the largest cluster which is the user's *hometown* circle. This also has a negative effect on expanding smaller social circles.

According to [67], social media users form different social circles each of which representing a distinct affiliation or interest of each individual. These social circles have interesting properties that we will talk about them in this chapter. For example, we show that majority of them are disjoint, and usually there is small overlap between them. To address these problems, we employ an egocentric approach to study how a user expands his network and connects to, or interacts with other users. Using the egocentric approach, we can study the heterogeneity of the connections and its effect on link formation problem.

In Figure 3.1 we assume that users' affiliations is known, however, in most of the social networks, this information is hardly accessible. Therefore clustering the users based on their affiliations or interests, and finding their social circles, is a challenging problem. We are interested in clustering the neighbors based on their affiliations or interests. However, the affiliation information is not available. In this research, we focus on every individual and their neighbors to analyze the structure of the egocentric network. This approach enables us to use the heterogeneity of the connections in social networks.

3.3 Analyzing Egocentric Networks

First, we need to study the structure of egocentric link formation and evaluate our hypothesis. The focus is on individual nodes and their connections to study egocentric link formation problem, and to study individuals' local network and their characteristics. More specifically, we are interested in answering the following questions: a) how connected are the neighboring nodes? b) what is the overlap between the clusters? Answering these questions helps us to understand link formation problem with an ego centric point of view.

3.3.1 Egocentric Local Clustering

Network clustering is our first step to analyze the egocentric network. The ego-centered network is a subset of the network that covers the node's immediate neighbors. In this step, we start with a node and cluster the network that includes the node and all of its neighbors and connections between them. For every ego node v_i, $G_{v_i} = (V', E')$, where $V' = \Gamma(v_i) \cup v_i$ and $E' \subseteq E, E' \in (V' \times V')$, where $\Gamma(v_i)$ is the set of v_i's neighbors. In a network with given nodes' attributes, it is recommended to use nodes' affiliations and interests to cluster the nodes into clusters [81]. However, most of the social media users do not reveal this type of information. In the absence of this information, network-based clustering such as spectral clustering is shown to be an effective approach to cluster the neighboring nodes into clusters of like minded nodes or those with common affiliations [75, 67]. To achieve this goal we employ spectral clustering to cluster the network in a way that there are minimum connections between clusters or minimize the size of the cut. The main problem with spectral clustering is that, in order to minimize the cut between partitions, the algorithm might return a single node as one cluster and rest of the network as the second

community to minimize the size of the cut. To address this problem, the objective function is modified in a way that considers the size of each clusters as well as size of the cut. Two popular spectral clustering techniques *Ratio cut* and *normalized cut* are commonly used for community detection problems in networks. These techniques partition the network in a way that the total number of edges between clusters is minimized. In a given graph $G(V, E)$, with k partitions of $P_1, P_2, \ldots, P_k$ such that $P_i \subseteq V$, $P_i \cap P_j = \emptyset$ and $\bigcup_{i=1}^{k} P_i = V$, the objective function for *Normalized Cut* and *Ratio Cut* is defined as follows:

$$\text{Normalized Cut}(P) = \frac{1}{k} \sum_{i=1}^{k} \frac{\text{cut}(P_i, \bar{P}_i)}{\text{vol}(P_i)}, \tag{3.1}$$

$$\text{Ratio Cut}(P) = \frac{1}{k} \sum_{i=1}^{k} \frac{\text{cut}(P_i, \bar{P}_i)}{|P_i|} \tag{3.2}$$

where $\bar{P}_i = V - P_i$ is the complement cut set, $cut(P_i, \bar{P}_i)$ is the size of the cut, and volume $vol(P_i) = \sum_{v \in P_i} d_v$. As we changed the objective functions to avoid unbalanced clusters, we might face another problem of having equal size clusters. Both ratio cut and normalized cut techniques enforce the clustering algorithm to have more balanced community sizes. However, normalized cut, which uses the number of vertices in the cutset, has less effect than ratio cut on forcing the algorithm to balance the size of the clusters. It is also shown [67] that the normalized cut is more effective on clustering nodes in social networks and extracting latent social dimensions. To reduce the balancing effect of spectral clustering, we ran the algorithm for different number of clusters ranging from 5 to 10 and select the clustering that generates the highest clustering coefficient.

Using this approach, we are able to partition every node's network into a few egocentric local clusters (5-10 clusters in our experiments). The clustering results are unique for every node. Examples of these local clusters are *family members*, *colleagues*, or *neighbors*. The size of egocentric networks are usually small and is

limited to one's number of neighbors (in order of few hundred nodes on Facebook), and consequently most of the clusters have small number of members (10- 50 is the most popular cluster sizes on our Facebook dataset). Global clustering approaches such as social dimensions take the entire network as input and partition the network into some global clusters. This clustering is based on some global connectivity between the nodes in the entire network and size of the clusters is usually large (depending on the size of the original network). For example, we can use this approach to cluster a network of politicians based on their political orientation. As we can see, the types of the clusters for local and global approaches are different. The former, generates clusters like family members or teammates, and the later approach returns different clusterings such as Republicans and Democrats. From this point of view, the main difference between these two approaches is the level of personalization on the clusterings. The egocentric clustering is personalized and unique for every user, but in global approaches we will have only one clustering which is shared for all nodes in the network.

3.3.2 Cluster Overlap

After clustering the egocentric networks, we are interested in to see what is the overlap between the social dimensions. Every cluster in an ego centric cluster represents a dimension in the ego's network such as family, colleagues, or classmates. This will provide an important measure to see how effective is one cluster's information on predicting future interactions in other social dimensions. Figure 3.3 shows the average number of clusters that one's friends are distributed in them. As we can see from the figure, 56% of the nodes are only connected to one social circle, and do not have any connections in other social dimensions. The clusters are highly disjoint,

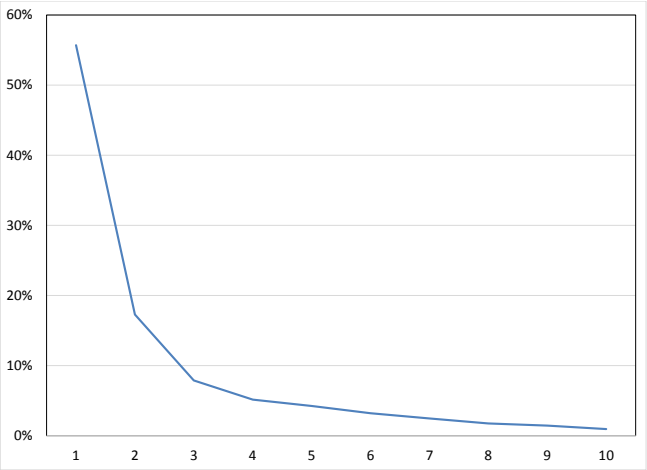


Figure 3.3: Number of clusters that every node in an egocentric network is connected to. The figure shows that 56% and 19% the nodes are connected to only one and two clusters respectively. In this figure, x-axis is the number of clusters that a node is connected to; and y-axis is the distribution.

and there is a small overlap between the egocentric clusters, which shows the single dimensionality of the connections in social networks. In very few cases, nodes have connections in more than one social circle. For example, there are very few nodes, if any, that have connections in one's family members, colleagues, and friends circles.

