Mining user similarity in online social networks : analysis,modeling and applications

Xiao Han

DOCTORAT EN CO-ACCREDITATION
TÉLÉCOM SUDPARIS - INSTITUT MINES-TÉLÉCOM
ET L'UNIVERSITÉ PIERRE ET MARIE CURIE - PARIS 6

Spécialité: Informatique

Ecole doctorale: Informatique, Télécommunications et Électronique de Paris

Présentée par

Xiao HAN

Mining User Similarity in Online Social Networks: Analysis, Modeling and Applications

Soutenue le 21 Mai 2015

Devant le jury composé de:

Xiaoming Fu	Rapporteur	Professeur, University of Goettingen - Germany
David Larrabeiti	Rapporteur	Professeur, Universidad Carlos III de Madrid - Spain
Pierre Sens	Examinateur	Professeur, UPMC - Paris, France
Danny De Vleeschauwer	Examinateur	Maître de Conférence/Chercheur, UGent/Alcatel-Lucent - Belgium
Nishanth Sastry	Examinateur	Professeur, King's College London - UK
Jean-Michel Portugal	Examinateur	Directeur Terrain de Conquête, Orange Labs - France
Leandros A. Maglaras	Invité	Maître de Conférence, De Montfort University - UK
Noël Crespi	Directeur de Thèse	Professeur, Télécom SudParis - France
Ángel Cuevas	Co-Encadrant	Maître de Conférence, Universidad Carlos III de Madrid - Spain

Thèse numéro : 2015TELE0013

Declaration

I, Xiao Han, hereby declare that this dissertation presents the results of my original research. I have not copied from any others' work or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Abstract

Online Social Networks (OSNs) (e.g., Facebook, Twitter and LinkedIn) have gained overwhelming popularity and accumulated massive digital data about human society. These massive data, representing individuals' personal and social information, provide us with unprecedented opportunities to study, analyze and model the complex network structure, human connections, people similarity, etc. Meanwhile, OSNs have triggered a large number of profitable applications and services which seek to maintain vibrate connections and advance users' experience. In this context, how to devise such applications and services, especially how to extract and exploit effective social features from the massive available data to enhance the applications and services, has received much attention.

This dissertation, aiming to enhance the social applications and services, investigates three critical and practical issues in OSNs: (1) *How can we explore potential friends for a user to establish and enlarge her social connections?* (2) *How can we discover interesting content for a user to satisfy her personal tastes?* (3) *How can we inform a user the exposure risk of her private information to preserve her privacy?*

Drawing on the insights about people's ***similarity*** in social science, this dissertation studies the effects and applications of user similarity in OSNs to address the aforementioned issues. Specifically, sociologists suggest that similarity breeds connection and induces homophily principle that similar people (e.g., similar age, education, or occupation) are more likely to contact, trust, and share information with each other than dissimilar ones. Inspired by these outcomes, this dissertation studies the widespread similarity principle in OSN in terms of whether similar users would be close in their social relationships, similar in their interests, or approximate in their geo-distance, relying on 500K user profiles collected from Facebook; it further explores solutions to effectively leverage the observed similarity principle to devise the following four social applications and services:

- **Effects of User Similarity on Link Prediction for New Users**: we investigate link prediction for new users who have not created any link. Based on the limited information obtained during new users' sign-up procedure, along with the attributes and links from existing users in an OSN, we extensively study how similarity between two users would affect the probability that they befriend. Accordingly, we propose an effective link prediction model for the new users.

- **Mining User Similarity for Content Discovery in Social P2P Network**: we investigate how similarity and knowledge of participants in OSNs could benefit their content discovery in P2P networks. We build a social P2P network model where each peer assigns more weight to her friends in OSNs who have higher similarity and more knowledge. Using random walk with restart method, we present a novel content discovery algorithm on top of the proposed social P2P network model.

- **Inspecting Interest Similarity - Prediction and Application**: we present comprehensive empirical studies on interest similarity and reveal that people are likely to exhibit similar tastes if they have similar demographic information (e.g., age, location), or if they are friends. Accordingly, given a new user whose interests are unknown, we provide a prediction model to identify some individuals who may have similar interests with her. We also illustrate a use case of recommendation system to show the practical use of the proposed prediction model.

- **Information Relevance and Leakage - Location Modeling and Privacy Preserving**: with a representative privacy-sensitive attribute of 'current city' in Facebook, we study the exposure

risk of a user's private information according to her self-exposed information. To this end, we firstly design a current city prediction approach by considering the relevant information (e.g., workplace) and location similarity between friends. We further analyze the prediction results and identify some measurable characteristics from users' self-exposed information, which can significantly affect the exposure probability of private current city. Eventually, taking into account these measurable characteristics, we construct an exposure estimator to assess the current city exposure risk level for an individual user.

Finally, we summarize the significant effects of user similarity in the social applications and services and discuss some promising research directions for the future work, including fusing data from multi-platforms, scaling out the proposed approaches and extending similarity effects into other applications.

Acknowledgments

This dissertation is the result of many experiences I have encountered at TSP from dozens of remarkable individuals who I also wish to acknowledge.

First and foremost I wish to thank my advisor professor *Noel Crespi*. Thanks to him for giving me the opportunity to start and enjoy this research journey. He helped me come up with the dissertation topic and offered me continuous support, guidance, encouragement. He has always been so generous and patient; I remember he used to say 'it does take time to accomplish excellent work, then take your time' to encourage me to face difficulties and keep exploration. He inspired me and led me to a better researcher. I would also like to thank Dr. *Angel Cuevas*. He has been nurturing and advising me throughout my study. His support and guidance have been the fundamental to shape my research and focus my efforts. He has supported me not only by providing a research assistantship over four years, but also academically and emotionally through the rough road to finish this dissertation. Thanks to the China Scholarship Council (CSC), for the financial support of my Ph.D. study.

Secondly, I am grateful to dozens of people who have helped and taught me immensely at TSP. I sincerely thank my talented colleagues who work/ed with and befriend me: *Amir Mohammadinejad, Bahram Alinia, Dr. Cuiting Huang, Dina Hussein, Ehsan Ahvar, Imran Khan, Dr. Rebecca Copeland, Reza Farahbakhsh, Samin Mohamadi, Shohreh Ahvar, Son Han, Dr. Soochang Park, Wipada Chanthaweethip, Dr. Yuanfang Chen*. They are so amazing persons in too many ways. I want to present my gratitude to professor *Jiangtao Wen* from Tsinghua University; I had met him at TSP and he offered me great opportunity for visiting his lab. He is such a gentle man full of wisdom and sense of humor. Studying, discussing and collaborating with him was one of the best experience I have ever had during the Ph.D. I appreciate very much Dr. *Xiaodi Huang* from Charles Sturt University. He has visited our lab for 3 months and provided me many helpful advices at that time and ever since.

A special acknowledgment goes to all my dearest friends Dr. *Chao Chen*, *Leye Wang*, Dr. *Mingyue Qi* and Dr. *Zhuowei Chen*. I had never thought I was lucky enough to meet you here in Paris, in Evry. It was you making my suffering Ph.D. journey amusing and energetic; and it was still you sticking together with me in my bad mood days. I will never forget the days we traveled, made dumplings, played poker until mid-night together. I would like to thank my old buddies who have been always supportive in every way although they are thousands of miles away. They are *Dingli Yan*, *Li Tan*, and *Wei Quan*.

I finish with my family where the most basic source of my life energy resides. I would like to express my deepest gratitude to my parents *Changwen Han* and *Jinfang Xiao*, aunt *Qiaoyun Xiao*, uncle *Guoxiang Xiao* and aunt *Meizhen Sun*, my little brother *Yi Han*, and ..., for their unconditional support and endless love for all these years. I would never accomplish this journey without them.

Publications

The following published, in press or submitted papers are partial outputs during my Ph.D. studies in Telecom SudParis and UPMC.

Journal Articles

- **Xiao Han**, Leye Wang, Noel Crespi, Soochang Park, Angel Cuevas. *Alike People, Alike Interests? Inferring Interest Similarity in Online Social Networks*. Decision Support Systems, vol. 69, pp. 92-106, 2015 (SCI, IF: 2.036)
- **Xiao Han**, Angel Cuevas, Noel Crespi, Ruben Cuevas, Xiaodi Huang. *On Exploiting Social Relationship and Personal Background for Content Discovery in P2P Networks*. Future Generation Computer Systems, Vol. 40, pp. 17-29, 2014 (SCI, IF: 2.639)
- **Xiao Han**, Leye Wang, Jiangtao Wen, Angel Cuevas, Chao Chen, Noel Crespi. *Is Your Hidden Location Undercover? Predicting Current City from Profile and Social Relationship*. ACM Transactions on Intelligent Systems and Technology. (SCI, IF: 9.39, **Under Review**)
- **Xiao Han**, Reza Farahbakhsh, Angel Cuevas, Ruben Cuevas, Noel Crespi. *Community Similarity Degree: Community Selection for Community Recommendation in Online Social Network*. Expert Systems with Applications. (SCI, IF: 1.965, **Under Review**)
- Reza Farahbakhsh, Angel Cuevas, Antonio M. Ortiz, **Xiao Han**, Noel Crespi. *How Far Is Facebook from Me? Facebook Network Infrastructure Analysis*. IEEE Communications Magazine. (SCI, IF: 4.46, **Under Review**)

Conference Papers

- **Xiao Han**, Leye Wang, Son Han, Chao Chen, Noel Crespi, Reza Farahbakhsh. *Link Prediction for New Users in Social Networks*. Proceedings of the IEEE International Conference on Communications (ICC), Jun. 2015, London, UK (Accepted)
- **Xiao Han**, Leye Wang, Soochang Park, Angel Cuevas, Noel Crespi. *Alike People, Alike Interests? A Large-scale Study on Interest Similarity in Social Networks*. Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Aug. 2014, Beijing, China, pp. 491-496
- Wipada Chanthaweethip, **Xiao Han**, Noel Crespi, Yuanfang Chen, Reza Farahbakhsh, Angel Cuevas. *"Current City" Prediction for Coarse Location based Applications on Facebook*. Proceedings of the IEEE International Conference on Global Communications Conference (GLOBECOM), May. 2013, Atlanta, US, pp. 3188-3193
- Dina Hussein, Son N Han, **Xiao Han**, Gyu Myoung Lee, Noel Crespi. *A Framework for Social Device Networking*. Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), May. 2013, Cambridge, Massachusetts, US, pp.356-360

- Reza Farahbakhsh, **Xiao Han**, Angel Cuevas, Noel Crespi. *Analysis of Publicly Disclosed Information in Facebook Profiles*. Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Aug. 2013, Niagara Falls, Canada, pp. 699-705

- Cuiting Huang, **Xiao Han**, Xiaodi Huang, Noel Crespi. *A Triplex-layer Based P2P Service Exposure Model in Convergent Environment*. In UP-TO-US'12 Workshop: User-Centric Personalized TV ubiquitOus and secUre Services. Jul. 2012, Berlin, Germany, pp. 306-314

Contents

Chapter 1

Introduction

Contents

1.1 Background

Online Social Networks (OSNs) (e.g., Facebook, Twitter and LinkedIn), which build up platforms to connect people and maintain social relationships, have attracted a huge number of people over the last decade. In OSNs, participants, rather than browse web sites in legacy web systems, are able to participate in many ways, such as publishing their profiles, making friends, producing various content (photos, answers/questions, videos, etc.) and interacting with each other by a variety of actions such as comment, post, like, sharing and so on. By satisfying people's needs both in communication and sharing information, OSNs have set root in many people's daily life. It is reported that over 1.2 billions monthly participants are active on Facebook to connect people, acquire knowledge, and announce news; 48% of them check 'what's happened' around their social circles everyday. At the same time, relying on OSNs, social media businesses that benefit from this upsurge of OSN popularity and generated tens of billions of dollars every year. In a nutshell, OSNs are playing an important role influencing people's daily life, as well as a significant role in the economy.

From the perspective of technology and science, OSNs provide us with unprecedented opportunities to address many long-standing scientific and technical questions. A number of interesting research challenges emerge in investigating OSNs, regarding efficiently maintained networks, well organized activities, and effectively designed future social applications and services with friendly user-experience. Therefore, in this dissertation, this dissertation is motivated to harness the opportunities of Big Data in OSNs and overcome challenges by enhancing applications and services in OSNs.

1.1.1 Opportunities

Thousands of millions daily activities by participants in various OSN platforms generate massive digital data that traces the behavior of human society.In the first place, as these data and traces are spontaneously produced by individuals, they can be used to capture and represent the individuals' background information, social relationship, personal preferences, even political opinions. Thus, instead of laborious data collection methods, such as surveys or face-to-face interviews, OSNs may become a compelling alternative that collects data samples for traditional social scientific research more effectively and cheaply, thus overcoming challenges for obtaining the inadequate or expensive data samples. Specifically, with the knowledge from OSNs, we can study, analyze and model the complex dynamic network structure, human mobility patterns, human communities, individual influence, information diffusion, etc.

Additionally, OSNs also bring us good opportunity to develop profitable social-featured applications and services. For instance, leveraging location information and relationship in OSNs, a location based social recommendation service can be created to suggest a user with nearby restaurants that are praised by her friends. This practical service not only benefits users with convenience but also earn considerable revenue for the service and OSNs providers. To date, there are 7 millions such applications and web sites integrated with Facebook. Moreover, collective knowledge about tight connections among users and their personal preferences, as well as location information, can potentially contribute to other large scale systems such as content delivery networks, Peer-to-Peer networks and cloud computing.

1.1.2 Challenges

Opportunities usually come along with challenges together. In the process of understanding OSNs and devising social related applications and services, we may encounter various challenging but interesting issues.

Collecting and processing large-scaled data. Given an OSN platform, the number of existing users and various digital data the users contribute is prohibitively large; whereas, the current largest available dataset presented in academic studies covers only a small portion of it. Gathering data in large scale is a practical problem. It is almost impossible to capture a complete OSN graph along with all the uploaded information, especially for a popular OSN with a huge number of active users such as Facebook, Twitter or Foursquare; while data sampling techniques are usually employed to collect a representative small fraction of the data for research purpose. The challenge here is how to ensure the representativeness of the sampled data. Additionally, some of users' information is not public available due to users' privacy concerns and business secret issues. Once a large amount of raw data is available, how to design effective techniques for cleaning, processing, computing and mining massive data is another challenge that restricts the scale of OSNs analysis.

Maintaining vibrant social connections and activities. Organizing OSNs and maintaining users' vibrant connections and activities become a challenge for OSNs, as the population grows and functionality expands. In particular, it is still challenging for OSNs to predict and recommend interesting friends to the participants and help them to attain a successful and active social circle. Especially, recommending appropriate friends for the newly joined participants is even more important to attain their interests in OSNs at the beginning. In addition, satisfying users' requirements in gaining and sharing information is also required to maintain users' vibrant activities in OSNs. Hence, identifying and delivering the personalized content to enormous number of users according to their personal interests and social circle is a compelling and non-trivial task.

Advancing applications and services with social information. In order to develop social-featured applications or advance the existing services, issues regarding data fusion have been addressed. This includes filtering out of 'white noise' data; importing useful social information into the applications and services; and re-modeling the strength of social links to create a new social graph under the certain circumstances. For instance, for a video P2P sharing network, more importance needs to be attached to the information about users' social relationships, interests (movies, music, drama, etc.) statements and interests groups; and perhaps further more weight should be given to the relationship between friends according to their interest similarity. Additionally, more sophisticated approaches are required to provide more convenient and friendly services while involving social information. Take social features for restaurant recommendation service for instance, beyond the user's current location and beyond general popularity of restaurants, the social relationships and the similarity in taste between the user and her friends should be considered to satisfy users with convenient and personalized services.

Estimating and preserving user privacy. Last but not least, while people share knowledge and publish personal information in OSNs, privacy concerns come with the increase of information leakage which may cause potential nuisance such as advertising spam, online stalking and identity theft. It is crucial for OSNs to create a reliable and secure environment for the users. Specifically, whether the users' hidden information are real secure, how to estimate the security of users' private information and how to help users to preserve their private information are all technical issues to be addressed.

1.2 Motivations and Goals

This dissertation aims to enhance the social applications and services regarding the aforementioned challenges, thereby increasing both user experiences and commercial profits in OSNs. It concentrates on three most classical, critical and practical applications: (1) ***Link prediction*** *— how can we explore potential friends for a user to help her establish and enlarge social connections?* (2) ***Interest exploration*** *— how can we discover interesting content for a user to satisfy her personal tastes?* (3) ***Privacy preservation*** *— how can we inform a user the exposure risk of her private information so as to preserve her privacy?*

In order to tackle these problems, this dissertation draws to the insights from social science since people are the kernel of OSNs. Specifically, it is based on a widely observed measure of social relationships and activities in people's daily life, namely ***similarity***.

In social science, McPherson et al. [1] attach much importance to similarity and argue that similarity breeds connections and structures network ties. They also state that similarity induces the significant homophily and correlation principle in social network on a broad range of dimensions. Generally speaking, this homophily and correlation principle shows that similar people contact each other with a higher rate than dissimilar ones. This principle can also be extended to a pervasive context that cultural, behavioral, genetic or material information flows through networks will tend to be localized in social space [1].

With this knowledge, this dissertation intends to achieve two specific goals:

- Empirically study and analyze the widespread presentations of similarity in OSNs, in order to rich the collective knowledge about OSNs. In particular, relying on real data collected from OSNs, a comprehensive study is conducted on whether similar users would be similar in their interests, approximate in their geo-distance, or close in their social relationships.

- Investigate how to effectively leverage the observed similarity related principle to devise and enhance social applications and services, so as to enhance the user experiences for the OSN participants and profits for the service providers. While concentrating on the classical applications, this dissertation investigates four research issues which are distinguished from the state-of-the-arts:

  – This dissertation is concerned with friend recommendation for a *new user* who has not created any links yet, instead of predicting new links for the *existing user* based on the user's existing links; then, it investigates how to identify friends who are very similar to the new user by extensively exploiting the very limited register information of the new user;

  – In order to satisfy users' personal interests, this dissertation also looks at how to introduce social information and similarity properties to improve content discovery in P2P network.

  – For a user whose interests are unknown, this dissertation tends to find out which of the interest-available users are probably similar to the user; the results can be used to support interest-based applications such as recommendation system;

  – In the existing work, a user's private information can be inferred due to the relevance between various pieces of information and the information similarity between friends. Concerning users' privacy, this dissertation estimates the exposure risk of a user' private information and help the user to preserve its privacy.

Notions of Similarity: As the approaches to estimate similarity between two objects are widely defined and applied in literature, before discussing the specific work and contributions in this dissertation, the definitions are given and notions of two users' similarity in this dissertation which primarily depends on two typical similarity notions. First, similarity of two users is defined based on the notion of ***commonalities and differences***. The similarity is higher if users exhibit more commonalities and fewer differences. Applying this notion for similarity estimation, the overlap of users' demographic information is calculated, as well as the common friends or shared interests, using various computation approaches (e.g., cosine similarity). Second, the ***distance*** between two users is considered as an alternative notion to describe users' similarity. In literature, Euclidean Distance, Manhattan Distance, Hamming Distance, etc. are employed to compute users' similarity. In this dissertation, two specific distance similarity are employed — ***geographic distance*** to measure users' location similarity and ***link distance*** to assess users' relationship closeness. Especially, two users are regarded more similar in location if they are closer at geographic distance; and if two users are friends connecting to each other, it is assumed that they are similar as the link distance is zero.

1.3 Contributions

In the course of achieving the goals of this dissertation, original contributions have been made in many facets across collecting data, proposing and formulating the specific research problems, conducting problem-driven analysis and devising the approaches and models. In particular, the contributions are listed as follows:

- **Data Collection from Facebook:** In order to carry out the empirical studies on real social network data, and address the proposed research issues, a crawler has been implemented and social information of around 500K user profiles on Facebook was collected from March to

June in 2012. In the data set, each profile contains three aspects of information including demographic information, social relationships and user interests. In particular, demographic information refers to the attributes such as age, gender, hometown and etc., social relationships is represented by friends list, while user interests cover various interests' domains, e.g., music, movies, TV series, and books. Besides, some other necessary information is also captured to complement the research. For instance, location related attribute (e.g., high school, work place, university) is associated with the located city along with the collected latitude and longitude information for all the cities that appear in user profiles. To the best of the author's knowledge, the crawler has collected one of the largest Facebook dataset with comprehensive social information up to date.

- **Effects of User Similarity on Link Prediction for New Users:** Link prediction for new users who have not created any link is a fundamental problem in OSNs. It can be used to recommend friends for new users to start building their social networks. The existing studies use cross-platform approaches to predict a new user's links on a certain OSN by porting her existing links from other OSNs. However, it cannot work when OSNs are not willing to share their data or users do not want to connect different OSN accounts. In this contribution, a single-platform approach is used to carry out the link prediction; it tends to explore the users' profile attributes (e.g., workplace, high school and hometown) which can be easily obtained during the new users' sign up procedure. Based on the limited available information from the new user, along with the attributes and links from existing users, three types of social features are extracted: basic feature, derived feature and latent relation feature. A link prediction model is proposed using these social features based on Support Vector Machines. Finally, using the large Facebook data set, the proposed model is evaluated. The result reveals that the model outperforms the baselines by achieving the AUC value of 0.83. It also demonstrates that each of the proposed social features contribute significantly to the prediction model.

- **Mining User Similarity for Content Discovery in Social P2P Network:** Content discovery is a critical issue in unstructured Peer-to-Peer (P2P) networks as nodes maintain only local network information. However, similarly without global information about human networks, one still can find specific persons via her friends by using social information. Therefore, in this work, the investigated the problem is of how social information (i.e., friends and background information) could benefit content discovery in P2P networks. First extensive studies are carried out on the Facebook data set, which reveal the importance of friendships in discovering users' personal interests. Guided by the observation, a social P2P network model is built to enrich nodes in P2P networks with social information and link nodes via their friendships. Each node extracts two types of social features - *Knowledge* and *Similarity* - and assigns more weight to the friends that have higher similarity and more knowledge. Furthermore, a novel content discovery algorithm is developed which can explore the latent relationships among a node's friends. A node computes stable scores for all its friends regarding their weight and the latent relationships. It then selects the top friends with higher scores to query content. Extensive experiments validate performance of the proposed mechanism. In particular, for personal interests searching, the proposed mechanism can achieve 100% of Search Success Rate by selecting the top 20 friends within two-hop. It also achieves 6.5 Hits on average, which improves 8x the performance of the compared methods.

- **Inspecting Interest Similarity — Prediction and Application:** Understanding how much two individuals are alike in their interests (i.e., *interest similarity*) has become virtually essen-

tial for many applications and services in OSNs. Since users do not always explicitly elaborate their interests in OSNs like Facebook, how to determine users' interest similarity without fully knowing their interests is a practical problem. This work investigates how users' interest similarity relates to various social features (e.g. geographic distance), and how to infer whether the interests of two users are alike or unlike where one of the users' interests are unknown. Relying on the Facebook data set, comprehensive empirical studies are carried out to verify the *homophily* of interest similarity across three interest domains (movies, music and TV shows). The homophily reveals that people tend to exhibit more similar tastes if they have similar demographic information (e.g., age, location), or if they are friends. It also shows that the individuals with higher interest entropy usually share more interests with others. Based on these results, a practical *prediction model* under a real OSN environment is established. For a given user with no interest information, this model can select some individuals who not only exhibit many interests but also probably achieve high interest similarities with the given user. Eventually, a use case is given to demonstrate that the proposed prediction model could facilitate decision-making for OSN applications and services.

- **Information Relevance and Leakage — Location Modeling and Privacy Preserving:** Privacy has become a major concern in OSNs due to the threats such as advertising spam, online stalking and identity theft. To protect privacy, many users hide or do not fill their privacy-sensitive attributes in OSNs. Existing studies try to infer users' hidden attributes through some other information exposed by the users themselves, which implies a potential disclosure of the hidden attributes. However, these studies do not quantify the exposure risk for a user based on her self-exposed information. Thus, an individual user still cannot understand the exposure probability of her privacy-sensitive attributes, let alone take effective countermeasures. This work attempts to study the exposure probability of a user's hidden attributes via her self-exposed information, with a representative privacy-sensitive attribute - *current city* - in Facebook. To this end, a novel current city prediction approach is designed to disclose a user's hidden current city from her self-exposed information. Based on user information crawled from Facebook, it is verified that the proposed prediction approach can predict a user's current city more accurately than state-of-the-art approaches. Furthermore, based on the proposed prediction approach, the exposure probability, which indicates that a user's current city can be correctly predicted via some measurable characteristics of the self-exposed information, is modeled. An exposure estimator is constructed to assess the current city exposure risk for an individual user, given her self-exposed information. Some case studies are illustrated to show how to use our proposed exposure estimator to protect users' privacy.

1.4 Organization

The organization of the remaining of this dissertation is visualized in figure 1.1. Chapter 2 is a survey of the state-of-the-arts from three perspectives including the metrics to quantify user similarity, the empirical observations regarding user similarity and the relevant social applications and services in OSNs. Before introducing the main contributions, Chapter 3 describes the data collection approach, presents the user profile data crawled from Facebook for this dissertation and discusses the limitations of the data set. Then the main works of empirical studying and modeling in OSNs are introduced in detail one by one respectively in Chapter 4, 5, 6 and 7. More specifically, Chapter 4 investigates link prediction problem for new users who have not created any link. It explores a variety of social features from the very limited available information from new users, study the correlations

between these social features and the probability of link creation, and eventually train a link prediction model to recommend friends to new users. Chapter 5 evaluates the effects of user similarity in social P2P network and exploits users' social relationship and personal background to improve its content discovery. Chapter 6 investigates the correlations between interest similarity and a variety of features and proposes an interest similarity prediction model which can select some interest-similar individuals for a user who does not present any interests. In this chapter, the model is further exploited by a new user recommendation system, verifying its advantages. Chapter 7 discusses the leakage problem of private location information due to the information relevance between different attributes. This work constructs an accurate current city prediction model according to users' social relationships and the relevance among different attributes. An exposure estimator is proposed to assess the exposure risk of current city information given users' self-exposed information. Chapter 8 concludes the dissertation and charts the future research directions.

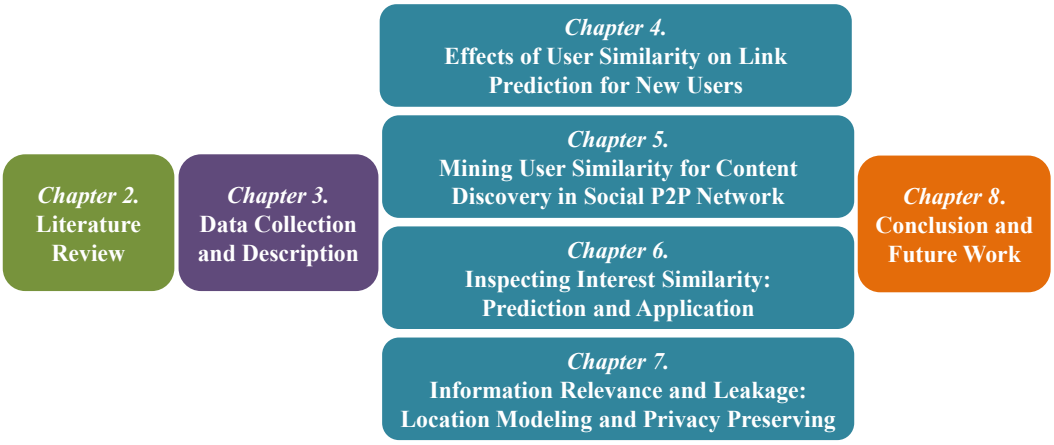


Figure 1.1: Organization of the rest of the dissertation

Chapter 2

Literature Review

Contents

In this chapter, we review state-of-the-art by three aspects. We first investigate the existing metrics and a variety of social features employed to measure user similarity in OSNs. We continue to go over the empirical studies on user similarity and summarize its observations. Lastly, we introduce the user similarity based applications and inspect the effects of user similarity on social applications and services.

2.1 Measuring User Similarity in OSNs

Evaluating similarity is a practical and fundamental problem with a long history which serves to various research domains such as geographic information science [2], biology[3], decision-making [4]. Generally speaking, measuring similarity refers to two factors, namely the measuring approach and variables (i.e., the features are being used to measure similarity). Therefore, in this section, we will first introduce the widely used measuring approaches and then the social features that are employed to measure user similarity.

2.1.1 Similarity Metrics

Concerning about the estimation of similarity in certain applications and services in OSNs, we classify the existing similarity metrics into three categories. We start with the well-known *classical similarity metrics* such as overlap similarity, cosine similarity and Pearson correlation coefficient. Then we review some *information theoretic similarity* and *global structural similarity* metrics which are defined to adapt to specific kinds of applications.

Specifically, we denote the two objects (e.g., users) respectively as u and v, and organize their associated attributes (e.g., friends, demographic information or purchased products) into a set, denoted as $\mathscr{A}_u$ and $\mathscr{A}_v$; $s(u,v)$ represents the similarity between u and v.

2.1.1.1 Classical Similarity Metrics

(1) Overlap Similarity. Overlap similarity is a straightforward estimation of users' commonality by calculating the intersection of two users' attribute sets or the number of two users' shared items, denoted as:

$$s_O(u,v) = |\mathscr{A}_u \bigcap \mathscr{A}_v| \tag{2.1}$$

Since overlap similarity is a simply defined and calculated metric, it is one of the most common used metrics of similarity over studies on OSNs. To predict the potential friends that a user (u) may link to in the future on basis of users' interest similarity, the number of shared information items, authors and tags are counted to measure the item-based, metadata-based and tag-based interest similarity respectively [5]. Newman [6] employs the number of common collaborators between two authors to measure their scientific collaboration and point out that the probability of scientists collaborating increases with the number of common collaborators. Besides, Kossinets et al. [7] demonstrate that the links are more likely to be established between two users if they possess more number of interaction contexts.

Nevertheless, overlap similarity has its limitation on similarity estimation. For instance, assume that users u_1 and v_1 both with 10 friends share 9 of them, while users u_2 and v_2 who both own 30 friends also have 9 friends in common. It is obvious that the similarity between u_1 and v_1 is larger than it between u_2 and v_2; while overlap regards them as identical in similarity. Note that, the value of overlap similarity is not constrained within a certain range.

(2) Cosine Similarity. Cosine similarity, which is defined on two vectors of attributes, calculates the cosine value of the angle (θ) between the two vectors. It can be represented by the dot product and magnitude product of the two vectors, as:

$$s_{cos}(u,v) = \cos(\theta) = \frac{\mathscr{A}_u \cdot \mathscr{A}_v}{||\mathscr{A}_u|| ||\mathscr{A}_v||} \tag{2.2}$$

Cosine similarity can easily be well introduced to a variety of applications. Ye et al. [8] apply cosine similarity to measure the weights of friends according to their distance for location recommendation. In a social tagging system, tag similarity is calculated by using the cosine similarity between two tag vectors [9]. Cosine similarity also plays a fundamental role in computing topic similarity between two users so as to model latent user taste for social link prediction [10].

When the attribute value in the vectors is either 1 or 0, we call it ***binary cosine similarity***. In other words, binary cosine similarity is suitable for the context with binary attributes, like a user's tags vector where each element corresponding to a tag represents whether or not the user annotated questions/blogs with the tag, or like a user's friends vector indicating the relationship (befriend/unfriend) between the user and the others. To some extent, binary cosine similarity can also be regarded as a particular type of normalized overlap similarity. Therefore, it can overcome the above-mentioned limitation of overlap similarity and thus be widely used. By adopting binary cosine similarity, Ashton et al. [11] measure two users' interest similarity and social similarity and thereby investigate the effects of similarity on the evaluation that one user provides of another.

Besides, a modified cosine similarity with a regulating factor, namely ***weighted cosine similarity***, is given to consider more variations. It is defined as:

$$s_{w\_cos}(u,v) = \frac{\mathscr{A}_u \cdot \mathscr{A}_v}{\|\mathscr{A}_u\|\|\mathscr{A}_v\|} \cdot w_{uv} \tag{2.3}$$

Here w_{uv} represents the adjusting variations endowed with diverse indications with respect to context. In a web recommendation system, Amatriain et al. [12] divide the number of two users' co-rated items by the total number of items they rate as the weight. Similarly, to recommend social events with holding a user's home location, the location similarity is calculated by weighted cosine similarity taking into account the common events that users from both locations have attended [13]. Another recent work tends to reduce the cosine similarity value if two users have many dissimilar attributes [14]. Since a weight is used to adjust the estimation of similarity, weighted cosine similarity is expected to be more accurate in specific applications.

As cosine similarity measures the cosine value of the two vectors' angle, it fails to estimate their dissimilarity [15], if the two vectors are in the same direction but different value. For instance, given five items in a recommendation system, user u dislikes any of them and rates all of them with one star; while user v rates them all as five stars. The cosine similarity of u and v is equal to 1.

(3) Pearson Correlation Coefficient. Pearson correlation coefficient measures how strong two vectors are related to each other. It can be derived from the quotient of the two vectors' covariance to the product of their deviations.

$$s_p(u,v) = \frac{\sum_{i=0}^{|\mathscr{A}|}(a_{u_i} - \overline{a}_i) \cdot (a_{v_i} - \overline{a}_i)}{\sqrt{\sum_{i=0}^{|\mathscr{A}|}(a_{u_i} - \overline{a}_u)^2 \cdot \sum_{i=0}^{|\mathscr{A}|}(a_{v_i} - \overline{a}_v)^2}} \tag{2.4}$$

Pearson correlation coefficient is rather popular in collaborative filtering recommendation systems as it takes into account the variety of individual users' review. Specifically, assume two users u and v rate a group of items. It is possible that u normally rates them with high score (4 stars if not dislike) while v rates low ($2-3$ stars in the same case). Pearson correlation coefficient subtracts the average rating score from each rating, thereby eliminates the individual subjective differences [15][16][17]. In addition, Pearson correlation coefficient gives a value ranging from -1 to 1, thus it can represent a negative correlation when below 0. For instance, Ziegler et al. [18] use the negative Pearson correlation coefficient to represent users' opposite interests.

While by using Pearson correlation coefficient, the users' similarity could be assessed artificially high if the vectors merely contain a very limited number of attributes (e.g., only very few items are rated by both of the users in a recommendation system [16]). This issue can be mitigated by using a weighted Pearson correlation coefficient, which will decrease the similarity value by multiplying a weight if the number of co-rated items is less than a threshold [16].

2.1.1.2 Information Theoretic Similarity Metrics

Semantic objects, such as comments, posts, answers to questions, descriptions or reviews about services/products, and tags to photos, videos, music, etc., are widespread over OSNs nowadays. Estimating two users' similarity by such semantic objects is a fundamental task, which can in turn support to a great number of applications (e.g., recommendation system, search, link prediction) [19]. For the sake of commonality and simplicity, the classical similarity metrics are sometimes used to compute the similarity between two semantic objects [20][21]; however, these classical metrics are only limited to similarity assessment of two numerical attribute vectors [22]. Moreover, the structural information between semantic objects may be missing when using classical metrics [21]. Therefore, in this section, we introduce information theoretic similarity metrics which are more commonly used than classical metrics in semantic similarity evaluation.

(1) Mutual Information (MI). From the information theoretic perspective, MI [23] evaluates the information that two objects u and v share. It can be defined as:

$$s_{MI}(u,v) = \sum_{a_u \in \mathscr{A}_u} \sum_{a_v \in \mathscr{A}_v} p(a_u, a_v) \log \frac{p(a_u, a_v)}{p(a_u)p(a_v)} \tag{2.5}$$

where $p(a_u, a_v)$ is the joint probability of a_u and a_v, and $p(a_u)$ and $p(a_v)$ are independent probability.

It has been proved that MI can estimate social tag similarity with higher accuracy compared to Overlap, Jaccard, Dice, and Cosine similarity [24]. By measuring the semantic similarity, mutual information is extended to other specific practical environment. In a study on location sharing services, the relation between user-generated terms and radius of gyration, as well as the relation between users' sentiment and their locations, is evaluated by MI. In order to observe the different patterns of language uses by users' classes (e.g., women/men), Caverlee et al. [25] exploit MI to measure the relatedness between a term and a user class. In addition, MI is also commonly used to test community detection algorithms [26][27]. Similar to the idea of community detection testing, Viswanath et al. [28] use MI to compare the nodes ranking results of four social network based Sybil defense schemes. Though MI is an effective and competitive similarity metric, it is too expensive in its computation [24].

(2) Information Content (IC). Resnik [29], borrowing information content idea, measures two concepts' semantic similarity by the information content of the concepts that subsume them in the taxonomy. The author further extends the measurement to word/tag similarity as:

$$\begin{aligned} s_{IC}(a_u, a_v) &= \max_{a \in \mathscr{S}(a_u, a_v)} [-\log p(a)] \\ s_{IC}(u,v) &= \max_{a_u \in A_u, a_v \in A_v} [s_{IC}(a_u, a_v)] \end{aligned} \tag{2.6}$$

where $s_{IC}(a_u, a_v)$ represents the similarity between two concepts a_u and a_v; $\mathscr{S}(a_u, a_v)$ is the set of subsumed concepts under a_u and a_v; $s_{IC}(u,v)$ calculates the similarity between words/tags u and v; A_u and A_v stands for the set of concepts associated to u and v respectively; and $p(a)$ defines the probability of concept a.

Accurate evaluation of concept probability is a fundamental and challenging problem in the application of IC metrics. Resnik [29] simply divides the frequency of words subsumed under

concept a by total amount of words in textual corpora as the concept probability. Although Resnik's IC metric may be challenged by many issues such as the availability of corpora or the need of manual pre-processing of text, it creates the basic formula of IC based evaluation for a number of more complex and accurate IC assessment [30][31] and it is commonly used in OSNs. Resnik's IC method is used to compute the similarity between either user-generated tags [32], or items and users' profiles [33] so as to achieve personalized recommendation. In order to tag a place, IC method is employed to assist matching a place to its Wikipage and obtaining the description of the place [34].

(3) Lin's Descriptive Similarity. Lin [22] proposes a definition of similarity based on a set of explicit assumptions about similarity in information-theoretic term. Specifically, Lin's similarity theorem defines similarity as the ratio between the amount of information in two objects' commonality and the amount of information to fully describe the two objects:

$$s_{Lin}(u,v) = \frac{\log P(common(\mathscr{A}_u, \mathscr{A}_v))}{\log P(description(\mathscr{A}_u, \mathscr{A}_v))} \tag{2.7}$$

Lin's descriptive similarity can be applied in a variety of domains as long as we know both the commonality and description of the two objects. GiveALink uses Lin's descriptive similarity to calculate users' similarity based on their resources [35]. A social semantic political web application (i.e., Ontopolis.net) [36] applies Lin's metric to compute similarity of pair-wise tags so as to identify similar issues/plans/users. Yi [37] addresses social tag sense disambiguation task with stable performance by leveraging Lin's method to estimate the similarity between a tag and its co-occurring tag. Nevertheless, Lin's similarity is not applicable to non-hierarchical taxonomies [38]. Maguitman et al. [39] extend Lin's descriptive similarity to both hierarchical and non-hierarchical topical ontology.

(4) Maximum Information Path (MIP). MIP [19] advances both the traditional shortest-path based similarity and Lin's descriptive similarity. On one hand, it considers Shannon's information amount of shared content compared to the traditional shortest-path based similarity; On the other hand, MIP overcomes Lin's limitation on non-hierarchical annotations by employing the maximum information path passing through the most specific shared tag among numerous paths between two objects. MIP is mathematically defined as:

$$s_{MIP}(u,v) = \frac{2\log(min_{a\in\mathscr{A}_u\bigcap\mathscr{A}_v}[P(a)])}{\log(min_{a\in\mathscr{A}_u}[P(a)]) + \log(min_{a\in\mathscr{A}_v}[P(a)])} \tag{2.8}$$

MIP is thus welcomed to the non-hierarchical semantic environments. Without hierarchical taxonomy, MIP is leveraged to measure the the similarity between users' interests and candidate posts in an effect study of topical interests on user behavior on Twitter [40]. In the design of social tagging games — *GiveALinks Slider* and *Great Minds Think Alike* — where users are requested to tag the current page and link it to a target page, the authors employ MIP to obtain the target's similar pages to assist players [41]. In folksonomies where annotations are represented by (*user, item, tag*) triples, MIP are applied to compute two users' similarity with respect to their social tags and thus to predict users' social links according to their topical similarity [38]. Aiello et al. [38] indicate that the computation of MIP is not as expensive as mutual information.

2.1.1.3 Global Structural Similarity Metrics

So far, we have introduced similarity metrics that only consider two users' commonality or differences in terms of their own local information. In this section, we are going to review another type of similarity metrics taking into account the global topology information based on structural networks.

(1) Katz. Katz metric [42] is one of the fundamental path-ensemble similarity metrics. Provided that two nodes (u and v) are more similar if there exist more and shorter paths between them, Katz sums the number of paths from u to v and exponentially damp the paths' weights according to their lengths (i.e., the shorter the path, the larger the weight). We use $paths_{u,v}^{<l>}$ to represent the set of length-l paths from u to v, then we can formulate Katz metric as:

$$s_{Katz}(u,v) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{u,v}^{<l>}| \tag{2.9}$$

where β is a very small variable larger than zero. It modulates the contribution of path by assigning a very little value to a long path.