Following are our major findings from link formation in social networks from an egocentric point of view. 1) Nodes are multi-dimensional. Every node connects to a couple of social circles which represent different affiliations or interests. 2) Connections are single-dimensional. Every link connects to a pair of users with common affiliation or interest. Therefore, the connections can be labeled based on the commonality of the nodes. 3) Majority of the neighbors are also connected together (high clustering coefficient in the egocentric network). This coefficient is significantly higher when we measure it inside every social circle.

3.3.3 Observations

In this section, we analyzed the link formation in social network with an egocentric approach. We use spectral clustering to divide egocentric network into number of clusters. The analysis shows that every new connection is either an expansion to the existing social circles or adds a new social circle. Majority of the clusters are disjoint, therefore we cannot use one cluster to predict future links for other clusters. The results paved the way to propose our egocentric link prediction approach. In the proposed approach, instead of looking at the entire egocentric network to find the most similar individuals and recommend them to connect, we look at each node's social circles and try to find other users who matches this person the best. Our strategy for finding the best match is not having the most similarity with the entire neighborhood. We cluster the neighbors into some social circles and the recommendation would be along one of these circles. In the rest of this chapter, we study two problems, a) egocentric link prediction when new connections expand the existing social circles, and b) egocentric link prediction when new connections do not fit in the existing social circles and form a new social circle.

3.4 Egocentric Link Prediction with Existing Social Circles

Observations from the previous section show that, in most cases individuals connect to others by expanding their existing social circles. This observation strengthens our original skepticism about the effectiveness of using entire egocentric network to predict future connections. As we saw, networks expand around specific social circles. So it seems reasonable to use nodes in the social circle to predict possible extensions to that specific circle. In this section, we propose a new link prediction approach

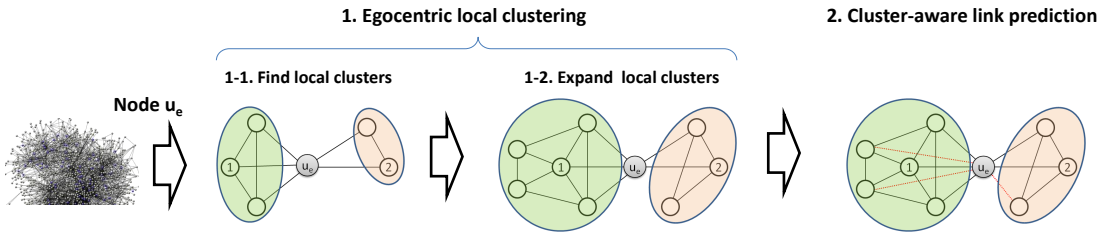


Figure 3.4: Egocentric link prediction process

Algorithm 3 Egocentric link prediction

Input: The network information $G(V, E)$

Output: A list of most likely links to appear in future

1: **for** Each user v_i in **V do**

2: $\Gamma(v_i) \leftarrow$ neighbors of v_i

3: *Egocentric local clustering:* $C_{v_i} \leftarrow$ Spectral Clustering ($\Gamma(v_i)$, k); k is number of social clusters.

4: *Prediction:* Calculate S_{v_i,v_j} using C_{v_i} and C_{v_j}, $v_j \in V \backslash v_i$

5: **end for**

that utilizes local social circles to predict future links in the network. The proposed algorithm has two major steps; 1) *Finding egocentric social circles*, and 2) predicting future links based on cluster information. The process is shown in Figure 3.4.

3.4.1 Link Prediction with Egocentric Local Clusters

Link prediction algorithms usually compute a score for every potential link, and rank these scores in descending order. Those with the highest score will be considered as missing or future links. We use a similar approach; however we include egocentric clustering information into the algorithm. We use the nodes in the social circles as the only source of information to predict future links. Here, the main assumption

is that the nodes expand their network by expanding their social dimensions. We assume that every time a node expands his network, it considers only one dominant feature (or affiliation) and the expansions form in that exact direction. Therefore, to find a new member for every social circle, we only need the information from nodes in that specific circle. For example, to find a new colleague, we only use user's existing colleagues, and do not consider the information from nodes in other circles. We repeat this process for every node and every circle that the node has. Therefore, we limit the neighborhood to the original cluster, and ignore members of other clusters. Then we aggregate the results for all of the circles. Following, we describe the proposed algorithms on link predicting with egocentric local clusters.

Cluster-based Common Neighbor (Ego-CN)

Cluster-based Common Neighbor (Ego-CN) is a local algorithm that computes the score based on number of common neighbors between two nodes ($S(u,v) = |\Gamma(u) \cap \Gamma(v)|$). In egocentric common neighbor algorithm we perform this equation for every cluster, and then aggregate the results.

$$S(c_{u_i}, c_{v_j}) = |\Gamma(c_{u_i}) \cap \Gamma(c_{v_j})| \tag{3.3}$$

where C_u is the set of u's social circles and $c_{u_i} \in C_u$.

Cluster-based Adamic-Adar (Ego-AA)

Cluster-based Adamic-Adar is another local link prediction algorithm, which weights the importance of common neighbors proportional to the inverse of the log of degree centrality of the node ($S(u,v) = \sum_{t \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log(|\Gamma(t)|)}$). AA-Ego calculates the similarity score by applying the above equation for every social circle, then uses the aggregation method that we described in the previous section to generate a single

scalar score for every pair of nodes.

$$S(c_{u_i}, c_{v_j}) = \sum_{t \in \Gamma(c_{u_i}) \cap \Gamma(c_{v_j})} \frac{1}{log(|\Gamma(t)|)} \tag{3.4}$$

In the following algorithms, we use a linear combination of the conventional link prediction algorithms and the egocentric algorithms introduced before.

Weighted Clusters Common Neighbor (Ego-CN+)

Weighted Clusters *Common Neighbor* considers a weighted combination of conventional and egocentric common neighbor approach to calculate the similarity score.

$$S_{CN+}(u, v) = S_{CN}(u, v) + \alpha S(c_{u_i}, c_{v_j}) \tag{3.5}$$

where $S_{CN}(u, v)$ is similarity of two nodes based on common neighbor method ($|\Gamma(u) \cap \Gamma(v)|$), $S(c_{u_i}, c_{v_j})$ is the egocentric common neighbor score which can be calculated using Equation 3.3, and α is a parameter that controls the contribution of each method.

Weighted Clusters Adamic-Adar (Ego-AA+)

Weighted Clusters Adamic-Adar is a weighted combination of conventional and egocentric Adamic-Adar approach to calculate the final score.

$$S_{AA+}(u, v) = S_{AA}(u, v) + \alpha S(c_{u_i}, c_{v_j}) \tag{3.6}$$

Aggregation Mechanisms

For every pair of nodes u and v, we calculate a similarity matrix $S_{|c_{u_i}| \times |c_{v_j}|}$. We use two approaches *Max* and *Sum* to aggregate these scores and convert them into one scalar score for every pairs of the nodes.