Katz metric is effectively used in link prediction applications, such as link recommendation [43][44] and friend discovery [45]. Specially, to predict the structure of social network without knowing any author-author relationships, Makrehchi [46] constructs auxiliary networks based on author-topic and topic-topic relations, and uses Katz metric to calculate the closeness of either author-topic or topic-topic. In addition, Katz is leveraged to estimate the proximity of a user to a community so as to achieve community recommendation [47][48]. Katz is improved in its recent use to be adapted to the multi-modal network [49] or to integrate topology information with node attributes and time characteristics [50].

(2) Rooted PageRank (RPR). Grounded on the idea that *a web page is significant if it receives numerous links from other important pages* [51], PageRank was devised to measure the significance of pages by randomly walking with a probability of α through outgoing links, and with a probability of $1-\alpha$ to a certain designated page [52]. On the basis of PageRank, Rooted PageRank [53], rooted at a node u, estimates the similarity between u and v by the stationary probability of u walking to v where u takes a probability of $1-\alpha$ move to a random neighbor and a probability of α return to u at each walk step. The stationary probability for all vertex pairs in a graph can be computed by [54]:

$$s_{RPR}(u,v) = (1-\alpha)(I-\alpha D^{-1}M)^{-1} = (1-\alpha)D^{-1/2}(\sum_{k=0}^{\infty}\alpha^k T^k)D^{1/2} \tag{2.10}$$

where M is the adjacent matrix; D is the diagonal degree matrix in which $D[i,i] = \sum_j M[i,j]$; and T is the normalized adjacent matrix ($T = D^{1/2}MD^{-1/2}$).

RPR is directly applied to or modified to a variety of applications. First, RPR is widely leveraged to social link prediction [53][55][56]. For instance, Liu et al. [57] define an AuthorRank to represent the co-authorship in social digital library networks by a weighted RPR. Backstrom et al. [58] calculate PageRank score to predict and recommend social links in a supervised way. RPR is also exploited to evaluate the trust of nodes according to the principle that 'close' users are often more trustworthy [59][60]. Besides, semantic similarity can be gauged by RPR [61]. In a semantic environment of Folksonomies, an adaptive PageRank algorithm is proposed and applied to measure tag similarity [62] and further facilitate recommendation [63].

Note that, besides RPR, there exist several other random walk based similarity measurements including Hitting time [64], Escape Probability [65], HITTS [66], Commute Time [67], PropFlow [68], etc. All these methods share the fundamental idea of graph based random walk but are defined independently to adapt to different applications and circumstances. Besides, these random walk methods attract huge attention on their improvement in scalability [54][69][70] and computation speed [71][72].

(3) SimRank. SimRank [73] regards two object u and v similar if they are related to similar objects. Specifically, it estimates the similarity between u and v by average similarity between the

neighbors of u and v in an iterative manner on a graph, and it defines the similarity of u and v equal to 1 if u and v are the same. SimRank can be mathematically written as:

$$s_{sr}(u,v) = \begin{cases} 1 & \text{if } u = v \\ \frac{C}{|\mathscr{A}(u)||\mathscr{A}(v)|} \sum_{a_u \in \mathscr{A}(u), a_v \in \mathscr{A}(u)} s_{sr}(a_u, a_v) & \text{if } u \neq v \end{cases} \tag{2.11}$$

where $\mathscr{A}(u)$ and $\mathscr{A}(v)$ stand for the neighbors of u and v respectively; and $C \in [0,1]$ is a constant.

SimRank usually yields the best performance in structural context [74]. Bao et al. [75] propose a SocialSimRank which adapts SimRank to compute the similarity between social annotation and web queries in order to facilitate web search. Yu et al. [76] cluster photo-sharing groups into categories by using SimRank to analyze similarity of groups and tags. SimRank is also suitable for friends recommendation [77][78], and trust estimation [79]. However, SimRank cannot satisfy the automorphic equivalence property [80]. In addition, SimRank also suffers from the expensive computation, and thus many fast SimRank algorithms have been proposed [74][81]

2.1.1.4 Summary

We summarize the most typical and widely used similarity metrics into three categories. The classical similarity metrics are almost leveraged to every specific domain of social analysis, models and applications, whereas they are only limited to similarity assessment of two numerical attribute vectors [22]. Moreover, the structural information may be missing when using classical metrics [21].

In OSNs, information theoretic similarity metrics primarily serve to estimate the similarity between objects based on their semantic information. Compared to the classical similarity metrics, this type of similarity metrics takes more computation time and space. And it may also miss some structural connections among objects.

Global structural similarity metrics extract the structural information implied between two objects and are more beneficial to link prediction, trust estimation and community detection. The biggest issue residing in all global structural metrics is their high computation complexity, though they are usually more accurate. Additionally, this type of metrics is easily to neglect the local network characteristics and other attributes information.

In a nutshell, each type of similarity metrics has its own advantages and disadvantages in specific application domains. We'd better select the appropriate metric according to the application context. An alternative way is to consider the combination of multiple metrics together. For instance, we can combine both local and global similarity measurements for link prediction, by respectively applying cosine similarity to estimate two users' profile similarity and leveraging Katz similarity to capture their global structural relationship closeness.

2.1.2 Features of User Similarity

In different social network platforms (e.g., Facebook, Twitter or Foursquare) or diverse applications such as recommendation, link prediction and location based services, the obtainable, sensitive and effective features are different. In this section, we introduce the varied features with respect to three classes: *attribute features*, *activity features* and *structural features*.

2.1.2.1 Social Attribute Features

Social attribute features refer to all the features derived from users' basic information. Different OSN platforms usually maintain different basic information. For instance, Facebook holds users'

gender, age, education background, current city, interests, contacts, etc., while Twitter only contains a simple user profile with three attributes, i.e., bio, location and website. All the profile information can be explored to investigate users' characteristics in OSNs, such as homophily. Specifically, users' locations are explored to the homophily of location of friends and further to infer some others' locations [82]; recommendation systems take advantage of users' interests and all the available profile information to predict and recommend new items for users [83]. These attribute features are also leveraged into P2P networks to identify the similar peers [84].

The analysis based on social attribute features can effectively help us to understand the character and structure of OSNs, thus further provide a ground to us for modeling and applying OSNs. However, we cannot fully depend on these attribute features. First, it is regarding of data limitation. In some OSNs (e.g., Twitter), the obtainable profile attributes are very limited; even in Facebook which attempts to require more users' basic information, only a fraction of them can be publicly accessed due to the privacy concern. Second, extracting specific features in certain applications is a challenging issue. For instance, to predict users' links, it is rather hard to decide which features (e.g., age, college, work) are conducive and how much of them respectively. Third, the social attribute features are relatively stationary which may not represent users' dynamic characteristics. Fourth, concentrating on social attribute features may lead to the miss of structural features in OSNs.

2.1.2.2 Social Activity Features

Social activity features relate to users' real-time activities, such as comments, posts, shares, likes, forwarding, tagging, communicating, check-ins, purchase, download, rating, question and answer, edit and so on. The dynamic activity information is usually used to analyze the evolution of OSNs [85], or to predict users' mobility traces [86]. The recent purchase, download and rating records can also be leveraged to predict users' favorite products in recommendation systems [83] or to assist social P2P networks to trace the source of content [87]. Research in information retrieval can exploit features like forwarding and shares to investigate the properties of information flows in OSNs [88]. Check-ins are significant information for a variety of location based applications [89].

Social activity features can compensate for social attribute features to some extent. First, they are evolving information and can indicate users' recent behaviors. Second, it is relatively cheaper and easier to obtain. For instance, tweets and check-in data can be accessed through public APIs from Twitter and Foursquare, respectively, for free. However, social activity features cannot overcome the rest shortages of social attribute features, including the miss of structural features in OSNs and the difficulty in extracting appropriate features for certain applications.

2.1.2.3 Social Structural Features

Social structural features are the features defined on the basis of the various explicit and implicit links connecting users in a social graph. *Mutual friends relationship* [6], *follower-followee* [89], or *the member relations in a same group* [55] are the most natural links in OSNs. These natural links imply some other information that can be used in applications. Mutual friends may indicate two users having the same age, living in the same city, preferring similar products, etc. The relation between follower and followee directs the information flow and reflects users' influence.

Besides the explicit links between nodes, there is much auxiliary information including either attribute features or activity features being explored to describe the nodes' implicit connections and assisting to build auxiliary social networks [84]. For example, if a user comments to another one's posts or photos, or two users comment to, like or share the same post, or two users use the same tag, an implicit link between the two users could be set up.

The existing studies deal with the links in OSNs in two ways: *all equal links* [82] or *biased strong/weak links* [58]. All equal links mechanism takes all the links as equal and the social network is regarded as a unweighted graph; Biased strong/weak links mechanism measures users' cohesion and assigns a weight to a link to describe its strength. Users' cohesion is usually measured by the qualities such as the number of communications, or common friends, or common 'likes' between them.

Based on these explicit or implicit, unweighted or weighted links in OSNs, a number of structural metrics, including the number of links, path lengths, local/global cluster coefficient, modularity, number of triads, are employed to enhance social application, especially in community detection, link prediction. In fact, structural features are built upon the other two types of features; therefore, they may inherit their shortages such as the limitation on data availability. While one of the manifest advantages of structural features is that they take the structures of OSNs into accounts.

2.1.2.4 Summary

In a nutshell, a plenty of various social features are leveraged to study and model OSNs and enhance social applications. However, we encounter several issues in the practical research and application with these potential features. First, various platforms maintain diverse information and this information may be not accessible; therefore, we can only select the obtainable information in certain cases. Second, among the available information, it is still challenging to determine the effective ones with the corresponding weights during similarity estimation.

2.2 Empirical Observations of User Similarity

In recent years, extensive empirical studies have been carried on over various OSN platforms. Mislove et al. [90] look into the structural properties of four online social networks of Flickr, YouTube, LiveJournal and Orkut. Through all these four OSNs, the authors confirm the properties of the power-law, small-world and scale-free that are discovered in real social networks. The structure of Facebook graph is anatomized in [91], which presents its fully connection, dense structure, and assortative pattern. These structural analysis shed light on the OSNs grounded on the comprehensive graph features in one or several OSNs.

Beyond the structural characteristics which are captured from the graph of the users (nodes) and their social connections (edges), we attach more significance to the attributes of the users themselves, the representative meaning of the social connections, and the potential relations in users' attributes which may cause, or indicate, or correlate to their social connections. Because these studies not only lead us to the essential reasons of social connections among people but also stir us to more appropriately take advantage of social relations to improve the online applications.

Particularly, we touch on the effects of users' similarity in terms of similarity in demographic background, interests or behaviors on their 'connections'. Note that the 'connection' here is a general representative of relation between users but not necessarily a real link as friendship. For example, in the study [92] where Leskovec et al. point out that people are more likely to communicate to the others in similar age, language and location, the connection represents the probability of communication between users. In other words, we overview the widespread homophily phenomenon in OSNs where similar users tend to connect to each other. In particular, we verify three specific types of connections in terms of interest similarity, location relevance and friendship.

2.2.1 Effects of User Similarity on Interest Similarity

The studies on interest similarity tend to find out powerful implications that may indicate the degree of interests sharing between users. Users with the similar interests are quite conducive information in social applications and services. For instance, users of similar interests are usually captured to assist recommendation for electronic business and content discovery in P2P networks. However, due to the unavailability or inadequacy of users' explicit interests in some context, identifying the effective implications of interest similarity is required.

Many existing studies, depending on common senses, turn to the social relationships and examine whether or not two socially linked users would share more interests than the unlinked users. It is revealed that friends indeed exhibit more common interests than strangers; it is also demonstrated that the interest similarity relates to the friends' distance [5][93]. Membership also holds the tendency of sharing interests. Users in a same forum are observed being more likely to attend the same threads [94].

In addition, the effects of similar background information, including gender, location, age, occupation, etc., is comprehensively examined. It is revealed that users' with more similar background information are also more alike in their interests [95]. For instance, users with less age difference or in the same generation present higher interest similarity; users' nationality or their geographic distance manifest their interest similarity to some extent.

Some other users' relations have also been explored to indicate users' interest similarity and leveraged to enhance applications. The correlation between trust and interest similarity is verified in [18]. It is also pointed out that users using the same tags or tagging the same items may share more interests [96]. In P2P networks, two users are considered as similar in interest if they can provide files to each other [97]. Collaborative filtering recommendation systems are based on the principle that two users will have more common interests if they share some interests already [98].

2.2.2 Effects of User Similarity on Location Relevance

Location becomes the most essential information in Location-Based Services. Google map leverages a user's current location to show her surrounding routes; Coupon advertises promotion information according to users' living cities. OSNs with location, i.e., Location-Based Social Networks (LBSNs), combine social-spatial properties together and bring users even more fantastic services. For instance, Foursquare can recommend a nearby restaurant to a user according to her friends' check-ins and ratings. Therefore, it is promising to understand how users' similarity relates to their location relevance and further apply this collective knowledge to enhance the both social and location based services in LBSN.

The correlation between social relationships and geographic distance is a fundamental spatial characteristic of OSNs. Much existing work demonstrates that the probability of two users socially linking to each other is a function of their distance. There is a universal agreement on the function that the probability decreases with the increase of distance, whereas the decay relation is not the same in different contexts, e.g., $p(d) \sim d^{-2}$ in a mobile phone communication network [99] while $p(d) \sim d^{-1}$ in Facebook [82].

On the other hand, the relation between friendship and users' mobility is examined by check-ins in OSNs. Cho et al. [86] reveal that people are more likely to visit a distant place where friends live around, while the social connections less affect people's short-ranged travel. They also point out the limits of explanation on users' mobility from their friends, based on the empirical findings that 84% of the users have less than 20% of the check-ins visited by one of their friends. Moreover, time effect is also reported. The probability of a user traveling to her friends' check-ins is decreasing

by the time window since her friends' visiting [86]. Besides, temporal distance between two users' check-ins at a same time is verified efficient to enhance location prediction [100].

Beyond the effect on location relevance from the direct links between two users, Leskovec et Horvitz [101] systematically report a complex interplay between topological and geographical properties of OSNs. They verify that topologically shortest paths between any of two users in OSNs are proportional to their geographic distance. Besides, Volkovich et al. [102] state that the ties in a tightly linked social group span short distances.

2.2.3 Effects of User Similarity on Links

Social link, connecting users in OSNs, is the most fundamental component. The aforementioned basic structural analysis are conducted based on the users' social links. Nevertheless, in this section, we discuss how users' similarity would affect the creation of a link, instead of inspecting the work on OSNs' structural properties. As social link represents some certain relationship between users, mining the users' similarity will provide a significant clue for their social link. For instance, the classmates who go to the same college may be friends. In the previous two subsections, we review the existing work that evaluates the effects of friendships on interest similarity and location relevance respectively. Then, we survey the effects in the opposite side: user similarity on links in terms of interest and location, and examine some other effect observations subsequently.

In a study for disentangling the puzzle whether people befriend with others who are similar to them, and whether friends get more similar over time, Lewis et al. [103] find that users sharing certain interests in music and movies are significantly likely to make friend with each other. By mining the semantic interest similarity on social tags, the effectiveness of interest similarity on the performance of friendship prediction is also confirmed [38][104].

Although people may make friends in both far and close distances [105], Backstrom et al. [82] visualize that the probability of being friends drops quickly with the increase of geographic distance. Cho et al. [86] demonstrate that similarity of users' check-ins trajectories strongly indicate a link in OSNs. In addition, recommending friends on a basis of user similarity according to their location histories is validated in [106].

Many other user similarity metrics influence the probability of the creation of links. Backstrom et Leskovec [58] reveal that the probability of becoming friends is increasing with the number of mutual friends between two users. Leroy et al. [55] measure the probability of being friends according to the interest groups that users are joining in. They show that two users will befriend if they join many same groups as well as if they join in the groups with little time interval. Besides, co-contact, co-subscription, co-subscribed, are proved to be conducive for link prediction [107].

2.2.4 Summary

In this section, we review how user similarity impacts their interest similarity, location relevance and relationships. We confirm the homophily properties in OSNs and conclude that similar users are more likely to exhibit more common tastes, live closer, and become friends. In fact, similar users also present some other homophilous characters: similar users provide each other with positive evaluation in a high probability [11]; the mutual trust between people who are similar to each other is high [18].

2.3 Improving Social Applications with User Similarity

Inspired by the aforementioned findings about effects of user similarity, researchers extensively exploit these effects to develop and improve a variety of applications in OSNs, and also introduce social information with their effects into existing systems such as Peer-to-Peer (P2P) network and recommendation systems. In this section, we primarily review state-of-the-arts leveraging user similarity with respect to four specific categories of applications and services, including content discovery in Peer-to-Peer network, recommendation systems, location prediction and link prediction.

2.3.1 User Similarity in Peer-to-Peer Network

Identifying the users who have similar interests is a fundamental task for content discovery or prefetching in social P2P network. Numerous exiting studies leverage a variety of techniques to build up peer to peer relationships according to their similarity. We classify them into the following three categories.

A classical category of social P2P searching approaches forms social-like relations based on user similarity learning from historical behaviors. By studying the historical behaviors, Sripanidkulchai et al. [108] identify the similarity principle of interest-based locality: it is more likely to find content on a particular peer if it occurred on the peer in the past. Exploiting users' historical queries, peers connect to others with the same interests gradually by the result of daily searches [87]. In [109], the authors also look into users' friends circles and exploit the link prediction method to extract peers' proximities, consequently enhancing the capacity for resource discovery in P2P circumstances.

The recent surge of the OSNs brings the new trend of leveraging real social information into P2P networks. Li and Shen [84] map Facebook users' information into P2P networks to cluster users by their common interests and organize nodes into a structured graph and perform searching by distributed hash table. Using an existing co-authorship graph, Chirita et al. [110] generate a large P2P collaboration network, investigating diverse search mechanisms and indicating its quality. Apart from improving content discovery, social information is also employed to solve other issues in P2P networks. Sanchez-Artigas and Herrera [111] leverage the implicit trust in social networks to address the churn problem in P2P systems. In [112], the authors reduce startup delays in P2P video sharing networks through a prefetching approach based on users' preferences. [113] accelerates the performance of BitTorrent file sharing with the Twitter social network grounded on the observation that the nodes in Twitter communities are likely to meet each other again.

Different from the previous two categories of approaches that focus on the similarity between two individuals, another category of methods applies various strategies to cluster users who are close to each other in tastes into a community, namely community-based solutions. Fast et al. [114] improve P2P performance by means of clustering users and creating a social network akin to the one based on users' music preferences, with the Hierarchical Dirichlet Processes. In [115], the authors present the self-organized interest-based clusters in affinity networks which are further exploited to devise a proactive P2P recommendation system. [116] proposes an approach to grouping similar nodes and producing a super-peer for constructing Semantic Overlay Networks (SONs). It can achieve high-quality searching by posing similar queries to the N most-similar SONs. Liu et al. [117] introduce a small world architecture for P2P networks and propose a semi-structured algorithm to achieve content discovery in multi-group P2P systems. Generally, as peers in the same community share more attributes and content, organizing a community is an ideal way to accelerate the search process. Meanwhile, detecting and establishing a useful community is not an easy task.

2.3.2 User Similarity in Recommendation Systems

Recommendation system is another typical application of users' interest similarity. In other words, once we can identify a user's personal interests and capture the other ones who have similar interests, we can recommend this user items/products in two basic ways: either selecting the items which are similar to her personal interests (i.e., content-based recommendations); or collecting the items from users who share interests (i.e., collaborative recommendations). The hybrid approaches which combine the content-based and collaborative methods are also popular alternatives. However, these traditional recommendation approaches may encounter severe performance issues due to the sparse interest information, especially for the new users [98].

Based on the observation from OSNs that friends usually share more interests than strangers, social relationships are leveraged to increase recommendation effectiveness. Liu et Lee [83] capture users' preference ratings and friendships from an OSN and enhance recommendation performance by incorporating these social information into collaborative recommendations. By incorporating social friendships information, Ma et al. [118] leverage a matrix factorization framework with social regularization to improve traditional recommendation.

Apart from the friendships, users' social demographic information is also employed to enhance recommendation performance, especially to reduce cold-start problem or recommend for users who do not present any favorite items/products. Specifically, Chen and He [119] integrate demographic attributes including age, occupation and gender to collaborative filtering recommendation. Based on α-community spaces model and 'level of agreement' of the community, Nguyen et al. [120] introduce demographic information to build a rule-based induction for new user recommendation.