Max, returns the maximum similarity among the pairs of social circles belonging to two nodes. This measure considers the clusters with the highest similarity and uses those to compute overall node similarity. Max disregards the similarity of all other clusters except those with the highest.

$$S_{max}(u,v) = \max S(c_{u_i}, c_{v_j}), \tag{3.7}$$

for all i, j s.t. $c_{u_i} \in C_u,\ c_{v_j} \in C_v$

Sum, takes the summations of squared similarity among the pairs of social circles belonging to two nodes. It considers the similarity of all of the clusters into the final score. By taking the square, we boost the contribution of clusters with high overlap.

$$S_{sum}(u,v) = \sum S(c_{u_i}, c_{v_j})^2, \tag{3.8}$$

for all i, j *s.t.* $c_{u_i} \in C_u,\ c_{v_j} \in C_v$,

3.5 Egocentric Link Prediction with New Social Circles

In almost all of the link prediction algorithms, nodes' past behavior is the main indicator for their future behavior, which in general case is a promising assumption. In the previous section, we show that we can use the existing social circles and predict future links by expanding these social circles. However, a challenging problem in link prediction problem is that of the links occurring due to some real-world events, such as when an individual moves to a new location and makes new friends or finds new colleagues. In this situation, the new connections barely can be predicted using the node's existing links or social circles. In the case of real-world events that lead to new connections, the individuals usually have a limited (if any) network overlap with the

existing egocentric network, therefore the conventional link prediction algorithms are usually unable to predict further connections related to these events. Considering this fact, we can define two distinct forms of link formation behavior in social networks. In the first form, users expand their existing social circles and add friends to their existing social circles. In this case, new nodes fit in one or more of the existing social circles. Therefore, usually there is a high similarity between the existing network and new neighbors. In the second form, usually there is a low, if any, structural proximity between new connections and the existing nodes in the user's network. Therefore, the new connections create a new social circle. The former is extensively studied and almost all of the existing link prediction approaches try to address this type of connections. However, the second part which is forming new social circles is almost forgotten in link prediction studies. The main challenge on studying this problem is the lack of information on how to predict the new links, because of low similarity between the egocentric network and new connections. In this section, first we show that the existing algorithms are not well equipped to predict the links for the new connections that are happening as a result of users' social life events, then we expand our egocentric link prediction algorithm to cover this new problem.

3.5.1 Formation of New Social Circles

In the previous section, we show that social circles are almost disjoint and we cannot use information from one social circle to help finding others. For example, information form one's family members barely help expanding his classmates, colleagues, or sports club social circles. In this section, we use a real-world dataset to study the formation of new social circles. Usually emergence of a new cluster is associated with a real-world event in ego's life. In that situation, the node exposes to

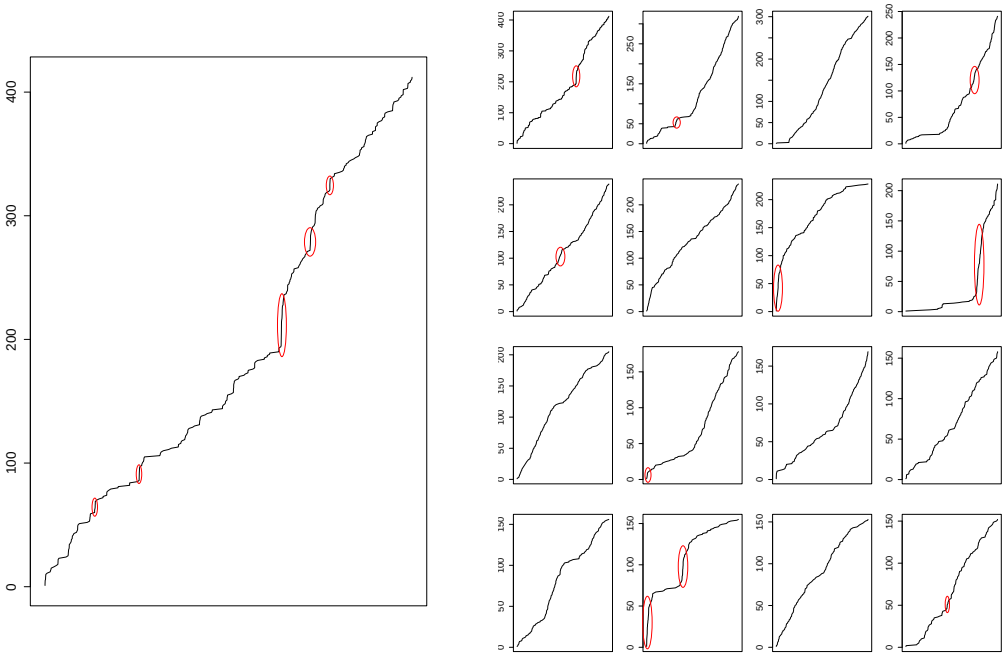


Figure 3.5: In the case of event-based link formation, nodes show a relatively intense activity in connecting to other people. In this figure, x-axis is time, and y-axis is number of friends a node has. The graph shows link formation for 10 different users.

a group of new individuals in which majority of them do not fit in any of the node's existing social circle. To address this problem, we propose *event-based link prediction*. The goal is to predict links that appear in the network due to some events. These events can be either real-world events such as moving to a new neighborhood or online events such as joining an online community. *The event-based link formation is the process of link formation when the most recently connected nodes hold the following criteria: having low structural similarity with the ego, having high similarity among themselves, and appearing in bursts.*

3.5.2 Detection of New Social Circles

Observations from previous section revealed that, while egocentric networks tend

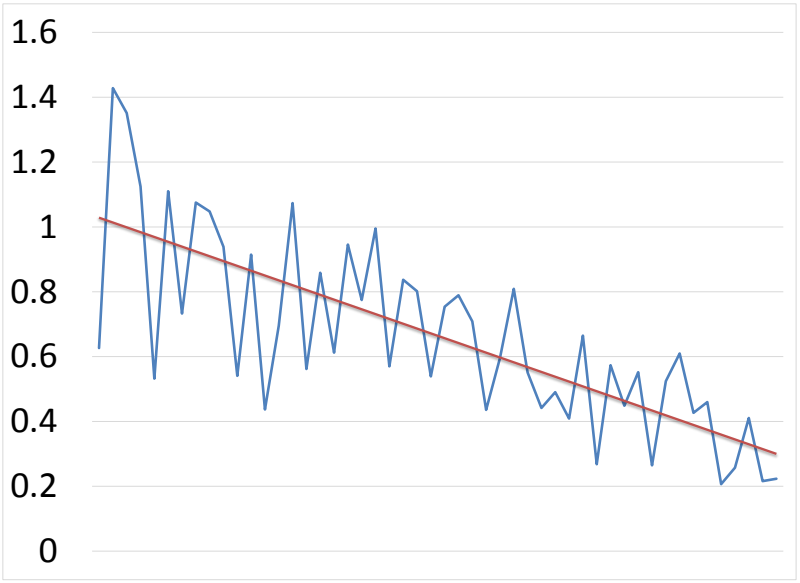


Figure 3.6: The figure shows the similarity of new connecting nodes with the existing network vs. the rate of connections. X-axis is similarity and y-axis is $d(n)/d(t)$. The graph shows that when nodes connect to many new nodes in a short period of time, the similarity between these nodes and the existing network decreases. The red line depicts the trend-line which is decreasing.

to grow at a steady rate, there are often short bursts of rapid growth. Within these short bursts, newly added neighbors exhibited low similarity with existing neighbors but high similarity amongst themselves, suggesting that they form a new social circle. We propose an algorithm that uses these characteristics to detect the emergence of such a new circle.

In order to detect the formation of a new social circle, we need both network structure and times of link formations in the form $G = (V, E, T)$, where T is the set of times at which each link was formed. The detection algorithm is triggered upon the addition of a neighbor v to the egocentric network for node e. First we check to see if a burst of rapid growth is taking place by comparing the time gap between v

Algorithm 4 Detection of new social circle for ego node e.

Input: Timestamped egocentric network $G_e = (V_e, E_e, T_e)$
Output: Members of new social circle $N_e \subset E_e$

$N_e \leftarrow \emptyset$ {Empty set denotes no new social circle}
for $i = 1$ **to** k_1 **do**
  $\Delta t_i \leftarrow$ time gap between ith and $(i+1)$th most recently added neighbors
end for
$\lambda_{k_1} \leftarrow (\Delta t_1 + \ldots \Delta t_{k_1})/k_1$ {Baseline growth rate}
if $\Delta t_1/\lambda_{k_1} \geq \tau_1$ **then**
  return {No burst detected}
end if
$v \leftarrow$ most recently added neighbor
$S(e, v) \leftarrow |\Gamma(e) \cap \Gamma(v)|$ {Similarity between e and v}
if $S(e, v) \geq \tau_2$ **then**
  return {Newly added node v is too similar to ego node e to be start a new circle}
end if
$V_{k_2} \leftarrow k_2$ most recently added neighbors
$L_{\text{new}} \leftarrow$ local clustering coefficient for nodes in V_{k_2}
$L_{\text{exist}} \leftarrow$ local clustering coefficient for nodes in $V_e \backslash v$
if $L_{\text{new}}/L_{\text{exist}} \leq \tau_3$ **then**
  return {Newly added nodes are too dissimilar to form a new circle}
end if
for all $v_i \in V_{k_2}$ **do**
  $S(e, v_i) \leftarrow |\Gamma(e) \cap \Gamma(v_i)|$ {Similarity between e and v_i}
  if $S(e, v_i) < \tau_2$ **then**
    $N_e \leftarrow N_e \cup v_i$ {Add v_i to new social circle}
  end if
end for

(the most recently added neighbor) and the neighbor added prior to v to the average time gap between the k_1 most recently added neighbors. This allows us to compare the instantaneous growth rate of the egocentric network to a sliding-window estimate of the baseline growth rate.

If the ratio between the most recent time gap and the baseline estimate is below a threshold τ_1, we then check the similarity between v and e using common neighbors. If the similarity is low, then v would not have been given a high link prediction score using a similarity measure that treats all links equally and suggests that a connection to v would not be highly predictable given the existing neighbors. If the similarity is

Algorithm 5 Link prediction for ego node e.

Input: Timestamped egocentric network $G_e = (V_e, E_e, T_e)$
Output: Link prediction scores for nodes in $V \backslash (V_e \cup e)$

$N_e \leftarrow$ new social circle detected by Algorithm 4
$C_e \leftarrow$ clusters of $V_e \backslash N_e$ {Perform egocentric clustering on all neighbors not in new social circle}
for all $v_i \in V \backslash (V_e \cup e)$ **do**
    for all $c_{e_j} \in C_e$ **do**
        Calculate $S(c_{e_j}, v_i)$ {Cluster-based similarity measure, e.g. Ego-CN or Ego-AA}
    end for
    Calculate $S(e, v_i)$ {Link prediction score for node v_i}
end for

below a threshold τ_2, we then compute the local clustering coefficient among the last k_2 most recently added neighbors and among all existing neighbors. This step is to verify that a new social circle is being created rather than a collection of dissimilar nodes. If the ratio of the local clustering coefficient among recently added nodes to the coefficient among all existing neighbors exceeds a threshold τ_3, we then consider the k_2 most recently added neighbors as candidates for a new social circle. We add all of the candidates with similarity less than threshold τ_2 with the ego node to the new social circle. This last step ensures that only new nodes that are highly dissimilar from existing neighbors are placed in the new social circle. The process for detecting the formation of a new social circle is shown in Algorithm 4.

After the creation of the new social circle, the remaining neighbors can be clustered as in Section 3.2 to estimate the remaining social circles, and the link prediction score can be computed as in Section 3.4.1. The entire link prediction process is shown in Algorithm 5.

3.6 Evaluation

We test our proposed Egocentric link prediction algorithms on two social network data sets, Facebook and Google+. Since we are taking an egocentric approach to link prediction, we are interested in the accuracy of our predictions for each ego node rather than for the entire network as a whole. Hence we evaluate accuracy using two egocentric metrics:

1. AUC, the area under the receiver operating characteristic (ROC) curve, calculated over all nodes at distance 2 from the ego; that is, all nodes that are candidates for edges based on common-neighbor approaches (those with non-zero similarity scores).

2. P@n, the precision over the n nodes with the highest similarity scores to the ego node.

We estimate social circles by modularity maximization using the Spectral clustering algorithm with a maximum of 10 clusters per egocentric network. For detection of new social circles, we choose the following parameters: $k_1 = k_2 = 10$ nodes, $\tau_1 = \tau_3 = 2$, and $\tau_2 = 4$.

3.6.1 Facebook Dataset

We first test our Ego-LP algorithms on the Facebook data set from Viswanath et al. [1, 74], which consists of over $60,000$ nodes and $800,000$ edges along with the times at which edges were formed. The link prediction results on the Facebook data are summarized in Table 3.2. The accuracy metrics are averaged over all of the nodes in the data set. The cluster-based link prediction scores perform better than the baseline link prediction score, with the percentage improvement typically around 8%. When we use a combination of conventional link prediction algorithm

Table 3.1: Link prediction accuracy metrics for Google+ data. The mean over all ego nodes in the data set is shown. Quantities in parentheses denote percentage improvement over the same link predictor without clusters. Best performer for each metric is shown in bold.

Metric	No clusters		With clusters		No Cluster + $\alpha\times$ With clusters	
	CN	AA	CN	AA	CN	AA
AUC	0.614	0.654	0.680 (10.8%)	0.695 (6.2%)	0.702 (7.3%)	**0.720** (10.1%)
P@10	0.147	0.164	0.146 (−0.9%)	0.169 (2.8%)	0.174 (6.1%)	**0.179** (9.2%)
P@20	0.143	0.159	0.145 (1.1%)	0.159 (−0.1%)	0.168 (5.4%)	**0.170** (7.2%)
P@50	0.141	0.154	0.146 (3.9%)	0.157 (1.9%)	0.163 (6.1%)	**0.164** (6.4%)

and our egocentric link prediction algorithm, the accuracy improves even more up to 9.3%. Since the accuracy measures are computed over egocentric networks, they are easily interpreted for the task of link recommendation, i.e. actively suggesting links for the ego to connect to. The P@n measure is particularly well-suited for this type of interpretation. In our experiments our selection for n is 10, 20, and 50. Given our results, if we were to recommend nodes for an ego to connect to, roughly 1 in 6 recommendations would be relevant.