2.3.3 User Similarity Based Location Prediction

Users' location becomes a kind of essential information when understanding the spatial structure of OSNs and providing Location-Based Services. By making use of location information, Friend Finder enlarges users' social circle by recommending new friends geographically around them; exploiting the knowledge about users' locations and their social relationship, Foursquare can help users to locate some surrounding restaurant that their friends visited. Moreover, the location information accompanying with users' behaviors, such as resource sharing and content consuming, is helpful to large scale systems such as cloud computing and content delivery network [86]. Although there exist many ways to obtain users' fine or coarse locations, such as GPS, wifi, open access check-ins and predefined location information in users' profiles, in real-life OSNs, only a small fraction of users expose their locations. For instance, 16% of users in Twitter reveal home city [89] and 0.6% of Facebook users publish home address [82]. Thus, in OSNs, a number of existing studies tend to infer missing locations based on the accessible location and users' similarity. We review this branch of research in this section by classifying them into four categories: *relationship-based* prediction, *content-based* prediction, *hybrid content-relationship* prediction and *multi-indication* prediction.

Based on the the observation that the probability of being friends is declining with geographic distance, some approaches infer a user's location according to the visible locations of her friends. We call this type of approaches as *Relationship-based Prediction*. For example, depending on this observation, Backstrom et al. [82] build a maximum-likelihood location prediction model and eventually refine the prediction with an iterative algorithm. However, the pure relationship-based model might miss a lot of useful location-sensitive information. Especially to the users who do not expose their relationships, it is even hard to infer their locations.

The rise of Twitter has spawned a mass of tweets which may contain location-specific data, therefore, another category of prediction approaches [121][122][123] infers a user's location rely-

ing on her location-related tweets. The basic idea of these approaches is that the users who post tweets mentioning same locations have a large chance of being close to each other. Therefore, such approaches detect the location-related tweets and construct a probabilistic model to estimate the distribution of location-related words used in tweets. Nevertheless, capturing location information from their tweets is a very complex task. Twitter messages consist of texts that are unstructured; The users often use shorthand or informal expression on Twitter, and many of the texts have grammatical and spelling errors; The very sparsity of multiple location-sensitive terms in tweets is another obstacle. Moreover, the captured location might exist ambiguity. For example, the location-sensitive terms extracted from a user's tweet content probably bias the prediction, as they could just implicate her interested places which might be far away from where they are staying; Also, polysemous words could lead to incorrect prediction.

An alternative compelling category combines the location indications from relationships and tweet content and builds upon the two users' similarity principles. TweetHood identifies a user's location by exploring both her tweets and her closest friends' locations [124]. Tweecalization improves TweetHood by employing a semi-supervised learning algorithm and introducing a new measurement which combines trustworthiness and the number of common friends to weight friends [125]. Li et al. [126] integrate the location influences captured from both social network and user-centric tweets into a unified discriminative probabilistic model. By considering a user may be related to multiple locations, MLP model [89] proposes to set up a complete 'location profiles' prediction which infers not only a user's home location but also her other related locations.

Besides users' relationships and content, multiple location indications are explored from other possible location resources to infer users' invisible locations. Such multi-indication idea has also been used to Foursquare, which specifically exploits mayorships, tips and dones that users marked [127].

2.3.4 User Similarity for Link Prediction

Predicting links that a user will create in the future is pervasive in OSNs to recommend friends for users and maintain their social activity. Distinguished from the previous link prediction surveys [53][128], from a perspective of the effects of user similarity on link prediction, we categorize the existing related literature with respect to whether the predicted user is new or not (i.e., has links already or not yet).

More work concentrates on the existing users who already exhibited many links; they predict new links primarily relying on the users' closeness through users' established links (i.e., existing friends). Hasan et al. [129] rely on a set of users' proximity features (shortest distance, keyword match count, etc.) extracted from a co-authorship graph to predict the likelihood of the future co-author relation between users. In addition, Menon et al. [130] improve the prediction by using a supervised matrix factorization method to learn several latent features that can represent users' proximity and connect them from the social network graph. Based on supervised random walks, Backstrom et al. [58] combine information from users and links to guide the random walk on the existing network graph to determine the closeness of two users.

Recently, a few studies started to deal with the problem of link prediction for new users who have not linked to anyone. Leroy et al. [55] employ memberships in interest groups as auxiliary information and predict links for new users by exploring user similarity features such as number of common groups, size of common groups and difference in joining time. With multiple sources in social networks, Ge et Zhang [107] predict new users' link based on various similarity auxiliary networks, such as co-contact network, co-subscription network, co-subscribed network and favorite network. Another work addresses the cold start link prediction problem without knowing any user-

to-user links in a platform by using some similarity information outside the platform (e.g., shopping history) [55].

Chapter 3

Data Collection and Description

Contents

3.1 Data Collection

3.1.1 Approaches to Collect Data

As data is the fundamental for studying users' behaviors and analyzing network characters, many approaches have been proposed to collect data. In this section, we are going to briefly introduce four typical data collection methods: conducting survey/interview, capturing network traffic, crawling web page and accessing API interface.

Conducting a survey or interview is the most traditional way to collect data in research. In this way, we collect data by requesting a number of people to complete a group of customized questions. Although we are able to purposely raise questions for the research, the samples normally are confined to a small scale. Additionally, it may cost by recruiting people for survey or interview.

The evolution of Internet leaves us a sheer amount of digital footprint of users' online behaviors. Capturing the network traffics generated by people accessing web sites or communicating via network is one of the widely used methods. A plenty of tools (e.g., Wireshark, Fiddler, and NetworkMiner) can help to freely monitor and capture these network traffics. Nevertheless, these tools

only can capture the network traffic passing through the same router; and the transmitting data are usually encrypted, hence we probably cannot acquire the detailed information such as the content of a web site or a user's background information.

Instead of capturing the network traffic, crawling web pages is an alternative extensively applied method. A crawler, as an agent requesting the server, extracts sensitive information from the replied web page. It will continue to crawl and jump to another associated web page. One competitive advantage of crawler is that it can collect all the detailed information that one can view with a browser. Besides, the crawled data usually keep a topology of the crawled nodes in the network graph, even though the sampled data may bias to the local graph of the first crawled node.

Besides, almost all the social network platforms, such as Facebook, Twitter, and Foursquare, provide public APIs for the third-party or individual to collect data. The API request is relatively simple and the offered data is structured. Unfortunately, the available data via API are limited or access token needed, which may not satisfy researchers' requirements for conducting deep studies and analysis.

3.1.2 Crawling Facebook

Facebook is the most popular social network that has been succeed to attract attention from all over the world, including celebrities, merchants, politicians, artists and demographic researchers. Facebook is also a comprehensive repository which integrates various user information, social applications, user communities, events, etc. Moreover, Facebook concerns about users privacy and provides users with custom-privacy functions, which satisfy our needs in studying privacy issues. Therefore, Facebook is an ideal and representative OSN where we can collect data and conduct studies.

In this dissertation, we use both crawler and API methods to collect our Facebook data set. Particularly, we implement crawler to capture user profiles, including demographic information, social relationships and user interests. We crawl the user profiles by two methods - Breadth First Search (BFS) [131] and random methods. Using BFS, we randomly select some root users, visit the root users and their friends (one-hop friends), and subsequently continue to access the friends of the root users' friends (two-hop friends). We extract user profiles of all the visited users, and construct a structural Facebook data set, namely *Friends Group* data set.

Additionally, we crawled user profiles by the random method to establish an unstructured data set, called *Random Group*. We tend to achieve two goals with this *Random Group*: (1) we compare user profiles' characteristics in both *Friends Group* and *Random Group* to demonstrate the representativeness of the data sets; (2) we construct a baseline with the *Random Group* to compare with for the work of content discovery in social P2P network (Chapter 5).

Besides, we complement some necessary public information through Facebook Graph API[1]. We primarily focus on location information of some location relevant attributes (e.g., high school, college and employer). We collect all the values of location relevant attributes emerged in our user profiles and obtain the city name and latitude/longitude information by querying Facebook Graph API.

Since Facebook allows users to leave empty to any of the information attributes or only to display certain information to confined social circles according to users' customized-privacy, we merely intend to extract public information from users as our data set resources. In addition, we anonymize all the user profiles to conduct the research.

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

3.2 Data Description and Preliminary Visualization

This section will introduce our crawled data set in detail, demonstrate its representativeness for the subsequent researches, and visualize some preliminary characters presented in our data set.

3.2.1 User Profile Description

Each user profile is composed of three parts of information including demographic information, social relationships and user interests. Specifically,

- *Demographic Information*: It refers to seven specific profile attributes[2]: age, gender, current city, hometown, high school, college and employer (i.e., work place). Current city and hometown are two location attributes which are linked to the corresponding latitude/longitude position. High school, college and employer, as location relevant attributes, are associated with a city name and latitude/longitude values.
- *Social Relationships*: We captured users' friend lists, thus here we define social relationship as user-claimed friendship. Note that friendship in Facebook is bidirectional, i.e., A is B's friend if B is a friend of A.
- *User Interests*: Facebook encourages users to explicitly describe their favorite music, movies, TV shows and so on. Nine interest domains are collected in our data set: movies, music, TV shows, books, games, athletes, teams, sports, and activities.

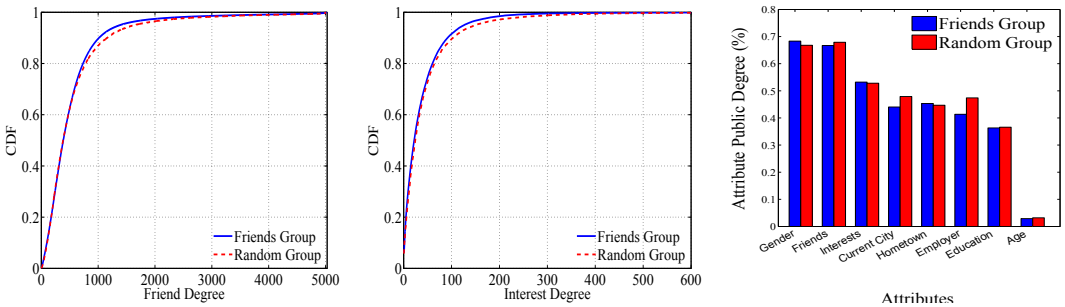
We crawled Facebook from March to June 2012 and collected profile data. Specifically, we collect 479,048 user profiles by BFS method for *Friends Group* while 41,595 profiles for *Random Group* by random method. We mainly rely on *Friends Group* to conduct our analysis and experiments in the specific researches while use *Random Group* to compare with when necessary.

3.2.2 Data Representativeness Evaluation

Since the representativeness of the information in the two groups guarantees the reliability of the following data studies, comparisons and data-based experiments, we first compare several statistics of public social information drawn from two data sets to reveal their consistence and representativeness. We assume that the social information in the two groups are representative if the statistical characteristics in the *Friends Group* approach to the corresponding ones in the *Random Group*. In particular, we tend to compare Friend Degree, Interest Degree and Attribute Public Degree between *Friends Group* and *Random Group*.

A user's Friend Degree is defined as the number of her friends. Figure 3.1(a) plots Cumulative Distribution Function (CDF) of users' Friend Degree for both *Friends Group* (the blue solid line) and *Random Group* (the red dotted line). Note that the data shown in the figure has been excluded the users who have no friends. We observe that most of users maintain a number of friends, in which 95.5% and 96.5% of users have a Friend Degree higher than 50 in *Friends Group* and *Random Group*, respectively. The Friend Degree of around 1% of users even exceeds 4000 in both groups. The median Friend Degree is 387 in *Friends Group* and 384 in *Random Group*, which are very similar.

[2]In this dissertation, profile attribute is different to social feature. Profile attributes are the information that users claim on their Facebook page (e.g., age, hometown, gender); social feature indicates the quantitative values, like age distance, location distance, friend similarity, etc., which are derived from attributes.



(a) CDF of friend degree (b) CDF of interest degree (c) Attribute public degree

Figure 3.1: Preliminary statistics studies

Similarly, a user's Interest Degree is defined as the total number of her interest. Note that in this presentation we sum the number of music, movies, TV shows, books and games as Interest Degree since they are the most typical interest domains. Figure 3.1(b) draws CDF of users' Interest Degree and shows that the users in *Random Group* present slightly higher Interest Degree than do the users in *Friends Group*. The median Interest Degrees is around 24 and 22 respectively in the two groups.

Besides, we consider a user as Public User regarding one attribute if she publish her information of this attribute. For instance, if a user U has two public attributes, named age and gender, we call U is a Public User both regarding age and gender. Accordingly, we define Attribute Public Degree of one attribute in a group as the number of Public User regarding the attribute divided by the total number of users in the group. We use Attribute Public Degree to further reveal the representativeness of the data in the two groups. We compare eight attributes: gender, friends, interest, current city, hometown, employer, education and age. Education here is the combination of high school and college. Figure 3.1(c) shows that the largest difference of Attribute Public Degree between the two groups is approximately 6% in terms of employer. The average Attribute Public Degree difference between the two groups is only about 1.1%.

In summary, the CDF of Friend Degree of the two groups match well with each other in figure 3.1(a); and so does the CDF of Interest Degree in figure 3.1(b). Figure 3.1(c) shows that the Attribute Public Degrees of the eight attributes in *Friends Group* are all very similar to those in *Random Group*. Therefore, we believe that the social information in the two groups are representative and feasible to be used for the studies, comparisons and data-based experiments in this dissertation.

3.2.3 Preliminary Characters Presentation

Before applying the data sets into some specific work, we would like to reveal some high-level characteristics and patterns of demographics that emerge from the collective users. We use *Friends Group* as primary data resource and *Friends Group* is indicated in the rest of this dissertation if there is no special illustration.

3.2.3.1 Demographic characteristics of individuals

Considering current city as representative location, figure 3.2 displays the geographical location distribution of current city reporters over the globe. The color of each dot in the figure corresponds to the number of users in a city, applying a spectrum of colors ranging from blue (low), green, yellow to red (high). We can see that the red dots are mainly located in the east coast of North America as

well as Europe, thus we infer that people from North America and Europe are the dominant users on Facebook. We also observe that people in coastal regions are more active than people situated inland. In addition, a few blue dots are noticed in the oceans, which might indicate some users report fake locations. We ignore them as the number is very small.

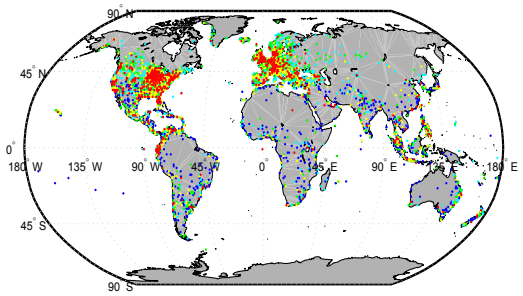
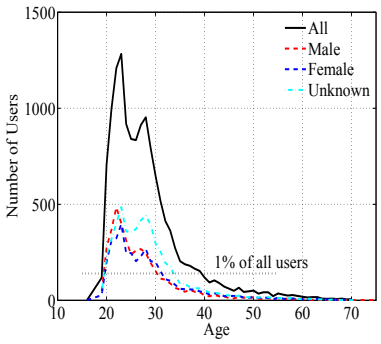
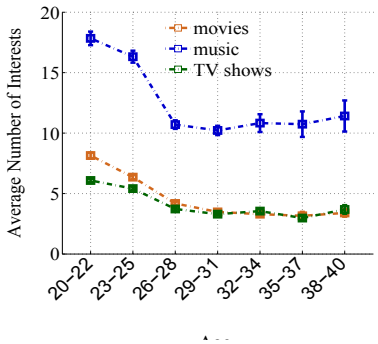


Figure 3.2: Location distribution of users. The color of each dot in the figure corresponds to the number of users in a city, applying a spectrum of colors ranging from blue (low), green, yellow to red (high).

Moreover, we study the distribution of users by age. Figure 3.3(a) displays the distributions of age reporters with respect to female, male, unknown gender and all. Among all the age reporters, 4196 are male and 4096 are female. We notice that the age distributions of males and females are similar to each other. We also observe that the user distributions are skewed by age following with a long tail. The users in the 20-30 span of years are the most representative users in our data set; while the proportion of the users older than 40 or younger than 20 in our data set is rather small (less than 10% in total). Besides, we choose 3 years as an age interval and cluster age reporters in the age range of 20-40 into seven age groups. Taking movies, music and TV shows as the representative interest, figure 3.3(b) examines the average number of interests that each user exhibits according to different age groups. It reveals that the young users report more interests than middle-age users.



(a) Users distribution



(b) Interests distribution

Figure 3.3: Distributions by age. Figure 3.3(a) plots the number of users at each age. Figure 3.3(b) presents the average numbers of interests that users report at different age groups.

3.2.3.2 Demographic characteristics of friends

In this section, we further reveal the demographic characteristics between friends in terms of gender, location, and age respectively.

We first examine the distribution of friends by gender combinations: cross-gender friends and same-gender friends. This analysis is conducted on gender reporters. Particularly, for each gender reporter, we rely on her friends that are also gender reporters and calculate the percentage of friends in the same-/cross- gender respectively. Figure 3.4 displays the CDF of the percentage of friends by gender combinations. We observe that only around 40% of users exhibit the same gender with less than half of their friends, while more than 60% of gender reporters make fewer friends (i.e., less than half) with opposite gender. It indicates that people prefer to make friends with others of the same gender, especially for men.

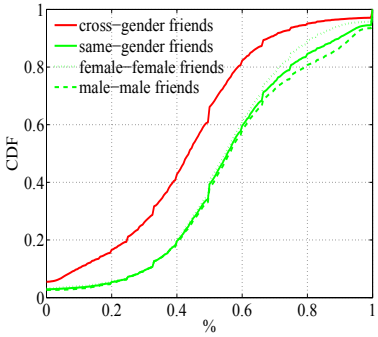


Figure 3.4: CDF of friends distribution by gender combinations

In addition, we track how age affects the friendship between people. Figure 3.5 displays the distribution of pairs at various age differences. It reveals that people are more likely to make friends with others at the same age or at an age gap of 1-2 years. The percentage of friend pairs decreases rapidly as age difference increases when it is larger than 1 year. Besides, we also notice that the percentages of friend pairs are less than the numbers of random pairs at the age differences in the range of $3-13$ years. When age difference is larger than 13 years, people make friends following the random probabilities. We infer that people are more likely to make friends with others who are in the similar ages.

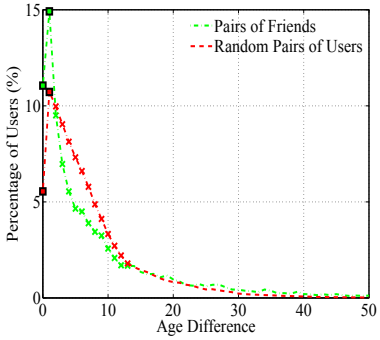


Figure 3.5: Pairs distribution by age difference

We calculate geographical distances between pairs and illustrate the pairs distribution with distances in Figure 3.6. From the upper subfigure, we see that the distance distribution of friend pairs

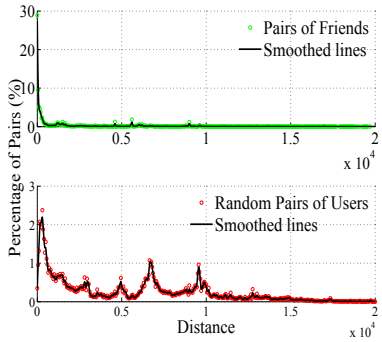


Figure 3.6: Pairs distribution by distance

is strongly skewed to the left. It falls dramatically from the start, bottoms out at the distance of 400 kilometers, and then stays at a very low value as the distance increases. Among all the friend pairs in the experiments, 28.9% of them come from the same city and 43.43% of the friends live less than 100 kilometers apart. Whereas, the lower subfigure shows that the percentages of random pairs fluctuate by distances with a gradual downward trend. The peaks and drops at some specific distances may reveal geographical characteristics. For instance, the peaks at distances of 5000 km and 6500 km may respectively indicate the width of America and the width of Atlantic. The different distributions of friend pairs and random pairs, in other words, mean that people tend to make friends within a short distance.

3.3 Data Limitation

We discuss two limitations of our data set: amount limitation and information limitation. We further demonstrate that our research is not constrained by these limitations.

3.3.1 Amount Limitation

Comparing to 1.2 billion monthly active users in Facebook, we admit that around 500K users in our date set is just a tip of ice-berg. We have tended to demonstrate the representativeness of our data set by comparing the characters (e.g., location distribution, age distribution) revealed from *Friends Group* and *Random Group* in section 3.2.2. Therefore, we believe this data set and the research based on the data set are representative. To the best of our knowledge, this data set is one of the largest and most comprehensive data set regarding Facebook users' profile information and social links in academic research.

3.3.2 Information Limitation

On one hand, we are only able to collect users' public information. Two possible reasons lead us not seeing users' information. One is that a user has not describe the information in her Facebook profile; the other is that she sets the privacy of information. However, this limitation stirs us to explore whether it is possible to infer users' invisible information on basis of the public information; or is it really secure of users' private information if users simply hide the privacy-sensitive attribute from public. We will illustrate the details about these questions in the following chapters.