3.6.2 Google+ Dataset

Next we test our Ego-LP algorithms on the Google+ data set from Gong et al. [24], which consists of over $5,000$ nodes and $14,000$ edges. Unlike in the Facebook data set, times at which edges were formed are not available, so we cannot utilize the new social circle detection algorithm. However, we do have snapshots of the network at 4 different time steps, so for each time step $t \in \{1, 2, 3\}$, we train the link predictor using all edges observed at or prior to time t and attempt to predict edges formed in later time steps.

The link prediction results on the Google+ data are summarized in Table 3.1.

Table 3.2: Link prediction accuracy metrics for Facebook data. The mean over all ego nodes in the data set is shown. Quantities in parentheses denote percentage improvement over the same link predictor without clusters. Best performer for each metric is shown in bold.

Metric	No clusters		With clusters		No Cluster + $\alpha\times$ With clusters	
	CN	AA	CN	AA	CN	AA
AUC	0.645	0.667	0.691 (7.1%)	**0.740** (11.0%)	0.698 (8.2%)	0.715 (7.2%)
P@10	0.155	0.172	0.172 (11.0%)	**0.186** (8.4%)	0.169 (9.3%)	0.185 (7.3%)
P@20	0.152	0.165	0.163 (7.2%)	0.174 (5.2%)	0.166 (9.0%)	**0.175** (5.4%)
P@50	0.147	0.159	0.149 (1.3%)	0.170 (7.1%)	0.156 (6.2%)	**0.172** (8.1%)

Table 3.3: Link prediction accuracy metrics for Facebook data with new social clusters. The mean over all ego nodes in the data set is shown. Quantities in parentheses denote percentage improvement over the same link predictor without clusters. Best performer for each metric is shown in bold.

Metric	No clusters		With clusters		With clusters + new social circle	
	CN	AA	CN	AA	CN	AA
AUC	0.645	0.667	0.691 (7.1%)	0.740 (11.0%)	0.730 (13.2%)	**0.748** (12.1%)
P@10	0.155	0.172	0.172 (11.0%)	0.186 (8.4%)	0.173 (11.4%)	**0.188** (9.2%)
P@20	0.152	0.165	0.163 (7.2%)	0.174 (5.2%)	0.166 (9.4%)	**0.176** (6.7%)
P@50	0.147	0.159	0.149 (1.3%)	**0.170** (7.1%)	0.159 (8.5%)	0.169 (6.6%)

For both common neighbors and Adamic-Adar, the accuracy is generally better with clusters, and the improvement is in the same range as for the Facebook data. In two instances, the P@n is actually slightly worse than without clusters, on average. This may be partially due to errors in the estimation of the social circles. Unlike Facebook, the Google+ egocentric networks are directed networks, but the modularity maximization procedure for clustering the network requires undirected networks, so we reciprocated edges. We believe there is potential to improve upon these results by utilizing a clustering algorithm for directed networks.

Table 3.3 shows the results of using our event-based link prediction algorithm comparing with two baselines. The first baseline is the conventional link prediction algorithms and the second baseline is egocentric link prediction. For each node, we run the detection algorithm to detect new social circles. Let $\mathbf{t}^{\text{new}}$ denote the times at which we detect the formation of new social circles and $|\mathbf{t}^{\text{new}}|$ denote the number of new social circles detected. At each time $t_i^{\text{new}} \in \mathbf{t}^{\text{new}}$, we train the link predictor and attempt to predict edges formed beyond time t_i^{new}. We also arbitrarily select $|\mathbf{t}^{\text{new}}|$ time steps where we repeat the previous procedure to provide a fair comparison to the baselines. When we add in detection of new social circles as well, the accuracy metrics improve even more than the egocentric algorithm, up to around 13.2%. The improvement is especially pronounced when using common neighbors as the proximity measure.

3.6.3 Discussions

In this chapter we proposed a local link prediction approach that incorporates egocentric clusters and detection of new social circles. We found that the majority of edges from neighbors of an ego node connect to a single social circle, suggesting that nodes often connect to other nodes along a single social dimension. We computed egocentric clusters to estimate the social circles, since circle memberships are often not available. We also found that, while egocentric networks typically grow at a steady pace, there were sometimes bursts of growth where many new neighbors are added. These neighbors were often very similar to each other but very dissimilar to the ego's existing neighbors, suggesting that these bursts may have been triggered by an event in the ego's life resulting in the formation of a new social circle. For such neighbors, existing links did not serve as good predictors for future links; hence

we proposed an approach to detect new social circle formation. We found that our proposed cluster-aware proximity measures for link prediction generally improved link prediction accuracy by about 11% on a Facebook and a Google+ data set.

Our study has several limitations. We examined the role of social circles in link formation as well as the role of link formation times. These two aspects were examined using two different data sets, one consisting of static egocentric networks with labeled circles, and one consisting of dynamic networks with link formation times but no information about circles. We found that the majority of links from neighbors of an ego connect to only one social circle. We took this as an indication that an ego node typically grows his egocentric network by adding nodes that fit his existing social circles rather than nodes that bridge multiple circles. If the latter was true, we would expect to see more links from neighbors to multiple social circles; however, this can only be confirmed by data on dynamic egocentric networks with both link formation times and labeled circles. We are unaware of the existence of such data, and collecting this type of data would be a useful development for future work.

Another limitation of our work involves link recommendation. The growth of egocentric networks on social networking sites including Facebook and Google+ is not an organic process because the sites also provide link recommendations. As such, a link prediction algorithm that mimics the link recommendation algorithm for a social network site, say Facebook, should be able to achieve excellent accuracy on Facebook data, but the results may not generalize to other sites or other link prediction settings. Our results showed similar improvements in link prediction accuracy for data both from Facebook and Google+, so it is less likely that our algorithms are mimicking the link recommendation algorithm of any particular site. This problem can be circumvented by conducting a randomized experiment rather than studying observational data. Since our proposed link predictors are local and require only access to nodes

at distances 1 and 2 from the ego, e.g. friends and friends of friends, respectively, it may be possible to run such a randomized experiment on Facebook or other social network site, and this is another interesting area for future work.

A third problem involves the detection of new social circles. The focus of this work is on link prediction, and as such, we devised a heuristic approach to detect the formation of new social circles in order to better predict future links. However the problem of detecting new social circles as an egocentric network grows is an interesting problem in itself and is quite different from the problem of learning social circles in static egocentric networks.

The experiments show that the egocentric approach improves the accuracy of link prediction comparing with conventional unsupervised approaches. However, the weighted combination of egocentric and conventional algorithms performs the best (Tables 3.1 and 3.2). One explanation for this improvement is that our clustering algorithm does not cluster the egocentric network properly. For example, we use maximum number of 10 clusters which might not be enough for some of the nodes. In reality some of the clusters have overlapped but our clusters do not have any overlap and each node is assigned to only one cluster. In addition, one of the objectives of the spectral clustering algorithm is to prevent clusters with different sizes. We used normalized cut to reduce this effect, however it affects the clustering task and forces the clusters to get closer in terms of having similar sizes. For all of these reasons, a pure egocentric algorithm does not generate the best results, and as we can see from 3.2 a combination of egocentric and conventional algorithms perform the best.

In this chapter, we have used most common proximity based algorithms including *Common Neighbors* and *Adamic Adar* as our baselines to evaluate the accuracy of the proposed egocentric and event-based algorithms. As we did not use any specific feature from these algorithms, it is easy and straight forward to use other link pre-

diction algorithms; including supervised approaches, and evaluate their performance when used with the proposed egocentric algorithms.

3.7 Summary

Link prediction is the process of predicting the most likely links to appear in the network in the future. The common approach on link prediction is to find the most similar nodes and recommend them to connect to each other. In this chapter, we showed that this approach does not match the way we find and connect to new friends in our real world experiences. Our connections on social networking sites can be clustered into groups of people with similar affiliations or interests, which are called social clusters. Our experiments show that we usually grow our network by expanding one of these so called social clusters or create a new social cluster. Using this fact, we proposed an egocentric link prediction algorithm. In this algorithm, we first cluster the ego centric network, and then use these clusters to predict future links. The results show a significant improvement comparing with the equivalent conventional algorithms.

Chapter 4

LITERATURE REVIEW

This chapter gives an overview on the related work on attribute prediction and link prediction in social networks.

4.1 Attribute Prediction

Predicting an individual's interests and preferences based on various cues from the individual and his environment has a long history in social science [26]. It has also attracted attentions in terms of using social media data to predict users' personal attributes and preferences. Predicting users' personal attributes, such as age, gender, location and political orientations and their interests and preferences is the core of many studies [48]. The advent of participatory web has enabled information consumers to become information producers via social media. This phenomenon has attracted researchers of different disciplines including social scientists, political parties, and market researchers to study social media as a source of data to explain human behavior in the physical world [2, 4]. With the availability of social media data and huge amount of user-generated data, it has been shown that we are able to investigate users' preferences by studying their online activities, postings, and behavior in social media [55]. There are plenty of studies are showing that it is possible to use information available in social networking sites to infer users' missing attributes such as age, gender, education level, political orientation and users' interests and preferences [50, 51, 42, 48, 16].

According to the type of information the prediction algorithms use, we can categorize them into *Content-based* and *network-based* approaches. *Content-based* ap-

proaches use user generated data, such as text, user profile, weblogs, product reviews, and status updates, to infer user preferences. They usually use classification algorithms to predict users' preferences. Support vector machines (SVMs) [57], Latent Dirichlet Allocation (LDA) [18], and boosted decision trees are the most prominent algorithms which is used in this category. Content-based approaches also can use users' historical information, such as credit card purchases, rating history, buying history, or browsing information, to infer users' preferences [19]. Preferences also could be directly inferred from analyzing the users' historical data such as log or browsing data. [28], [51], and [48] investigate the use of website browsing logs and the content of personal websites to predict personal attributes. Mislove et al. [50] show that the attributes of users, in combination with the social network graph, can be used to predict the attributes of other users in the same network. They use Facebook data and show that when only 20% of the nodes in the network reveal their personal attributes (including major, department, and year), it is possible to infer other users' attributes with an accuracy of over 80%. Tan et al. [65] use Twitter mention (@) data to construct a network and show that users who mention each other in their tweets, are more likely to hold similar opinions. Similar results also reported by [28]. Conover et al. [18] report an accuracy of up to 95% when predicting users' political orientation by employing users' content in combination with network information on Twitter. Carter et al. in [32] use the network structure and users' positions within a friendship network on Facebook to accurately predict users' sexual orientation. Kosinski et al. [36] use users' Facebook records to show the degree to which relatively basic digital records of social media users' behavior can be used to accurately predict a wide range of personal attributes. They use Facebook *likes* to extract users' positive association with online content, such as photos, videos, Facebook pages of products, businesses, people, books, places, and websites. They show that

it is possible to accurately predict users' basic demographic attributes, such as age, gender, relationship status, and personal traits such as political orientation, education level, sexual orientation, religion, and personality. They report that their model correctly discriminates between homosexual and heterosexual men in 88% of cases, between African-Americans and Caucasian-Americans in 95% of cases, and between Democrats and Republicans in 85% of cases. In most of the aforementioned studies, the authors used matrix factorization methods such as singular value decomposition (SVD) to reduce the size of the features. However, despite the performance of the proposed approaches based on global network information, scalability remains as the main challenge. *Random Projection* is an approach to address this challenge and is frequently used in information retrieval and text analysis [41] in lieu of statistical dimension reduction algorithms like SVD and Latent Semantic Analysis (LSA). This approach is computationally more efficient than matrix factorization methods, however produces comparable accuracy.

Network-based approaches use users' friendship or interaction information to predict their preferences. Most of the algorithms in this category, use the simple but effective social theories of *homophily* and *influence*, which indicates the similarity of connected users [81]. Relational learning or within-network classification [47] refers to the classification when data instances are presented in a network format. The data instances in the network are not independently identically distributed (i.i.d.) as in conventional data mining. To capture the correlation between labels of neighboring data objects, typically a Markov dependency assumption is assumed. That is, the labels of one node depend on the labels (or attributes) of its neighbors. Normally, a relational classifier is constructed based on the relational features of labeled data, and then an iterative process is required to determine the class labels for the unlabeled data. The class label or the class membership is updated for each node while the

labels of its neighbors are fixed [68]. This process is repeated until the label inconsistency between neighboring nodes is minimized. It has been shown [47] that a simple weighted vote relational neighborhood classifier [46] works reasonably well on some benchmark relational data and is recommended as a baseline for comparison.

4.2 Link Prediction

Many real world systems can be described as networks, where nodes represent entities and links represent connections and interactions between entities [44, 83, 73]. In social networks, nodes represent individuals and links represent friendship, collaboration, interaction, or influence between individuals [43]. Most of the problems that have entities and relation among the entities can be modeled as a network problem. Many disciplines study networks to analyze their complex relational data. The first known study on networks is the famous Seven Bridges of Königsberg. Leonhard Euler in his 1736 paper proposed a mathematical description of vertices and edges that later became the foundation of graph theory [64]. Due to the growing interest in using networks in different disciplines, the study of complex networks become a common focus of many researchers in different branches of science [8, 27, 54, 78]. An important scientific issue relevant to network analysis is the problem of predicting the relations between entities of the network, which is referred as *link prediction* problem. *Link prediction problem is the process of predicting the most likely links to appear in the network in near future.* The most common approach to predict future links is based on structural proximity between the nodes such as number of mutual friends [44, 43]. After measuring the proximity between the nodes in the network, algorithms recommend the most similar disconnected nodes to connect each other.

Link prediction algorithms can be categorized into *unsupervised* and *supervised* algorithms [10]. A common process among the unsupervised approaches is to find

similarity among the nodes and recommend the most similar nodes to connect in the future. Unsupervised methods use different algorithms to measure the similarity between the nodes. A common approach is to use *node neighborhood* to calculate the similarity between the nodes. *Common Neighbors* [53], *Adamic Adar* [8], *Jaccard Index*, and *Preferential Attachment* [12] are some of the most popular algorithms that use local structure of the network to measure the similarity between the nodes. These algorithms follow the natural intuition that if two nodes u and v have many common friends, they are more likely to come into contact in the future than a pair of random nodes. Another commonly used approach is to use global network structure to calculate similarity between nodes. *Katz* [33], *Random-walk with restart* [80, 70], and *SimRank* [29] are popular algorithms that use this approach. Algorithms in this category usually use the length of the path between two nodes as a measure of proximity. Their underlying assumption is that *shorter the path is, more similar the nodes are*. Some studies such as [11] use nodes' attributes and content information to calculate the similarity between them.