On the other hand, there are a lot of other data information we can access from Facebook, such as users' interaction, activity (e.g., sharing and posts), check-ins and so on. It is true we can advance our research if we leverage this information. But we have to admit the difficulty and even impossibility of obtaining all this information from Facebook due to privacy issues. Fortunately, the current research problems in this dissertation are not confined by this limitation.

Chapter 4

Effects of User Similarity on Link Prediction for New Users

Contents

4.1 Introduction

In OSNs, social links (e.g., *friends* in Facebook, *follower-followee* in Twitter) play an important role in users' experience as well as in the success of the OSN. If a user's links are well-established, he can use the social network more frequently [132][129][130]. Therefore, a high-quality link prediction is required to allow OSNs to recommend useful links to users. Especially, when it comes to a new user who has not created any link, the link prediction (*new-user link prediction*) becomes even more crucial, because it can be used to recommend friends for new users to start building their social networks. A poor prediction in the first place may discourage new users from using the platform.

Many approaches have been proposed to predict users' potential links depending on the existing ones that they have already established [129][130][58]. However, these approaches cannot be

adopted to the new-user link prediction since the new users have not created any link. Recently, by using *cross-platform* approaches, a few studies have begun to tackle this new-user link prediction problem. These studies predict a new user's links in a certain OSN platform by porting that user's well-established links from other OSNs [133][134]. Nevertheless, the application of these cross-platform methods in real-life scenarios may face some problems. First, two OSNs may not agree to share users' links as users' information are generally private, confidential and valuable to them [55]. Second, users may not give their consent to be tracked back to their information in other OSNs or users intend to use different OSNs for different purposes (e.g., LinkedIn for professional and Facebook for personal). Due to these problems, in this work, we study the new-user link prediction problem in a *single-platform* instead of cross-platform.

Our single-platform approach is to leverage the attributes (e.g., workplace, high school and hometown) provided by new users when they register their accounts, as well as information from existing users. It is inspired by the previous studies showing that the similarity between users' attributes reflects their relationship to some extent. For instance, people who share more common interests are more likely to be friends [82][135][136][93]. As it is practical for OSNs to request user attributes during the registration, our approach is applicable in real-life scenarios.

Without the loss of generality, new-user link prediction problem can be considered to predict whether a new user u will link to a given existing user v or not. In particular, given u with attributes and v with both attributes and friends, we attempt to extract some social features that can indicate the probability of u linking to v. By using Support Vector Machines (SVM) [137], we train a link prediction model to determine whether u will link to v based on the combination of the extracted social features.

Exploring appropriate social features is crucial and challenging since it directly affects the capacity of the prediction model and the available information is very limited. We propose to fully use the obtainable information and extract the following three types of social features:

- *Basic features:* There are two types of basic features: *binary similarity* and *number of common attributes*. The former is calculated by comparing two users u and v by each attribute (e.g., current city). The latter is the total number of the same attributes between them.
- *Derived features:* We further describe the relation between two users' attributes by various ways, e.g., the geographic distance between their current cities or their interest similarity.
- *Latent relation:* We use a latent relation score to estimate how much u and friends of v share the same attributes. We show that two users probably obtain a higher score if they are friends.

In summary, this chapter has the following contributions: (1) We explore multiple social features to predict links for new users who have not created any link. To the authors' best knowledge, this is the first work to address the new-user link prediction problem by leveraging the information from a single-platform. (2) To evaluate our approach, we use a Facebook data set including 479,000 users. For each user, we record his demographics, interests and links (friends). Results show that all the features we proposed in this chapter can significantly improve the performance of the new-user link prediction.

4.2 Link Prediction

The goal of this work is, given an undirected social network graph and a new user u who has not created any link yet, to determine which users in the given graph the new user u will connect to. In this section, we formulate the link prediction problem and describe our solution.

4.2.1 Problem Statement

Considering a given undirected social network graph $\mathscr{G} = (\mathscr{U}, E)$, where U is a set of *existing users* in the social network graph; E is a set of undirected links $e\langle u, v\rangle$ between users u and v where u, $v \in U$. Apart from the links, users on social networks usually expose other personal attribute such as age, hometown, college and work. Therefore, for each *existing user* v, we generate an *attribute vector*, denoted as $\mathscr{A} = \langle a_1(v), a_2(v), \cdots a_m(v)\rangle$. We also gather all of v's links into a *friend set*, denoted as $\mathscr{F}(v) = \{f | f \in U \wedge e\langle v, f\rangle \in E\}$. Then, we can use a tuple to represent an *existing user* as $v : \langle a_1(v), a_2(v), \cdots a_m(v), \mathscr{F}(v)\rangle$. Note that, as the user v may not complete all the attributes or not expose his friend set, some elements in the tuple can be *null*.

For a new user who has not constructed any link, OSNs usually request him to provide some personal information when he is signing up. For this reason, without existing links (i.e., no friend set), a new user u can be represented by a tuple merely with attributes as: $u : \langle a_1(u), a_2(u), \cdots a_m(u)\rangle$.

According to the goal of this work — to distinguish which of the *existing users* $v \in U$ are preferred to construct a link by u and which are not, we classify a candidate set of *existing users* (i.e., $\mathscr{C}$) into two categories: *linked-users* (i.e., $\mathscr{L}$) and *de-linked-users* (i.e., $\mathscr{D}$). Note that $\mathscr{C} = \mathscr{L} \bigcup \mathscr{D}$, where $\mathscr{C} \in U$. We assume that the users in $\mathscr{L}$ are more likely to get linked by u than the users in $\mathscr{D}$.

On the basis of the above establishments, the problem of ***new-user link prediction*** can be formally stated as: *Given a social network graph $\mathscr{G} = (U, E)$ where each $v \in U$ contains an attribute vector and a friend set, $v : \langle a_1(v), a_2(v), \cdots a_m(v), \mathscr{F}(v)\rangle$, a set of existing user candidates $\mathscr{C} \subseteq U$, a given new user u who is represented by an attribute vector (i.e., $u : \langle a_1(u), a_2(u), \cdots a_m(u)\rangle$), predict which of the users in $\mathscr{C}$ that u may create links to, labeled as $\mathscr{L}$(linked-users), and which of the users that u may not, labeled as $\mathscr{D}$ (de-linked-users).*

4.2.2 Workflow of Link Prediction for New User

Given a new user u who reveals some profile attributes and a set $\mathscr{C}$ of existing users who exhibit both attributes and friends, the basic idea is exploiting all the obtainable information to figure out existing users that are similar to u from $\mathscr{C}$ as the *linked-users* ($\mathscr{L}$), for much existing work has proved that people are likely to connect to another if they are similar to each other [136][93][138][82][135].

Based on the mentioned idea, we model the friend probability, which measures the probability that u will create a link to $v \in \mathscr{C}$, by computing their similarity based on their obtainable information (i.e., u and v's *attribute vector*, v's *friend set* and v's friends' *attribute vector*). Specifically, we leverage SVM [137] to train a link prediction model, which describes the friend probability by a combination of multiple social features (i.e., ψ_{uv}). To train the model, we generate a training data set which gathers information of a number of user pairs. Each user pair corresponds to a label z_i and multiple social features $\boldsymbol{x}_i$. Note that z_i equals 1 if two users are friends; otherwise, z_i equals 0. With the data set, we aim at training a set of parameters (w) and making the social features' parameterized combination describe the pattern of connectivity between users. In other words, with taking the social features that are parameterized by the trained w, we can compute the friend probability between u and v, and then determine whether v belongs to $\mathscr{L}$ or $\mathscr{D}$ for the new user u.

Thus, constructing the SVM-based link prediction model is addressing the optimization problem as follows:

$$\min F(w) = \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{q} \xi_i$$
$$\text{subject to: } \begin{cases} \xi_i \geqslant 0 \\ z_i \langle w, \boldsymbol{x}_i \rangle \geqslant 1 - \xi_i \end{cases} \tag{4.1}$$

where q stands for the total number of the user pairs and i denote the ith pair; λ is a constant and $\xi_i (i = 1, ..., q)$ are slack variables for optimization.

4.3 Empirical Studies on Relationship Similarity

Capturing good social features that are exploited in the learning algorithm is critical and challenging [58]. For training the model with enough features, we take various ways to extract plenty of features with limited social attributes. Particularly, we conduct this study based on a real social data set which we have crawled from Facebook. We first briefly introduce the data set and then illustrate multiple captured social features. We also reveal some relations between friend probability and social features. Note that, although the social features seem tightly depending on the social attributes in Facebook, our work is easy to be extended to other social network platforms.

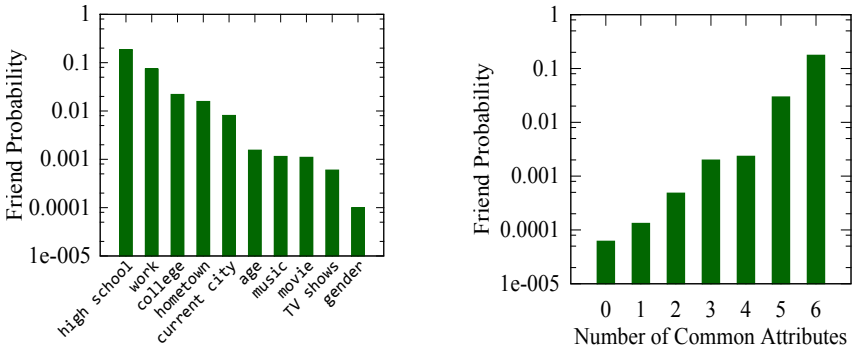
In this work, we think of the user's demographics and interests as social attributes to conduct the empirical studies. Specially, we consider ten social attributes: current city, hometown, high school, college, work, age, gender, user's favorite music, movies and TV shows.

4.3.1 Basic Features

With a new user's social attributes, the most straight-forward way to predict his link is to look for some users who exhibit some common attributes with the new user. For instance, if a new user u states that he is working at TELECOM SUDPARIS, he might know others working in TELECOM SUDPARIS. Therefore, in this OSN, if there is an existing user v stating that he is working for TELECOM SUDPARIS, it is more probable that u will link to v than others. We define two types of basic features: *binary similarity* and *number of common attributes*.

Binary similarity estimates whether two users are same or not in one certain attribute. For instance, binary similarity on work of u and v equals 1 if they work in the same company or organization; otherwise, it is 0. Moreover, we sum up the binary similarity on all the attributes to obtain the *number of common attributes* as another basic feature, since two users are more likely to be friends if they share more attributes. With the Facebook data set, we study the relations between *friend probability* and the two basic features.

Figure 4.1(a) displays the friend probability of two users if they have same value on a certain attribute. We observe that users from the same high school and workplace might connect with each other with the highest probabilities — around 15% and 7.4% respectively. Figure 4.1(b) reveals the increase of the friend probability when the number of common attributes grows. The user pairs who share five common attributes merely have 3% of probability to be friends. Only 0.3% of user pairs could share six common attributes, although their friend probability reach to 17.6%. The above observations imply that only using the two kinds of basic features may still be hard to predict links correctly and inspire us to explore more social features to describe user pairs' connectivity patterns.



(a) Individual attribute (b) Common attributes

Figure 4.1: Friend probability by basic features

4.3.2 Derived Features

In this section, we try to capture more social features from two user's attributes, which are called derived features. For different attributes, we propose three feature extraction methods, and thus get three sub-categories of derived features: *distance features*, *correlation features* and *similarity features*.

4.3.2.1 Attribute Distance

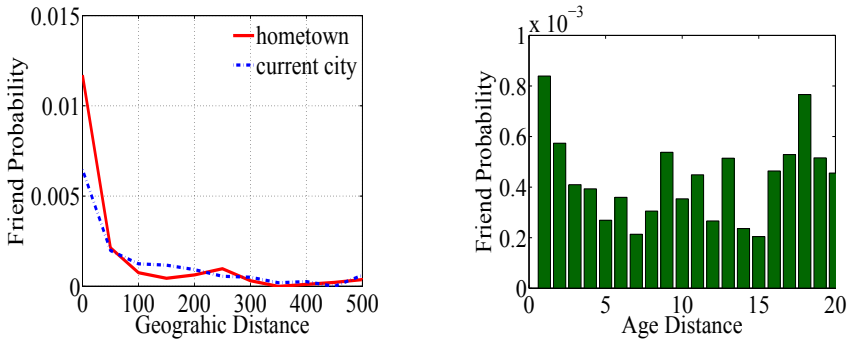
Some existing studies indicate the homophily principle that people are more likely to link to others who are closer to them [82][93]. We attempt to use the distance function to describe the closeness in terms of the location-related attribute (e.g., current city and hometown) and age. We can calculate users' geographic distance by exploiting the location's coordinates and look further into how the distance would affect the friend probability. The absolute age difference between two users is also introduced as a distance feature.

Figure 4.2(a) shows the effects of geographic distance between users on the users' friend probability which holds the homophily principle both for current city and hometown. Figure 4.2(b) reveals that the friend probability does not correlate to the age distance. Nevertheless, the observation exhibits its rationality: people usually link to various people in different ages. For instance, a teenager may link to his parents, and a younger employee could link to an elder leader.

4.3.2.2 Attribute Correlation

We have found that people from the same high school, workplace or college link to each other with a relatively larger possibility. Besides, in reality, people from different organizations may also exhibit frequent links because of the tight collaboration and relations between them. For example, TELECOM SUDPARIS as a telecommunication institute may have a very close relationship with TELECOM ORANGE LAB because of their regular project collaborations. Therefore, many of the employees from these two workplaces may know each other and establish links.

To accurately describe this connectivity pattern, we construct a *attribute correlation matrix* which learns the friend probability between users with specific value combination in one attribute (i.e., high school, workplace and college). For instance, to set up a *work correlation matrix*, both of the columns and rows represent all the workplaces that users report, and the cross-cell of ith column (representing work W_i) and jth row (representing work W_j) stands for the friend probability between



(a) Geographic distance (b) Age distance

Figure 4.2: Friend probability by distance

the employees in work W_i and the employees in work W_j, i.e., $fp_w(i,j)$. Marking the work attribute as a_w, we get the following formula:

$$\begin{aligned} fp_w(i,j) &= P(e\langle u,v\rangle \in E \mid a_w(u) = W_i \wedge a_w(v) = W_j) \\ &= \frac{|\{e\langle u,v\rangle \in E \mid a_w(u) = W_i \wedge a_w(v) = W_j\}|}{|\{(u,v) \mid a_w(u) = W_i \wedge a_w(v) = W_j \wedge u \neq v\}|} \end{aligned}$$

The numerator in the above formula is the number of friend pairs where one's work is W_i and the other's work is W_j; the denominator is the number of all possible user pairs where two users work in W_i and W_j respectively. Back to the previous example, assume W_i is TELECOM SUDPARIS and W_j is TELECOM ORANGE LAB, then $fp_w(i,j)$ is the probability that two employees from the two institutes are friends. Besides the attribute of work, we also construct such matrices for high school and college.

Note that the friend probability study relies on an aggregation number of existing users with complete required information (i.e., friendships and value on the attribute). According to the size of population and various number of distinct attribute values (e.g., the number of workplaces reported by users), the construction of *attribute correlation matrix* may take a long time. However, it is feasible as the matrix construction can be calculated off-line, and does not need to be updated frequently.

4.3.2.3 Interest Similarity

Cosine similarity is widely used to estimate the closeness of two vectors. Hence, for the attribute with a value of vector, like favorite music, movies and TV shows where users present multiple items, we apply the cosine similarity to describe two users' interest similarity. For the detailed description about how to calculate cosine similarity between two users' interests, please refer to Chapter 2. According to the Figure 4.3, we verify that users with similar interests link to each other with high probability, which is also observed by other work [136][93][138].

4.3.3 Latent Relation

Both basic and derived features are constructed by only considering the new user u and the *existing user* v's attribute vectors; besides, another kind of information is still available— v's friend set. If u and v's friends are similar, the link $e\langle u,v\rangle$ will probably be created. We call the relation between u

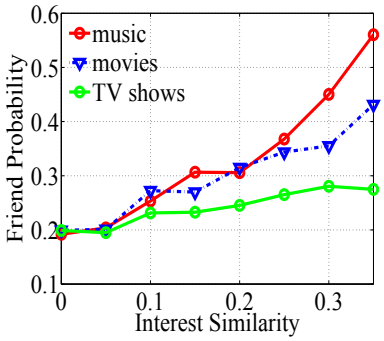


Figure 4.3: Friend probability by interest similarity

and v captured through the relations between u and the friends of v as *latent relation*. We consider the *latent attribute relation* between u and v, which is estimated by the *latent attribute links* between u and the friends of v. Specifically, one latent attribute link is created if u has a same attribute with one of v's friends.

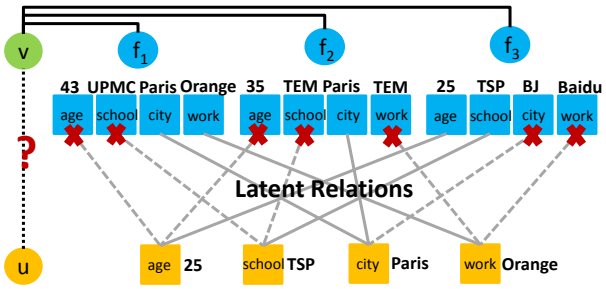


Figure 4.4: An example of latent relations between two users

Figure 4.4 illustrates an example to show the latent links between a new user u and an *existing user* v, where v has three friends f_1, f_2 and f_3. We observe that u and f_1 share two attributes — work and current city — which construct two latent links between u and v. u also links to v's other friends f_2 and f_3 by various attributes. In addition, we observe many disconnections between u and v's friends on attributes, denoted by the red cross and dotted lines in the figure.

The problem then becomes how to quantify these latent links and disconnections between u and v's friends, so as to model the latent relation between u and v. Intuitively, u and v exhibit higher probability to be friends if there are more latent links and less disconnections. Therefore, we *reward* the latent relation of u and v if there is one latent link, and *punish* the latent relation if there is one disconnection. According to this idea, we estimate u and v's latent relation by $r - \alpha q$, where r equals the number of latent links, q is the number of disconnections and α is a regulator for *punish* value [139]. Accordingly, we compute a latent relation score as:

$$scr_{lr} = \frac{1}{1 + e^{-\beta(r-\alpha q)}} \tag{4.2}$$

where β is an exponential regulator. Figure 4.5 displays the relation between friend probability and the latent relation score when $\alpha = 0.05$, $\beta = 0.05$. It reveals that the friend probability would increase if two users exhibit more latent links and less disconnections (i.e., larger latent relation score).

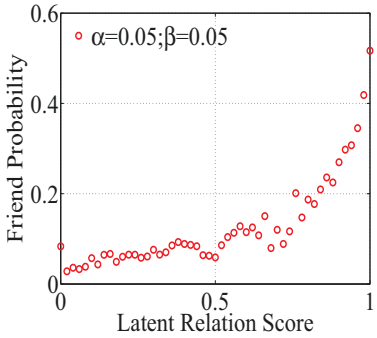


Figure 4.5: Latent relation scores between two users

Note that, when u reveals few attributes and most basic and derived features of u and v cannot be obtained, this latent relation score can be especially important for determining whether $e\langle u, v\rangle$ will exist.

4.4 Evaluation

In this section, we evaluate our proposed approach on the crawled Facebook data set. We first introduce the experiment setup and then report the experiment results.

4.4.1 Experiment Setup

Taking the prediction algorithm described in Section 4.2.2, we leverage all the introduced social features in Section 4.3 to train our new user link prediction model. In particular, we use 1) ***B***asic feature, i.e., the number of common attributes; 2) ***D***erived feature, including the distance of current city and hometown, the attribute correlation on work, high school and college; and 3) latent ***R***elation score. We call our proposed model as ***BDRlink*** model. Note that, ***BDRlink*** model does not exploit binary similarity on each attribute because it has already been involved in derived feature. For instance, if the distance of current city between two users equals to 0, it indicates they are in the same city (i.e., binary similarity is 1); otherwise, binary similarity is 0.

We compare ***BDRlink*** model with three baselines — ***Blink*** model, ***Dlink*** model, and ***BDlink*** model:

- ***Blink*** model merely considers basic social feature which includes the number of common attribute and binary similarity on all attributes (i.e., current city, hometown, high school, college, work, age, gender, user's favorite music, movies and TV shows).
- ***Dlink*** model merely considers derived social feature which includes current city distance, hometown distance, age distance, high school correlation, college correlation, work correlation, music similarity, movies similarity and TV shows similarity).
- ***BDlink*** model takes into account the number of common attribute (basic feature) and all the derived features that are used to train ***Dlink***.

Note that these models are trained with users who reveal friends and more than 3 attributes. We randomly couple two users into a user pair and select one of the two users as the new user by removing his friends.