In supervised link prediction, the problem is modeled as a machine learning problem (usually a binary classification problem) of predicting unobserved links [9]. A model is trained based on the observed links and their features. The model then is used to label unobserved links as positive or negative. Positive labels mean the prediction for future links. In this approach, we can use various supervised learning/classification algorithms like decision tree and support vector machines (SVM) to predict whether a link exists between two pairs of nodes or not. Another common approach is to model link prediction problem as a classification problem. The prediction task is to predict whether there should be a link between two specific nodes or not. Many algorithms based on relational learning, use supervised machine learning in link prediction problem. Supervised approaches show a great success in link prediction

domain, however scalability is the major challenge for these approaches [10].

Cluster-based link prediction is another approach to use cluster information to improve the accuracy of link prediction algorithms. The common practice in this approach is to use cluster information on top of other link prediction algorithms such as proximity-based approaches. In this approach, the prediction algorithm takes input from a classical link prediction algorithm, and gives higher weights to those links that appear in the same cluster [35, 13].

Chapter 5

CONCLUSIONS AND FUTURE WORK

Individuals use social media sites to connect, interact, share, and create user-generated data. This rich environment provides tremendous opportunities for many different players to easily and effectively reach out to people, interact with them, influence them, or get their opinions. Due to the availability of the data on this platform, it provides a fertile field with many great opportunities and challenges for data mining. In this dissertation, we use an egocentric approach to address attribute and link prediction in social media. The key contributions of this work are summarized below, followed by future work.

5.1 Key Contributions

The contributions of this dissertation are (1) proposing a framework to study social media users through their attributes and link information, (2) proposing a scalable algorithm to predict user attributes; and (3) proposing a novel approach to predict attributes and links with limited information. The proposed algorithms use an egocentric approach to improve the state of the art algorithms in two directions. First by improving the prediction accuracy, and second, by increasing the scalability of the algorithms.

5.2 Future Work

In the previous chapters, we have discussed attribute and link prediction in social networking sites. Following the proposed prediction framework, there are many

promising directions to explore for future work. We highlight two of them below.

5.2.1 Comparative Study of LSocDim and Random Projection

The proposed LSocDim algorithm in some aspects, including performance improvement and scalability, is similar to random projection method. *Random Projection* is an approach to address scalability challenge and is frequently used in information retrieval and text analysis [41, 58, 59] in lieu of statistical dimension reduction algorithms like SVD and Latent Semantic Analysis (LSA). This approach is computationally more efficient than matrix factorization methods, however produces comparable accuracy. From this point of view both of the algorithms try to address scalability problem by avoiding statistical dimension reduction algorithms. Random projection method is widely used in information retrieval and text analysis and our LSocDim is designed for network data. These algorithms have similar goals and try to address a similar problem, however use different approaches to achieve their goals. Our proposal for future work in this direction is to have a comparative study between these two algorithms and assess their efficiency and effectiveness on text and network analysis.

5.2.2 Jointly Prediction of Links and Attributes

In this research, we propose algorithms to predict nodes' attributes and their future or missing links. In our algorithms we make use of the existing network structure to predict attributes. In almost all of the experiments, we have observed that the more we know about the nodes' connections, the better we can predict their attributes. One possible extension to this work is to first predict further links then use the new network to further predict the attributes. As the related work [23, 14] suggest, we

expect to achieve a higher accuracy comparing the case that we only use the original network information to predict links or attributes. By using the new network with predicted edges, every node has more connections and those connections are expected to help achieving higher accuracy on attribute prediction. In the special case of silent user, or users with very limited data, this approach might be even more effective than the average users.

In addition, the effect of social influence and homophily suggests that nodes' attribute information helps predicting the missing and future links. Therefore, we suggest using both the network structure information and nodes' attribute information together to improve the accuracy of link prediction task. Our expectation is to achieve higher attribute prediction accuracy using the results from link prediction task, and achieve higher link prediction accuracy by using attribute information. We also propose to alternate these processes until no longer improvement can be achieved. This approach is expected to further improve both of the attribute and link prediction task, especially in the case of limited data including silent user in attribute prediction and event-based in link prediction. However, it might introduce noise to the system, especially when we alternate the algorithms and use the results of one section on predicting the other one, which needs to be studied in future.

5.2.3 Egocentric Movie Recommendation

Recommender systems try to find items that match the best with users' preferences. To define users' preferences, recommender systems either use users' long time activity information or their last activity. The former is often used by movie recommender systems such as Netflix, and the later is mostly used by online retailers such as Amazon. Our preliminary observations show that users' preferences and interests change overtime. For example, a machine learning scholar changes his interest from

one topic to another one, due to the involvement in a new project. Another example is a person who was constantly watching movies from a specific director or actor, starts watching movies from a different actor or genre. This behavior is commonly observable among people. Their interests changes once a while and they become interested in a new set of items or activities (e.g., movies, news items, research papers). These interests and preference last for a while then will be replaced by new ones. Our proposal for the future work is to take a similar approach as we presented for egocentric link prediction to cluster the activities and cluster the items, and then use these clusters to better predict the future items of interests for the user. The proposed recommender system has two major steps; 1) Detecting the taste-clusters, and 2) use these taste-clusters to recommend new items to the user.

REFERENCES

[1] Facebook friendships network dataset – KONECT, June 2014.

[2] M. A. Abbasi, S.-K. Chai, H. Liu, and K. Sagoo. Real-world behavior analysis through a social media lens. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 18–26. Springer, 2012.

[3] M. A. Abbasi, S. Kumar, J. A. Andrade Filho, and H. Liu. Lessons learned in using social media for disaster relief-asu crisis response game. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 282–289. Springer, 2012.

[4] M. A. Abbasi and H. Liu. Measuring user credibility in social media. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 441–448. Springer, 2013.

[5] M. A. Abbasi, J. Tang, and H. Liu. Scalable learning of users preferences using networked data. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 4–12. ACM, 2014.

[6] M. A. Abbasi, J. Tang, and H. Liu. Trust-aware recommender systems. 2014.

[7] M. A. Abbasi, R. Zafarani, J. Tang, and H. Liu. Am i more similar to my followers or followees? homophily effect in directed online social networks. In *25th ACM Conference on Hypertext and Social Media*, 2014.

[8] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[9] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[10] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[11] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[12] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[13] P. N. Bennett, K. Svore, and S. T. Dumais. Classification-enhanced ranking. In *Proceedings of the 19th international conference on World wide web*, pages 111–120. ACM, 2010.

[14] M. Bilgic, G. M. Namata, and L. Getoor. Combining collective classification and link prediction. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 381–386. IEEE, 2007.

[15] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[16] A. Chaabane, G. Acs, M. A. Kaafar, et al. You are what you like! information leakage through users interests. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*, 2012.

[17] R. Cohen and D. Ruths. Classifying political orientation on twitter: Its not easy! In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[18] M. D. Conover, B. Gonçalves, J. Ratkiewicz, A. Flammini, and F. Menczer. Predicting the political alignment of twitter users. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 192–199. IEEE, 2011.

[19] K. De Bock and D. Van den Poel. Predicting website audience demographics forweb advertising targeting using multi-website clickstream data. *Fundamenta Informaticae*, 98(1):49–70, 2010.

[20] C. Desrosiers and G. Karypis. Within-network classification using local structure similarity. In *Machine Learning and Knowledge Discovery in Databases*, pages 260–275. Springer, 2009.

[21] P. Erdos and A. Rényi. On the evolution of random graphs. *Bull. Inst. Internat. Statist*, 38(4):343–347, 1961.

[22] L. Getoor and C. P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.

[23] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, D. Song, et al. Jointly predicting links and inferring attributes using a social-attribute network (san). *arXiv preprint arXiv:1112.3265*, 2011.

[24] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song. Evolution of social-attribute networks: measurements, modeling, and implications using google+. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 131–144. ACM, 2012.

[25] S. Gosling, D. Drummond, and I. NetLibrary. *Snoop: What your stuff says about you*. BBC Audiobooks America, 2008.

[26] S. D. Gosling, S. J. Ko, T. Mannarelli, and M. E. Morris. A room with a cue: personality judgments based on offices and bedrooms. *Journal of personality and social psychology*, 82(3):379, 2002.

[27] J. W. Grossman. The evolution of the mathematical research collaboration graph. *Congressus Numerantium*, pages 201–212, 2002.

[28] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160. ACM, 2007.

[29] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

[30] D. Jensen. Statistical challenges to inductive inference in linked data. In *Seventh International Workshop on Artificial Intelligence and Statistics*, pages 569–571, 1999.

[31] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 593–598. ACM, 2004.

[32] C. Jernigan and B. F. Mistree. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14(10), 2009.

[33] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[34] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[35] J. Kim, M. Choy, D. Kim, and U. Kang. Link prediction based on generalized cluster information. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 317–318. International World Wide Web Conferences Steering Committee, 2014.

[36] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[37] J. A. Krosnick, C. M. Judd, and B. Wittenbrink. The measurement of attitudes. *The handbook of attitudes*, pages 21–76, 2005.

[38] S. Kumar, G. Barbier, M. Abbasi, and H. Liu. Tweettracker: An analysis tool for humanitarian and disaster relief. In *Fifth International AAAI Conference on Weblogs and Social Media, ICWSM*, 2011.

[39] S. Kumar, F. Morstatter, and H. Liu. Twitter data analytics, 2013.

[40] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[41] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296. ACM, 2006.

[42] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1023–1031. ACM, 2012.

[43] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[44] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[45] Q. Lu and L. Getoor. Link-based classification. In *ICML*, volume 3, pages 496–503, 2003.

[46] S. A. Macskassy and F. Provost. A simple relational classifier. Technical report, DTIC Document, 2003.

[47] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8:935–983, 2007.

[48] B. Marcus, F. Machilek, and A. Schütz. Personality in cyberspace: personal web sites as media for personality expressions and impressions. *Journal of Personality and Social Psychology*, 90(6):1014, 2006.

[49] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[50] 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*, pages 251–260. ACM, 2010.

[51] D. Murray and K. Durrell. Inferring demographic attributes of anonymous internet users. In *Web Usage Analysis and User Profiling*, pages 7–20. Springer, 2000.

[52] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[53] M. E. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[54] M. E. Newman. The structure and function of networks. *Computer Physics Communications*, 147(1):40–45, 2002.

[55] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11:122–129, 2010.

[56] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[57] D. Rao and D. Yarowsky. Detecting latent user properties in social media. In *Proc. of the NIPS MLSN Workshop*, 2010.

[58] M. Sahlgren. An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5, 2005.

[59] L. Sellberg and A. Jönsson. Using random indexing to improve singular value decomposition for latent semantic analysis. In *LREC*, 2008.

[60] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266. ACM, 2008.

[61] B. Smyth and P. Cotter. A personalized television listings service. *Communications of the ACM*, 43(8):107–111, 2000.

[62] F. Stutzman, R. Gross, and A. Acquisti. Silent listeners: The evolution of privacy and disclosure on facebook. *Journal of privacy and confidentiality*, 4(2):2, 2013.

[63] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004.

[64] J. J. Sylvester. On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, with three appendices. *American Journal of Mathematics*, 1(1):64–104, 1878.

[65] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM, 2011.

[66] J. Tang, Y. Chang, and H. Liu. Mining social media with social theories: A survey. *SIGKDD Explorations*, 2014.

[67] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.

[68] L. Tang and H. Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1107–1116. ACM, 2009.

[69] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Advances in neural information processing systems*, page None, 2003.

[70] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. 2006.

[71] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[72] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election forecasts with twitter how 140 characters reflect the political landscape. *Social Science Computer Review*, 29(4):402–418, 2011.

[73] J. C. Valverde-Rebaza and A. de Andrade Lopes. Link prediction in complex networks based on cluster information. In *Advances in Artificial Intelligence-SBIA 2012*, pages 92–101. Springer, 2012.

[74] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in Facebook. In *Proc. Workshop on Online Social Networks*, pages 37–42, 2009.

[75] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[76] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55–67, 2008.

[77] F. Wang, C. Zhang, H. C. Shen, and J. Wang. Semi-supervised classification using linear neighborhood propagation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 160–167. IEEE, 2006.

[78] D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[79] Y. Yang, P. Cui, W. Zhu, and S. Yang. User interest and social influence based emotion prediction for individuals. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 785–788. ACM, 2013.

[80] Z. Yin, M. Gupta, T. Weninger, and J. Han. A unified framework for link recommendation using random walks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 152–159. IEEE, 2010.

[81] R. Zafarani, M. Abbasi, and H. Liu. *Social Media Mining, An Introduction*. Cambridge University Press, 2014.

[82] 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 *Proceedings of the 18th international conference on World wide web*, pages 531–540. ACM, 2009.

[83] T. Zhou, L. Lü, and Y.-C. Zhang. Predicting missing links via local information. *The European Physical Journal B-Condensed Matter and Complex Systems*, 71(4):623–630, 2009.

BIOGRAPHICAL SKETCH

Mohammad Ali Abbasi joined the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University in Fall 2009 and began his Ph.D. in Computer Science under supervision of Dr. Huan Liu. He received his Bachelors degree in Computer Engineering from Sharif University of Technology in 1997 and his Masters degree in computer engineering from University of Tehran in 2001. His research interests include Customization and User Profiling, Recommender Systems, Information Retrieval, Social Network Analysis, Data Mining, and Machine Learning. He coauthored a textbook titled "Social Media Mining: An Introduction", a book chapter, and several peer reviewed conference papers. He presented a tutorial at ICDM 2013. He was invited as a program committee (PC) member for SBP 2013, SBP 2014, and ICWSM 2014. He also served as reviewer for top tire conferences and journals.