4.4.2 Evaluation Results

We evaluate the proposed ***BDRlink*** model from three perspectives: 1) we compare the prediction performance of ***BDRlink***, ***Blink***, ***Dlink*** and ***BDlink*** models in terms of ROC curves with 10-fold cross validation; 2) we further carry out 'leave-one-feature-out' model comparison to investigate the influence of various social features on link prediction; 3) we evaluate the prediction performance of ***BDRlink*** by the number of available attributes from the new user, so as to inspect and verify whether the new user can derive better friends prediction if they provide more information.

Prediction performance comparison: We draw the ROC curves of four prediction models, shown in Figure 4.6. We also note the corresponding Area Under Curves (AUCs) in the legend. First of all, compared to the diagonal line (i.e., AUC= 50%) which represents the performance of random guess, all the four models with our captured social features can predict more accurately. We notice that ***Blink*** model and ***Dlink*** model exhibit equal prediction capacity as they almost achieve a same AUC of 68.8%. Additionally, the combination of basic features and derived features can slightly enlarge the AUC from 68.8% to 71%. Among the four compared model, ***BDRlink*** model generates the largest AUC and its AUC significantly outperforms the other three models by 14%, 14% and 12% respectively. It reveals that the attribute based latent relations between users not only works for the link prediction but also plays a very important role in the link prediction.

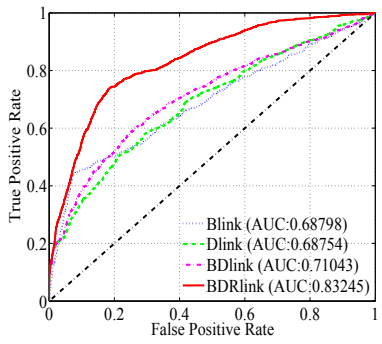


Figure 4.6: ROC curves comparison

Leave-one-feature-out: To investigate whether the social features leveraged in ***BDRlink*** model would improve the prediction performance or not, we leverage the state-of-the-art 'leave-one-feature-out' strategy and remove one feature from the overall features to train additional models. Specifically, five feature types — basic feature, distance feature, attribute correlation feature, interest similarity feature and latent relation score— are considered. Thus, we train five 'leave-one-feature-out' prediction models by taking out one of the five types of features, namely *No basic feature* model, *No distance feature* model, *No attribute correlation feature* model, *No interest similarity* model and *No latent relation score* model.

Table 4.1 compares the AUCs of the five 'leave-one-feature-out' models and the ***BDRlink*** model. We observe that ***BDRlink*** model outperforms all the other models which means removing any of the used social features would decrease its prediction power. In addition, comparing the five 'leave-one-feature-out' models, we find that various social features impact the prediction performance in different degrees. For instance, removing basic features or interest similarity, the prediction performance does not fall down much; whereas latent relation score is quite sensitive to the prediction as the performance decrease obviously when it is removed.

AUC by number of available attribute: In this experiment, we aim to validate whether ***BDR-***

Type of Model	AUC
No basic feature	0.8139
No distance feature	0.7563
No attribute correlation feature	0.7863
No interest similarity	0.8107
No latent relation score	0.7104
BDRlink	**0.8325**

Table 4.1: Leave-one-feature-out comparison

link can predict links more accurately if new users provide more attributes. We group the user pairs according to the number of attributes obtained from new users and test the prediction performance of ***BDRlink*** for each group in terms of AUC. Table 4.2 lists the AUC values by various attributes numbers. The results reveal that the prediction accuracy would increase if new users provide more attributes.

#Attributes	3	4	5	6	7	8	9
AUC	0.66	0.70	0.72	0.73	0.75	0.83	0.87

Table 4.2: AUC by varying number of available attributes

4.5 Summary

This chapter proposes a novel method to predict links for new users in OSNs. It leverages the attributes from new users provided at the registration phase and the profile information (attributes and links) from existing users to generate a number of effective social features. The correlation between the friend probability and these social features is investigated to select effective features for training a SVM-based link prediction model — ***BDRlink***. The empirical experiments show that the ***BDRlink*** model performs better than the other three baseline models. The leave-one-feature-out test reveals that each of the proposed social features contribute significantly to the prediction model.

Chapter 5

Mining User Similarity for Content Discovery in Social P2P Network

Contents

5.1 Introduction

Unlike the traditional client/server model, each node[1] in Peer-to-Peer (P2P) networks acts both as a server and a client. Thus, the node is allowed to share resources (e.g., files, peripherals) directly with others, which makes P2P networks quite popular. A report from Palo Alto Network [140] shows that P2P file sharing consumes 14% of overall bandwidth between November 2011 and May 2012, surpassing other applications. Furthermore, with the increasing demand for multimedia entertainment, P2P networks are being broadly used in video streaming applications, such as PPstream, PPLive and UUSee.

In P2P networks, content discovery is a critical problem. There are two typical classes of its solutions: structured and unstructured. Structured P2P, using Distributed Hash Table (DHT) [141][142][143], is efficient but inflexible under a dynamic environment. Compared to unstructured P2P, it also produces more overheads for finding popular content. Unstructured P2P is widely used over the Internet [144]. Gnutella [145] is the first practical implementation of unstructured P2P. However, it applies flooding to search content and cannot adapt to the complex networks. Although many improved approaches [146][147][109][148] have been proposed, content discovery still remains a challenge in unstructured P2P, especially for unpopular content which is stored by only a few nodes. This is due to the lack of global network topologies and content information.

Nevertheless, similarly without global information of complex human networks, humans can efficiently find out specific people by exploiting their own **Social Information** (i.e., *friends*, and friends' *background information* such as nationality, interests and city). On one hand, researchers tend to verify this through experiments. In 1950s, from real human networks, Milgram revealed that any randomly selected people can reach the others by about six people on average [149]. It has also been demonstrated that users on Facebook can reach others through 3.74 intermediaries [150]. On the other hand, researchers are also inspired to extract the underlying characteristics of people behavior (e.g., people communicate more with each other when they have more similarity [92]), and leverage them to enhance performance in diverse systems, such as prediction systems [151], recommendation systems [152], and advertisement systems [153].

In this work, we are motivated to investigate how social information could benefit content discovery in unstructured P2P networks. In particular, by learning from humans' experience on finding people, we propose to exploit social information from real social networks and look for content via a subset of friends that are selected based on their social information. Our approach is different from the existing work. First, we do not infer nodes' preferences and social relationships by monitoring their behavior as suggested in [108][154], since such information is explicitly exposed among friends on social networks. Either, we do not group nodes into communities by exploiting complex algorithms presented in [114][115]; instead, we use the user-generated friendships which

[1] nodes & users are exchangeable in this chapter

are straight-forward and reliable. In addition, we especially look into content discovery regarding users' personal interests (i.e., users' own interests which include both popular and unpopular content) rather than only focus on the popular ones.

5.1.1 Challenges

It is a non-trivial task of leveraging social information to improve content discovery in P2P networks. We encounter the following challenges:

First, to leverage social information into P2P network and verify the newly proposed social P2P mechanism, real social information data are required. Although the recent online social networks reflecting human networks provide plenty of users' social information, it is not easy to collect such social information.

Second, since the existing P2P platforms do not involve or exploit social information, how to associate the nodes in P2P networks with social information is another challenge.

Third, even if we are able to solve the second challenge and enrich nodes in P2P networks with their associated social information, it is still hard to properly exploit such information and achieve good performances (e.g., high success rate and low cost) for content discovery.

5.1.2 Method and Contributions

To solve the challenges, we first capture a large volume of social information from Facebook. The studies on these data reveal that: (1) a node shares higher similarity with its friends than with randomly selected nodes; (2) a node's friends present different degrees of *Similarity* to itself and report different amount of *Knowledge* (e.g., friends, interests). Intuitively, a node is more likely to find content from those nodes that present higher similarity and more knowledge. Therefore, we then build up a social P2P Network Model that connects nodes with their friends rather than randomly selected nodes. On top of this model, we propose a Top *K social-DRWR-P2P* Search Algorithm, which selects a subset of friends with higher similarity and more knowledge. The details are as follows:

Social P2P Network Model: The model projects users' social information in social networks into corresponding nodes of users in a P2P network, and links nodes according to users' friendships. In the model, a node estimates the weight of a link, which is defined as a friend's content discovery weight, by applying two types of social features: the friend's *Knowledge*; and the *Similarity* between the node and its friend.

Top *K social-DRWR-P2P* Search Algorithm: Based on the social P2P model, the algorithm extracts the latent friendships among a node's friends and computes scores for its friends according to their content discovery weights by using a modified Distributed Random Walks with Restart (DRWR) method. Eventually, by using the algorithm, a node ranks its friends based on the scores and forwards queries to its top *K* friends (receivers) on the ranking list.

The proposed method (i.e., *social-DRWR-P2P*[2]) is evaluated on Facebook data. It achieves a higher success rate and lower cost than *social-P2P*[3] and *traditional-P2P*[4]. Especially, *social-DRWR-P2P* could reach 100% of Search Success Rate (SSR) by selecting top 20 friends within two-hop for personal interests searching. Under the same condition, the compared methods achieve 90.5% and

[2] *social-DRWR-P2P* selects receivers by the proposed algorithm over the social P2P network model

[3] *social-P2P* selects receivers randomly among the sender's friends over the social P2P network model

[4] In *traditional-P2P*, receivers are randomly selected among all the other nodes

61.4% of SSR respectively. In addition, *social-DRWR-P2P* achieves 6.5 Hits on average, which is more than 8 times superior to the compared methods.

We conclude the contributions in this chapter:

- We collect social information of 500K user profiles from Facebook. We also carry out extensive studies on these data and extract useful characteristics which inspire the design of the content discovery mechanism.
- We propose a social P2P network model and associate the nodes in P2P networks with social information reasonably. The model exhibits two advantages for content discovery: first, the model links nodes with their friends who can discover users' interests with higher probabilities compared to the randomly selected nodes; second, the node in this model estimates its friends' content discovery weights by integrating social features of *Knowledge* and *Similarity*.
- Based on the social P2P network model, we extract latent friendships among a node's friends and further propose a Top K *social-DRWR-P2P* algorithm to select a subset of optimal friends. In addition, we exploit a parameter optimization approach to adjusting social feature parameters in the algorithm. The extensive evaluations reveal the efficiency of the proposed method, especially for users' personal interests search.
- We discuss reasonability of the social P2P network model in Section 5.3.1.2 and discuss practicality of the proposed mechanism in Section 5.6. We give suggestions about how to apply the proposed mechanism to unstructured P2P applications.

The rest of this chapter is organized as follows. Section 5.2 describes and analyzes Facebook data set. We discuss the proposed mechanism in Section 5.3. In Section 5.4 we elaborate experimental methodology and parameters setup. We evaluate the proposed mechanism in Section 5.5. Section 5.6 discusses the practicality of the proposed mechanism and Section 5.7 concludes this chapter.

5.2 Empirical Analysis

In this section, we conduct empirical studies on our data set and tend to inspect potential social information for content discovery in social P2P network. We assume that a user is easier to provide content if she associates with more interests; we also assume that it is more likely to request a content from a similar user. Therefore, we first look into which users are associated with more interests; then we compare user similarity inside *Friends Group* and *Random Group* to reveal the potential assistant of friendship in content discovery. Finally, we study the distributions of interests' popularity in both Groups.

5.2.1 Users' Associated Interests

Concerning an interest catalogue with M interests in total and a user associating with m interests, the possibility of discovering any interest in the catalogue from the user equals m/M. It is an increasing function of m, which implies that the users who associate with more interests can provide larger probability to discover any interests for others. Therefore, we expect to reveal the users who associate with more interests.

Method(M)1: A user's associated interests refer to both the user's own interests and her friends' interests. We can easily decide a user's own Interest Degree, thus here we focus on the relation between a user's Friend Degree and the total Interest Degree of all her friends. In particular, given a

user with Friend Degree of n, we compute its Total Interest Degree of Friends by the overall number of interests that all her friends present. We plot users' total Interest Degree of Friends by their Friend Degree in figure 5.1.

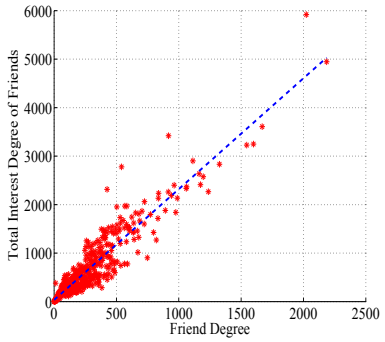


Figure 5.1: Total interest degree of friends by friend degree

Observation(O)1: Figure 5.1 reveals that the Friend Degree strongly correlates to the total Interest Degree of Friends. It is observed that the Total Interest Degree of Friends goes up with the increasing of Friend Degree. This indicates that a user can associate (access from her friends) with more interests if she has more friends. The correlation between the total Interest Degree of Friends and the Friend Degree can be modeled linearly.

Inference(I)1: O1 demonstrates that the users with higher Friend Degree can access more interests from their friends; while the users with higher Interest Degree have more interests by themselves according to the definition of Interest Degree. Hence, we infer that a user is more likely to find content from other users with higher Friend Degree and Interest Degree, since users associating with more interests can provide more probability to achieve content discovery.

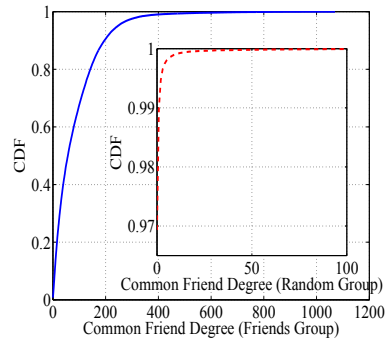
5.2.2 User Similarity

We suppose that if a user U shares more common interests with user A than with user B, it is easier for U to find her interests from A than from B. Similarly, if U shares more common friends with A than with B, we assume that U has a stronger relationship with A than with B. Therefore, intuitively, the stronger relationships imply the more latent connections, common activities and common interests which might be beneficial to content discovery. In this section, we conduct studies on users' similarity and expect that a user present more similarity with her friends than with strangers.

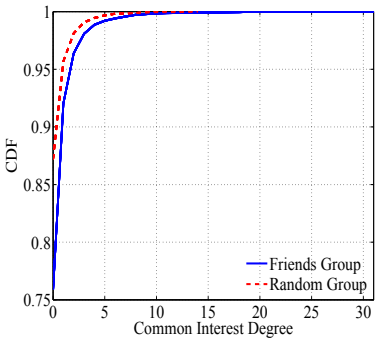
M2: We learn similarity between two users by Common Friend Degree and Common Interest Degree. We further define Interest Correlation to compare interest similarity inside the two groups.

M2.1: We calculate the Common Friend/Interest Degree in *Friends Group* by counting the number of common friends/interests between users and their friends. For *Random Group*, we select two users at random and compute the Common Friend/Interest Degree by counting the number of common friends/interests between them. The more common friends/interests two users share, the higher similarity they have. Figure 5.2(a) shows the CDF of Common Friend Degree. The inside figure plots the CDF of the Common Friend/Interest Degree between strangers in *Random Group* and the outside figure shows the CDF of the Common Friend/Interest Degree between friends in *Friends Group*. Figure 5.2(b) presents Common Interest Degrees of the two groups.

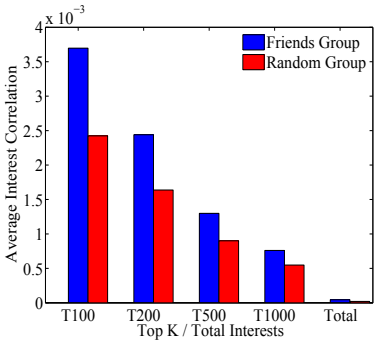
M2.2: If a user claims a certain interest as one of her own interests, we call the user as a *fan* of this interest. The Interest Correlation of a certain interest is defined as the fraction of the fan number



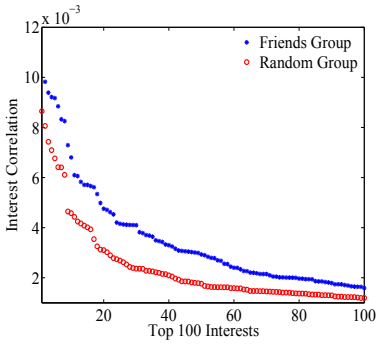
(a) CDF of common friend degree



(b) CDF of common interest degree



(c) Average interest correlation of top K interests



(d) Interest correlation of top 100 interests

Figure 5.2: User similarity

of the interest to the total fan number of all the interests in a corresponding group as given below:

$$IC_{I_j} = \frac{\sum U_{I_j}}{\sum_{I_i \in I} \sum U_{I_i}}$$

where $\sum U_{I_j}$ is the fan number of interest I_j and I is the total number of interests in the group.

We rank all the interests in *Friends Group* and *Random Group* respectively by their Interest Correlation, and compute the Average Interest Correlation of the top K interests ($K = 100, 200, 500, 1000$, and the total number of interests) to compare the entire Interest Correlations inside the two groups. The larger the entire Interest Correlation within a group obtains, the higher is the interest similarity among users inside the group. In addition, we compare the individual interest correlation of the top 100 interests in the two groups.

O2: The investigations for similarity between users show that a user has higher Common Friend/Interest Degree with her friends than with strangers. We also note that different friends of a user share different Common Friend/Interest Degree with the user. In addition, it is observed that the Interest Correlations are higher among friends in *Friends Group* than those among strangers in *Random Group*. In particular, we observe:

O2.1: In figure 5.2(a), more than 99% of the randomly selected pairs of users have no common friends in the *Random Group*. In contrast, more than half of the friend pairs share 100 common friends in *Friends Group*. Although the common interests between two users are very sparse, the maximum Common Interest Degree of *Friends Group* reaches 31 which doubles that of 14 in *Random Group*. The average Common Interest Degree of *Friends Group* and *Random Group* are 0.42

and 0.21 respectively.

O2.2: The average interest correlation of the top K interests (shown in figure 5.2(c)) and individual interest correlation of the top 100 interests (shown in figure 5.2(d)) both are higher in *Friends Group* than in *Random Group*.

I2: We suppose that a user is more likely to find content for another user if they have higher similarity. Therefore, we obtain the following two inferences.

I2.1: As the Common Friend/Interest Degree and the Interest Correlations are higher in *Friends Group* than in *Random Group*, friends present a higher similarity than strangers. Hence, we infer that it might be easier to discover content for a user via her friends than through strangers.

I2.2: We also conjecture that a user might be more likely to find a content from the friends with higher Common Friend/Interest Degree.

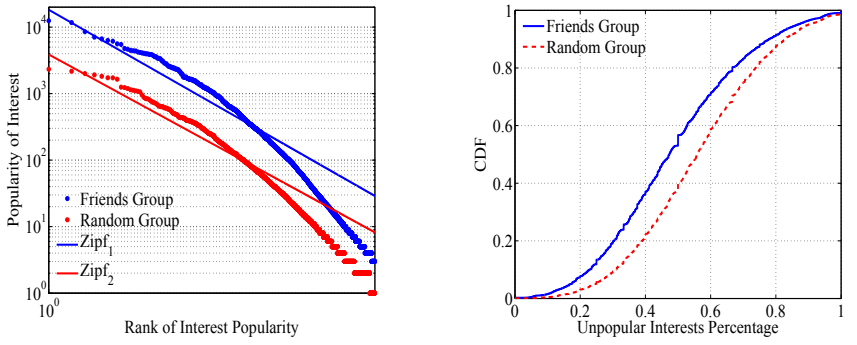
5.2.3 Interest Popularity Distribution

An interest is considered as a popular interest if many users state it as an interest on Facebook. In this section, we test how many percentages of interests are popular to most of the users in the two groups. For each user, we also study the percentage of unpopular interests that she presents. This study would reveal how important it is to take into account content discovery regarding users' personal interests (both popular and unpopular ones).

M3: we look into Interest Popularity Distribution and the percentage of unpopular interests among each users' personal interests. Interest Popularity Distribution is computed to estimate how popular the interests are. We also look into the Percentage of Users' Unpopular Interests.

M3.1: We define the popularity of an interest as the number of its fans. The interest is more popular if it attracts more fans. We rank all the interests based on their popularity. Figure 5.3(a) shows the interest popularity distribution in the log-log scale.

M3.2: We assume the top 500 interests are popular interests and the rests are unpopular ones. Figure 5.3(b) displays the CDF by the percentage of users' unpopular interests.



(a) Interests popularity distribution

(b) CDF of percentage of users' unpopular interests

Figure 5.3: Interests distribution

O3: We observe that the interest popularity distribution is very skewed - most of the interests are unpopular which only attracts quite few users. In addition, we also observe that almost 50% of a user's interests are not popular in both groups. Some details are reported as follows:

O3.1: Figure 5.3(a) shows that the Interest Popularity Distribution of both groups shapes in Zipf lines. In the *Friends Group*, only around 23.4% of users prefer the top One interest and the

500th interest attracts only 0.35% of users. While in the *Random Group*, the top One and the 500th interests are preferred only by 13.2% and 0.4% of users respectively. Generally speaking, most of interests are preferred by only a small number of users.

O3.2: Figure 5.3(b) reveals that, for more than 45% of users in the *Friends Group*, half of their interests are unpopular; while for nearly 75% of users when it comes to the *Random Group*.

I3: From the perspective of the interests in a group, most of interests are not popular; whereas from the perspective of users, unpopular interests account for around half of their interests on average. Therefore, we state that only improving discovery of popular content cannot satisfy users' requirements. We have to take into account users' unpopular interests meanwhile.

5.2.4 Analysis Summary

We briefly summarize the main inferences which might guide the design of content discovery mechanism as follows:

Summary(S)1: Concerning about content discovery for users' personal interests is very important for satisfying users' P2P experiences (see **I3**);

S2: A user discovers her personal interests more easily from her friends than from strangers (see **I2.1**);

S3: A user is more likely to find content from her friends with more friends and interests (see **I1.2**);

S4: The friends who share more common friends/interests would achieve content discovery with higher possibilities (see **I2.2**).

5.3 Social-Based Content Discovery Mechanism

Content discovery problem is normally approached by finding paths from a starting node to target nodes that store the queried content in a network. Our idea is to cast this problem as a task that a sender (starting node or any mediator node) ranks all candidate nodes and selects top-ranked ones as the next hop (i.e., receivers) on the paths. We aim to assign higher scores to the nodes that more likely reply to the sender's query.

Grounded on both the analytical results from the previous section and the idea of selecting receivers, we attempt to achieve content discovery with high performance for users' personal interests (see **S1**). First, we build up a social P2P network model which leads to the content discovery for a user via her friends (see **S2**). In this model, the nodes connect to their friends by using social relationships in social networks and weight their friends based on two types social attributes - **Knowledge** (refer to **S3**) and **Similarity** (refer to **S4**). On top of this model, we introduce a Top *K social-DRWR-P2P* search algorithm to select receivers for each sender. This algorithm chooses a user's friends that have more knowledge and share higher similarity with this user. The next two subsections explain the social P2P network model and search algorithm in details.

5.3.1 Social P2P Network Model

In order to construct the social P2P network model, shown in figure 5.4, we project users' social information on social networks into the corresponding nodes in a P2P network. The nodes thus inherit the users' basic profiles, friends' lists, and interests' lists. The nodes connect to each other if they are friends on social networks. Therefore, we define the social P2P network model as a weighted directed graph $G = \{V, S, E\}$, where V is the set of nodes in the network model; S is the set

of nodes' social information inherited from social networks; and $E \subseteq V \times V$ is the set of weighted links which are determined by users' friendships. In this graph, each node estimates the weights of its links with respect to the corresponding friends' probabilities of discovering content, namely friends' content discovery weights. In the following sections, we discuss the calculation of friends' content discovery weights and feasibility of the social P2P network model.

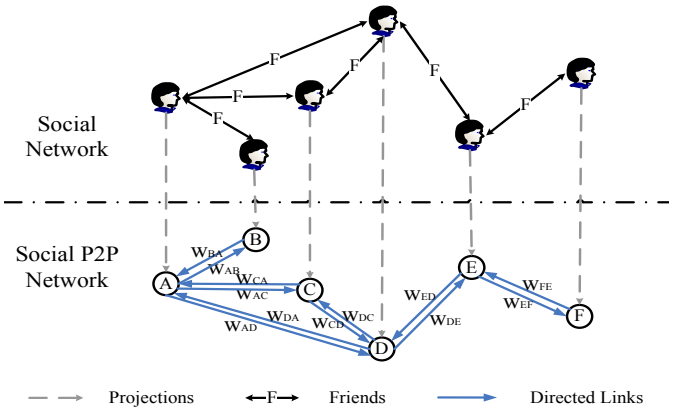


Figure 5.4: Social P2P network model

5.3.1.1 Friend's Content Discovery Weight

Referring to **S3** and **S4** in Section 5.2.4, we estimate friends' content discovery weight by two ways accordingly. Thus we obtain two types of social features, namely **Knowledge features** and **Similarity features**, which are detailed as follows.

Knowledge features: we define a node's knowledge features by the amount of resources (i.e., knowledge degree) with respect to various users' social attributes. In particular, each social attribute is associated with one knowledge feature. For example, regarding the social attribute of a user's friend (interest) list, we compute friend (interest) degree by counting the number of friends (interests). We expect that the friends with more knowledge would be more likely to reply the node's content query (refer to **S3**). Therefore, we assign higher weights to the friends with more knowledge.

Knowledge weight matrix: to explain how to weight friends by their knowledge, we consider a node i and its r friends. Specifically, assuming n types of knowledge features are employed, for one of its friend j, the node i denotes all the quantified knowledge degrees as $D_{ij}^{(K)} = (d_{ij}^{k1}, d_{ij}^{k2}, ..., d_{ij}^{kn})$. Similarly, for all of its friends, the node i generates a knowledge degree matrix $(D_i^{(K)})$. $D_i^{(K)}$ is a $r \times n$ matrix, in which each row stands for the knowledge degrees of one friend over n knowledge features; and each column represents the knowledge degrees on one particular knowledge feature by different friends. Using the logistic way, we normalized the xth knowledge degree of friend j by: $norm_x(d_{ij}^{kx}) = \frac{1-exp(-d_{ij}^{kx}/\theta^x)}{1+exp(-d_{ij}^{kx}/\theta^x)}$, where θ^x is a regularization parameter by the xth knowledge degree. Eventually, the node i calculates knowledge weights for all its friends by normalizing the matrix of knowledge degree $(D_i^{(K)})$, denoted as:

$$W_i^{(K)} = norm(D^{(k)}) = \begin{bmatrix} norm_1(d_{i1}^{(k1)}) & norm_2(d_{i1}^{(k2)}) & \dots & norm_n(d_{i1}^{(kn)}) \\ norm_1(d_{i2}^{(k1)}) & norm_2(d_{i2}^{(k2)}) & \dots & norm_n(d_{i2}^{(kn)}) \\ \vdots & \vdots & \ddots & \vdots \\ norm_1(d_{ir}^{(k1)}) & norm_2(d_{ir}^{(k2)}) & \dots & norm_n(d_{ir}^{(kn)}) \end{bmatrix} \tag{5.1}$$

Similarity features: we compute similarity of two users with respect to their social attributes as similarity features. Such features measure how much two users are similar regarding the corresponding attributes. For example, we can derive friend (interest) similarities between a user and her friends by employing their social attributes of friend (interest) list. We conjecture that the friends who have higher similarity with the node would be more likely to reply a satisfactory content (refer to **S4**). Hence, such friends should be assigned with larger weights too.

Similarity weight matrix: suppose that we discuss m types of similarity features, then node i's similarity features are expressed by a vector, $F_i^{(S)} = (f_i^{s1}, f_i^{s2}, ..., f_i^{sm})$. Regarding each feature, node i computes the similarity with its friend by the cosine distance. In regard of the lth feature, the similarity weight between node i and its friend j can be calculated by:

$$w_{ij}^{(sl)} = \frac{f_i^{(sl)} \cdot f_j^{(sl)}}{\|f_i^{(sl)}\| \cdot \|f_j^{(sl)}\|}$$

For friend j, node i records their similarity weights over all the m features by a similarity weight vector, i.e., $w_{ij}^{(S)} = (w_{ij}^{s1}, w_{ij}^{s2}, ..., w_{ij}^{sm})$. Similarly, node i calculates the similarity weight vectors for all of its friends (r in total) and further integrates them into a similarity weight matrix. Thus, the similarity weight matrix generated by node i equals:

$$W_i^{(S)} = \begin{bmatrix} w_{i1}^{(S)} \\ w_{i2}^{(S)} \\ \vdots \\ w_{ir}^{(S)} \end{bmatrix} = \begin{bmatrix} w_{i1}^{(s1)} & w_{i1}^{(s2)} & \dots & w_{i1}^{(sm)} \\ w_{i2}^{(s1)} & w_{i2}^{(s2)} & \dots & w_{i2}^{(sm)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{ir}^{(s1)} & w_{ir}^{(s2)} & \dots & w_{ir}^{(sm)} \end{bmatrix} \tag{5.2}$$

Integration of Knowledge and Similarity: at last, node i computes the integrative weights for its friends (i.e., friends' content discovery weights, $W_i^{(KS)}$) by incorporating their knowledge weights and similarity weights as follows:

$$W_i^{(KS)} = W_i^{(K)} \cdot \boldsymbol{\alpha} + W_i^{(S)} \cdot \boldsymbol{\beta} \tag{5.3}$$

where $\boldsymbol{\alpha} = [\alpha^{k1}\alpha^{k2}...\alpha^{kn}]^\top$ and $\boldsymbol{\beta} = [\beta^{s1}\beta^{s2}...\beta^{sm}]^\top$, are the parameters of the model, and $\alpha^{k1} + \alpha^{k2} + ... + \alpha^{kn} + \beta^{s1} + \beta^{s2} + ... + \beta^{sm} = 1$.

As different attributes might affect content discovery performance at varying degrees, we expect to find out a set of optimal feature parameters according to the feature's influence on performance of content discovery. The parameters optimization problem is discussed in Section 5.3.2.3.

5.3.1.2 Discussions of Social P2P Network Model

It is reasonable to map users' social information from social networks onto the nodes in a P2P network. Nowadays, a huge number of Internet users apply P2P platforms to share files, and meanwhile communicate on various social networks. For example, Bob often watches movies on PPStream,

while he also claims his favorite movies on Facebook. Although Bob's favorite movies are not explicitly claimed on PPStream, it is reasonable that PPStream uses these information to enhance Bob's experience. We further discuss the practicality of this model in Section 5.6.

In addition, there are two reasons that we set up a social P2P network model by linking nodes via friendships. First, we are inspired by the analytical result that a user is more likely to find her interests through friends than strangers. Second, considering the plenty of nodes in a P2P network, it is resource-consuming and time-wasting to compute links' weights and rank them. The social P2P network model considerably scales down a sender's candidate nodes to its friends and makes it lightweight to run a ranking algorithm.

5.3.2 Top K *Social-DRWR-P2P* Algorithm

In this section, we propose a Top K *social-DRWR-P2P* algorithm to further select a subset of friends over the social P2P network model. First we introduce the basic algorithm of Random Walking with Restart (RWR). Then we present a modified version of RWR, namely Distributed RWR (DRWR), which could be applied distributedly in our social P2P network model. DRWR biases the friends who are more likely to reply to the queries with higher scores. In order to score friends properly, we discuss the model parameter optimization problem subsequently. We eventually present the Top K *social-DRWR-P2P* mechanism and give an example of receiver selection.

5.3.2.1 Random Walk with Restart

Given a weighted graph $G(V,E)$, RWR performs walks starting from a node s to other nodes by following the probabilities of the edges that are proportional to their weights at each step. We assume that each step of a random walker is independent of its previous moves, thus we could employ a Markov chain to describe the path that the random walker visited. We denote the state that a random walker is visiting node i at step t as $i = i(t)$. The transition probability of a random walker shifting from state $i = i(t)$ to the next state $j = j(t+1)$ is:

$$p_{sj}(t) = p(j(t+1)|s(t)) \tag{5.4}$$

$\boldsymbol{p}(t) = \{p_{sj}(t)\}$ is called the transition probability vector at step t for all nodes. In addition, at each step we also consider a probability, namely the self-transition probability δ, of making the random walker go back to the starter s. We calculate the shifting rate by using the following equation recursively:

$$\boldsymbol{p}(t+1) = (1-\delta)\boldsymbol{A}\boldsymbol{p}(t) + \delta\boldsymbol{q} \tag{5.5}$$

In this equation, $\boldsymbol{q}$ is a vector where the elements equal 0 except for the one that corresponds to the initial node being set to 1. $\boldsymbol{A}$ is a matrix in which the elements stand for the state transition probabilities between two nodes. If i and j are disconnected to each other, $a_{ij} = 0$; and otherwise $a_{ij} = w_{ij}/w_{(i\cdot)}$ where $w_{(i\cdot)} = \sum_{j=1}^{n} w_{ij}$. w_{ij} is the weight that node i assigns to its friend j, calculated by equation 5.3. Therefore, the matrix A is computed as:

$$A = W^{(K)} \cdot \boldsymbol{\alpha} + W^{(S)} \cdot \boldsymbol{\beta} \tag{5.6}$$

Since the random walker's visiting pattern is a Markov process, the transition probability vector can converge after a number of steps l. Finally we obtain $p(l)$ as a stationary measure of the shifting rate.

5.3.2.2 Distributed RWR (DRWR)

Each node in the social P2P network constructs a <FRIEND, WEIGHT> table (denoted as $Ti < F,W >$) by computing its friends' weights and exchanges it with their friends. Each node, from its friends' $Ti < F,W >$, picks out the entries that reflect the latent relationship among its friends. By merging the selected entries from all its friends' $Ti < F,W >$, the node builds up a mixed <FRIEND, WEIGHT> table called $MTn < F,W >$ and calculates transition matrix $\boldsymbol{A}$. Finally, the node conducts a local random walk over all its friends and computes a stable transition probability for each friend as its score, by using the Eq.(5.5). DRWR method could extract the latent friendship behind a node to bias its friends' scores.

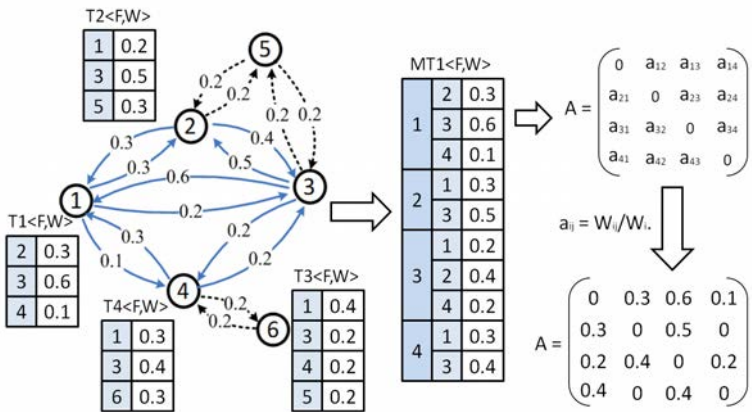


Figure 5.5: Distributed random walk

To further explain the DRWR method, we illustrate how node 1 in figure 5.5 assigns scores to its friends as an instance. Node 1 has three friends of nodes 2, 3 and 4 where node 3 connects to nodes 2 and 4 as well. We depict the links between two nodes in solid lines if both of them are either node 1 or its friends; while use dashed lines to represent the other links. The numbers on the links represent the friend content discovery weight. After exchanging $Ti < F,W >$, node 1 filters the weights of node 5 from $T2 < F,W >$ and $T3 < F,W >$. It also removes the weight of node 6 from $T4 < F,W >$. Node 1 obtains $MT1 < F,W >$ by means of filtering and merging all collected $Ti < F,W >$, as shown in the middle of figure 5.5. At the right side of this figure, we depict the initial transition matrix $\boldsymbol{A}$ on node 1. Node 1 computes the scores by solving Eq.5.5.

5.3.2.3 Parameters Optimization

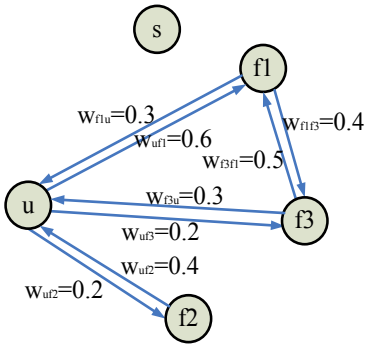
As we mentioned in Section 5.3.1, different information attributes affect content discovery performance at varying degrees. Hence we expect to find out a set of optimal feature parameters for the calculation of nodes' weight and finally to assign proper scores to friends by using DRWR. To address the problem, we begin with a sender s and divide all its friends into two subsets, denoted as F_k and F_r. We expect that the subset of F_k is comprised of the friends from which the sender could find the queried content with higher probabilities; while F_r consists of the friends of lower probabilities for content discovery. Therefore, we aim to find out an optimal parameter set for features that give the friends in F_k greater scores than those in F_r. We denote the parameter vector as $\boldsymbol{a}$ and define the optimization problem as:

$$\min_{\boldsymbol{a}} F(\boldsymbol{a}) = \| \boldsymbol{a} \|^2 + \lambda \sum_{k \in F_k, r \in F_r} h(p_r - p_k) \tag{5.7}$$

where λ is a regularization parameter and $h(.)$ generates a non-negative penalty which $h(.) = 0$ as $p_r < p_k$ while $h(.) > 0$ as $p_r > p_k$. To obtain the optimal parameters set, we exploit the gradient based optimization approach to minimizing the loss value [151] (Appendix A offers more details about parameter optimization.)

5.3.2.4 Top *K social-DRWR-P2P* Search Algorithm

In this section, we summarize the top *K social-DRWR-P2P* search algorithm: First, a node constructs connections based on its friendships presented by the corresponding user in social network. Then, the node leverages numerous features - namely friends' knowledge and similarity - to assign weights to its friends. By exploiting the DRWR algorithm, the node computes stable scores for its friends. Eventually, the node ranks all its friends based on their scores and selects the top K friends from the ranking list to forward queries.



R	Weights	DRWR Scores
f1	0.6	1
f3	0.2	0.669
f2	0.2	0.205
s	0	0

Figure 5.6: An example of top *K social-DRWR-P2P* search algorithm

An example of Top *K social-DRWR-P2P* Search is illustrated in Figure 5.6. First, every node connects with their friends and estimates its friends' weights based on their similarity and knowledge. In this example, user u connects to its friends $f1$, $f2$ and $f3$ and measures their weights (0.6, 0.2 and 0.2 respectively) based on friends' similarity and knowledge from u's own perspective. Similarly, $f1$, $f2$ and $f3$ also estimate weights for their friends. User u does not link to the stranger s and weights s as zero. Then, u runs DRWR algorithm to score each of its friends, shown in the right table.

Particularly, we note that the final scores assigned by DRWR are not as the same as the initial weights. For instance, from u's own perspective, $f2$ and $f3$ have the same weight. However, $f3$ should be assigned a higher score than $f2$ intuitively since $f3$ is also a friend of u's friend ($f1$). This relationship makes u, $f1$ and $f3$ much closer to each other and raises the content discovery probabilities of $f1$ and $f3$. It is the DRWR algorithm that explores the latent friendship between friends $f1$ and $f3$ to increase their scores. Finally, we select top K friends from the friends ranking list.

Furthermore, the proposed mechanism is flexible with the changeable knowledge and similarity features. Different feature parameters are assigned according to specific applications. In addition, we notice that the complexity of the algorithm is $O(l)$ defined by the convergent steps.

ALGORITHM 1: Top *K social-DRWR-P2P* search

Input: Friends' information from OSN:;
The set of friends' list of a node;
The set of friends' feature;
The number of selected friends: K;
Output: Top K friends of the node;
Initialize unstructured social_P2P network;
for $f^k \in KnowledgeFeatures(F^{(K)})$ **do**
  Assign the weight of feature f^k to each friend of the node(Equation 5.1);
end
for $f^s \in SimilarityFeatures(F^{(S)})$ **do**
  Assign weight of feature f^s to each friend of the node(Equation 5.2);
end
Combine all factored features' weight (Equation 5.3)
Iteration: Run DRWR until probability vector $\boldsymbol{p}$ converges. $l = 0$;
for $\boldsymbol{p}$ *is not convergent* **do**
  Calculate stable transition probability for each friend (Equation 5.5);
  $l++$;
end
Order friends based on friends' scores;
Selected Top K friends of the node;

5.4 Experiments Setup

We use the two Facebook data sets to evaluate the proposed the mechanism. The friendships are used to connect nodes in the social P2P network, and the information of a user's friends is applied to estimate the content discovery weights. In this section, we first introduce the experiment method and performance metrics. Then, we describe the parameter setup in the proposed social P2P network model.

5.4.1 Experiment Design

5.4.1.1 Assumption and Evaluation Strategies

Receivers (any mediator nodes or the target nodes) in the experiments store a set of content so as to reply to the queries from the starting node. Facebook supports user-generated interests explicitly. Here we assume the receivers store their favorite Movie, Music, Book, Game and TV series, or know how to find out their interests even if they do not store them on their disk. And then it is plausible to assume that a receiver's interest list on Facebook works as her content list.

From the perspective of a normal user (starting node), two kinds of interests are desirable: the user's personal interests and the most popular interests. Our evaluations are therefore composed of two parts: personal interests searching and popular interests searching. In personal interests searching, we assume that the starting node looks for all its interests from others. In popular interests searching, top 500 interests in each group are considered as its popular interests, and the starting node searches all the popular interests.

5.4.1.2 Comparison

We compare the newly proposed content discovery mechanisms (i.e., *social-DRWR-P2P*) to *social-P2P* and *traditional-P2P*.

- *social-DRWR-P2P:* we first project the information of users in *Friends Group* to the nodes in P2P network one by one and generate the *social-P2P* network topology by following the social p2p network model introduced in Section 5.3.1. We run Top K *social-DRWR-P2P* algorithm and launch queries to the selected top K nodes over this network topology.

- *social-P2P:* we use the same *social-P2P* network topology as *social-DRWR-P2P* mechanism does. However, the content discovery queries are forwarded to K randomly selected friends, instead of the top K friends selected by *social-DRWR-P2P*.

- *traditional-P2P:* we map the users' information in the *Random Group* to the nodes in the P2P network and then a node selects K users at random to send queries.

We launch one-hop and two-hop searching by forwarding K queries at each sender.

5.4.2 Performance Metrics

In each content discovery procedure, a node (i.e., sender) sends the content discovery queries to K selected nodes (i.e., receiver[5]), and in turn H nodes (i.e., replier) among them reply. In addition, we refer the node that stores the queried content as a storer and denote the total number of storers as C. Then we define the following four metrics to evaluate our proposed method:

Hits: Hits is defined as the average number of replies during content discovery procedures (i.e. H). Intuitively, it relates to the selected number (K) of receivers: a sender might get more replies while it sends queries to more receivers.

Query Success Rate (QSR): QSR equals the fraction of the number of replies to the number of receivers (i.e., $QSR = H/K$). Although increasing the number of receivers might lead to more Hits, it costs more network resources (e.g., bandwidth). To some extent, over-query could even lead to network congestion and lower network performance. Hence, Hits alone is not enough for performance evaluation. Given two mechanisms which achieve the same Hits, the one with a higher QSR performs more efficiently.

Search Success Rate (SSR): We consider a content discovery procedure to be successful as long as the sender receives a reply at least from the receivers. SSR is a metric for estimating the success rate of procedures. We run M procedures in total and S of them are successful. Thus, we calculate SSR by dividing the number of successful procedures by the total number of procedures (i.e., $SSR = S/M$). Note that different P2P applications have different requirements in content discovery: some of them are only interested in finding one single copy of content, while others look for as many copies as possible. Therefore, the former applications probably do not concern about QSR, since SSR is a very important metric for them. In contrast, QSR is meaningful for the latter applications.

Recall: Recall is computed as the number of repliers divided by the total number of storers (i.e., $Recall = H/C$). Recall reflects the capacity of a mechanism in terms of completely retrieving. If two mechanisms achieve the same Hits and QSR / SSR, the one that reaches higher Recall presents better performance.

[5] In *social-DRWR-P2P*, the receivers are the top K friends; in *social-P2P*, the receivers stand for the random selected friends; in *traditional P2P*, the receivers represent for the totally random selected users

5.4.3 Parameters Setup

As described in Section 5.3, the proposed *social-DRWR-P2P* algorithm applies two ways to quantify users' social attributes, which respectively produce knowledge features and similarity features. In our experiments, we employ two social attributes which are users' friends list and interests list. Drawing on equations 5.1 and 5.2, we obtain the normalized friend degree and interest degree as knowledge features; and we compute friend similarity and interest similarity as similarity features.

5.5 Performance Evaluation

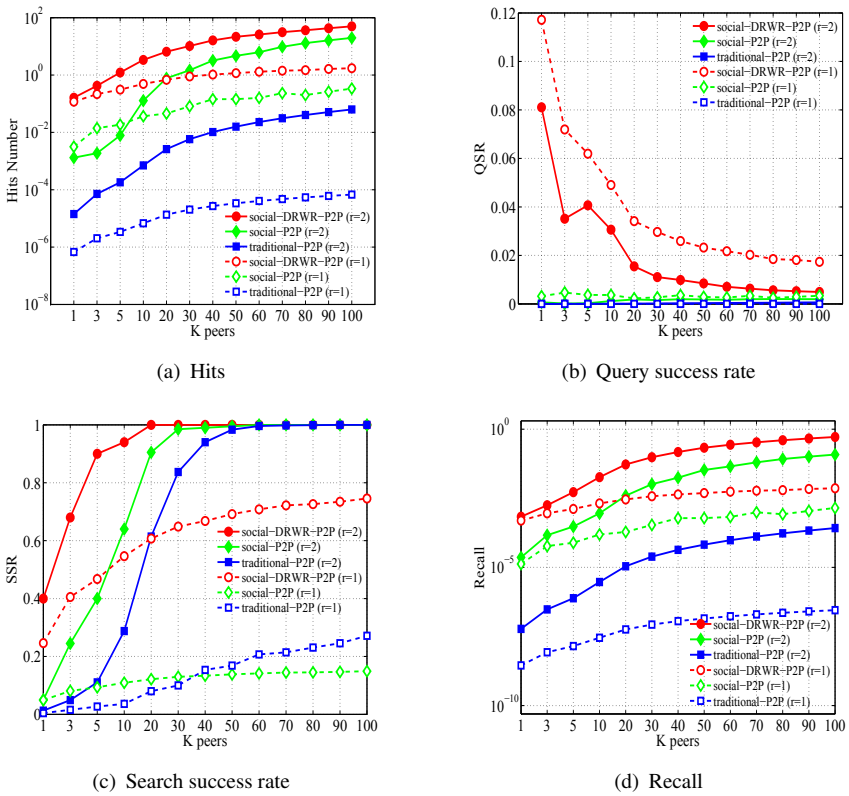
In this section, we compare the performance of *traditional-P2P*, *social-P2P* and *social-DRWR-P2P* with respect to personal interests searching and popular interests searching respectively. The results indicate that *social-DRWR-P2P* is superior to the other algorithms not only for discovering popular interests but also for nodes' personal interests.

5.5.1 Personal Interests Searching

To evaluate the discovery of users' personal interests, a starting node generates queries for all its personal interests and a receiver replies as long as it stores the queried interests. Figure 5.7 sequentially plots the personal interests search results of the Hits, QSR, SSR and Recall achieved by the three compared algorithms. The vertical axes are the values of the aforementioned four metrics and the horizontal axes represent the number of receivers. In the figure, K only represents the number of receivers to which each sender forward queries. Therefore, the total number of receivers for two-hop search is $K + K^2$ corresponding to K at the horizontal axes in figure 5.7 and 5.8. We perform the experiments with K being $[1, 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ respectively.

Figure 5.7(a) shows the average Hits. It is obvious that the values of Hits are getting higher as the number of receivers (K) increases. In cases of both one-hop and two-hop, *social-P2P* gains higher Hits than *traditional-P2P*. This implies that friends perform better than randomly selected nodes for personal interests searching. Compared with *social-P2P*, *social-DRWR-P2P* achieves even higher Hits. This observation indicates that friends with a higher similarity and more knowledge are more likely to find personal interests. Furthermore, in the one-hop experiments, the Hits of *social-DRWR-P2P* exceeds 1 when the receivers are more than 40; while the Hits of the other two mechanisms only reach 0.14 and 0.00003. In the two-hop estimations, *social-DRWR-P2P* can obtain 1.22 replies on average by sending queries within 5 receivers at each sender; however, *social-P2P* and *traditional-P2P* receive only 0.008 and 0.0002 replies respectively under the same condition. The results indicate that two-hop search costs fewer queries than one-hop search to achieve the same performance of Hits. For instance, to guarantee one Hits, a starting node, forwarding 5 queries at each sender in two-hop search, sends 30 queries in total; compared with 40 queries in one-hop search.

Figure 5.7(b) reveals that *social-DRWR-P2P* gains much higher QSR than *traditional-P2P* and *social-P2P*. Additionally, we observe that, for *social-P2P* and *traditional-P2P*, the QSR changes little in a broad range of K values, especially in one-hop searching; however, the QSR of *social-DRWR-P2P* decreases obviously as K increases. In other words, the efficiency of *social-DRWR-P2P* drops while more friends with lower weight (i.e., K increases) are requested to. These observations reflect that the friends of more knowledge and similarity benefit more for content discovery. Combining the results from both Hits (figure 5.7(a)) and QSR (figure 5.7(b)), we note that when K is



(a) Hits (b) Query success rate

(c) Search success rate (d) Recall

Figure 5.7: Performance of personal interests searching

between 5 and 20, *social-DRWR-P2P* can obtain a good Hits (between 1.22 to 6.5)and a good QSR (between 0.041 to 0.015) within two-hop search.

From figure 5.7(c), we can see that the proposed mechanism also outperforms others in terms of the SSR. The SSR of *social-DRWR-P2P* achieves 100% with selecting 20 receivers at each sender in two-hop search. This means that *social-DRWR-P2P* can guarantee its success by two-hop search with sending 420 queries (*i.e.*,K=20) in total. To accomplish the same performance, *social-P2P* needs to query 3660 receivers (i.e., $K = 60$) and *traditional P2P* queries to 8190 receivers (i.e., $K = 90$) at least. In other words, in order to guarantee a successful search, *social-DRWR-P2P* saves almost 8 and 18 times queries compared to *social-P2P* and *traditional P2P*.

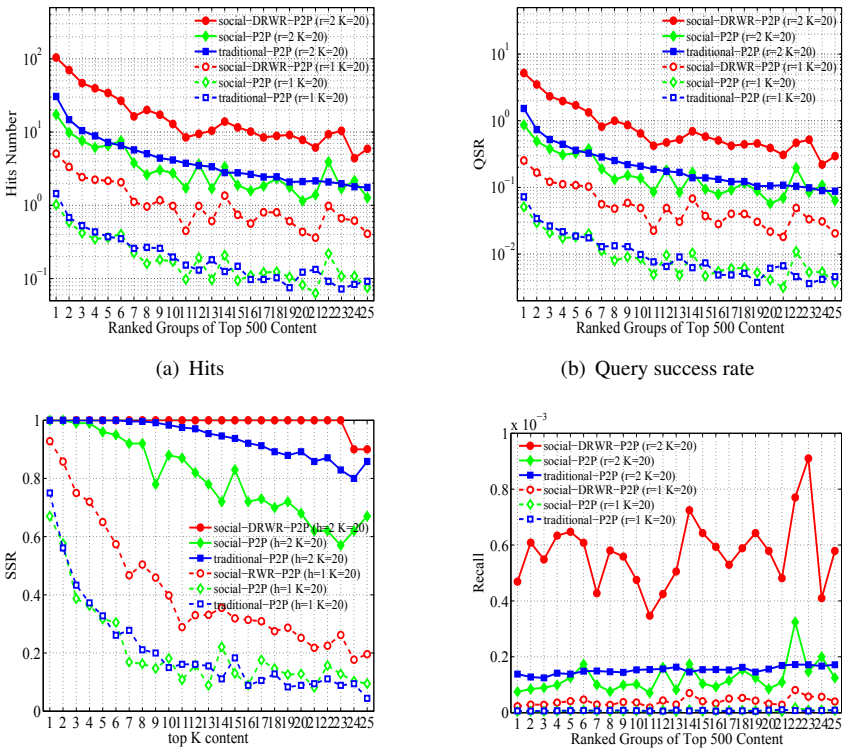
Figure 5.7(d) compares the completely retrieving capacity of the three mechanisms. In the best cases, *social-DRWR-P2P* can find out 0.73% and 52.12% of storers to reply queries in one-hop search and two-hop search respectively. Meanwhile, the *social-P2P* only locates 0.14% and 12.05% of storers, and the *traditional-P2P* explores about 3×10^{-5}% and 0.26% comparatively. On average, *social-DRWR-P2P* improves the percentage of retrieved storers by nearly 11 times in one-hop and 19 times in two-hop compared to *social-P2P*.

To summarize content discovery for personal interests, we suggest applying two-hop *social-DRWR-P2P* with selecting top 20 receivers at each sender. In this case, *social-DRWR-P2P* could guarantee a 100% successful content discovery. Also it achieves suitable Hits (6.5) with acceptable QSR (0.015).

5.5.2 Popular Interests Searching

In order to validate the performance of the three algorithms in terms of popular interests searching, we first rank all the interests by their popularity. Then we group the successive 20 interests from high to low in the ranking list into a bucket and calculate the average value for all the metrics. We consider the top 500 interests as the popular interests and generate 25 buckets. Figure 5.8 shows the one-hop and two-hop evaluations with $K = 20$ for *social-DRWR-P2P*, *social-P2P* and *traditional-P2P*.

We can see that, in the case of popular interest discovery, *social-DRWR-P2P* also outperforms *social-P2P* and *traditional-P2P* and achieves better performance of Hits, QSR, SSR and Recall under the same conditions. We account for friends' knowledge amount as a factor when we rank friends in *social-DRWR-P2P*. Therefore, the observations may respond to the fact that the selected receivers with higher scores can provide more content which also contain many popular content. However, *social-P2P* does not perform better than *traditional-P2P* method for popular interests searching, which implies that the algorithm merely involving the friendship does not benefit popular interests searching obviously.



(a) Hits (b) Query success rate

(c) Search success rate (d) Recall

Figure 5.8: Performance of popular interests searching. We group the top content by size of 20, thus the buckets are:[T1-T20],[T21-T40],...,[T481-T500].

We also observe that, in general, the content discovery queries for the interests in the higher position in a ranking list receive replies with higher probabilities as well as higher efficiency. This might suggest that for popular interests searching we could downsize the number of receivers to some extent in order to obtain enough Hits and reduce the cost at the same time; and contrarily, we would have to send more queries to achieve similar performance for searching unpopular interests.

In addition, we notice that the value of recall does not decrease as the ranking goes down. That is to say, the capacity for retrieving interests can maintain a certain level no matter how popular the interests are.

5.5.3 Result Discussions

The results obtained in this section provide a number of interesting insights that we summarize as follows:
(1) Due to the large number of available resources for the popular interests, retrieving such interests is a relatively easier task. Additionally users present many unpopular interests in general (see Section 5.2). Therefore, a good content discovery solution should be characterized by its twofold abilities of finding popular interests and personal interests. The experiments reveal that our proposed *social-DRWR-P2P* significantly improves the performance of content discovery not only for popular interests but also for personal interests.
(2) We also notice that, for popular interests searching, *social-P2P* which merely considers the friendship among nodes does not show any advantage over *traditional-P2P*. This just indicates that the two aspects of our proposed mechanism - the social P2P network model and the Top *K social-DRWR-P2P* Search Algorithm - are both necessary in order to improve content discovery.
(3) It has been demonstrated that, for a certain number of queries, the proposed *social-DRWR-P2P* might perform better within two-hop search than one-hop search. For instance, if we query 110 friends within one-hop, the sender selects receivers including the relatively low ranking ones among all its friends. However, if the same amount of queries are issued within two-hop, the queries are sent to the 10 highest ranked friends and sequentially forwarded to 10 highest ranked friends of them.
(4) We can state that the friends with a higher similarity and more knowledge are more likely to reply the content from two perspectives: (i) *social-DRWR-P2P*, which selects the receivers with friends of higher weight, performs better than *social-P2P* (i.e., randomly select friends); (ii) the QSR of *social-DRWR-P2P* decreases with involving more friends of lower weights (see figure 5.7(b)). Furthermore, we devise an experiment to verify this statement:

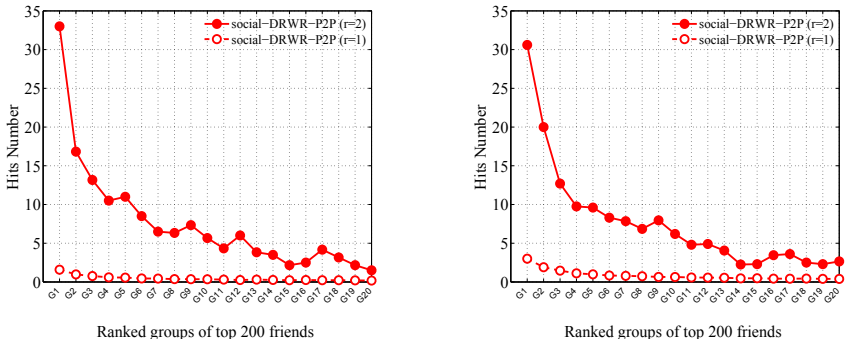
- With the ranked friends list generated by *social-DRWR-P2P*, we cluster each 10 successive friends from top to bottom into a group and compare the average Hits of each group. That is to say, the top 10 friends are clustered into group 1, and the next successive 10 friends (i.e., top 11th to top 20th) into group 2. In this way, we generate 20 ordered groups by the top 200 friends. Note that the friends in nth group have higher scores than friends in $n+1$th group. Figure 5.9 shows that, both for personal interests searching and popular interests searching, the friends with more knowledge and higher similarity can achieve better performance.

5.6 Discussion

In this section, we discuss and explain three practical issues of the proposed mechanism.

5.6.1 Feasibility of Social P2P Model

In this chapter, we project user social information into a P2P network to build up the social P2P network model. This model is the basis for the proposed content discovery algorithm. Therefore the feasibility of the model determines the practicability of the proposed systems. The core issue here



(a) Personal interests searching (b) Popular interests searching

Figure 5.9: Hits number of *social-DRWR-P2P*

is whether or not it is possible to set up a real social P2P sharing platform (i.e. a P2P network with users' social information).

For enlarging the influence and user volume, many existing P2P applications recommend users to combine their P2P accounts with their social network accounts. For instance, when logging on PPstream, a user would receive a message of "Login with your Weibo (Chinese Twitter) account; Login with your Renren (Chinese Facebook) account; Login with your QQ (Chinese MSN) account". Even though we have no idea of what percentage of users would meet these requirements, we believe users would accept this recommendation if they could obtain better performance for their P2P experience. The existing unstructured P2P applications could directly follow the proposed mechanism to improve content discovery.

Existing P2P applications (e.g. PPstream, PPlive) encourage users to register on their platforms. They record users' basic profile like age, gender, education, etc., and track users' history of uploading, watching and storing. As long as allowing users to make friends and encourage them to communicate with each other on their platforms, these P2P applications could easily construct their own social P2P networks. The proposed mechanism can be integrated into exiting platforms.

As Facebook is currently the biggest OSN in the world, we leverage Facebook to extract users' social information, setup the social P2P model and conduct the experiments in this research. However, for the two above-mentioned reasons, our proposed method is practical not only for P2P platforms with user accounts associated with Facebook, but also for others with their own user social networks.

5.6.2 Effectiveness of Facebook Data set

Because of Facebook's privacy policy, we only crawl the public information from the public users from Facebook. Thus, one might doubt that the results of studies on Facebook and the data-based experiments are biased by the incomplete data sets. However, as shown in figure 3.1(c), 65% of users in the data sets present their friends and 53% of users show their interests to the public. From the point of studies, we have a considerable number of samples. In addition, we use two ways to collect information and present their generality in Section 3.2.2. From the perspective of data-based experiments, a node could probably achieve a better performance if it has more social information .

5.6.3 Selection of Social Features

To estimate friends' content discovery weights with respect to social attributes (in Section 5.3.1.1), we provide two ways by which we obtain knowledge features and similarity features respectively. We refer knowledge features as the quantifiable resources of a node; and regard similarity features as the metrics, which measure how much users are alike with respect to diverse attributes. In our opinion, this model can be flexibly extended to contain more relative social features regarding the available social information. For instance, age similarity might be a practical similarity feature, as in general younger generation of 1990s might present different tastes in movies or music, compared with middle-age people who were born in 1970s. In addition, the proposed algorithm, summing up the products of the features' values and the biased parameters (see Equation 5.3 in Section 5.3.1.1), seeks to achieve the best performance by taking overall advantage of the considered features.

5.7 Summary

In this chapter, we present a social P2P mechanism grounded on the real social network information. By linking nodes through their social friendship, we build up a social P2P network model; we weight the friendship regarding of knowledge features and similarity features. Based on this model, we further propose a content discover algorithm which selects a subset of friends by the modified version of the RWR algorithm (i.e., DRWR). This algorithm is able to explore the latent friendships among a node's friends. Although online social networks are mainly centralized nowadays, the social information that users generate and maintain can be exploited into a P2P environment. Besides, relying on a large data set with 500K Facebook user profiles, we conduct comprehensive experiments to evaluate our proposed method. The experiment results have demonstrated that our proposed approach is capable of improving content discovery in P2P not only for popular content but also for users' personal interests. In the future, we plan to extend the current solution by selecting friends regarding their social features as well as the features of the requested content, so as to make the mechanism more effective and intelligent. Besides, we will take into consideration more specific social features.

Chapter 6

Inspecting Interest Similarity: Prediction and Application

Contents

6.1 Introduction

With the evolution of OSNs, understanding to what extent two individuals are alike in their interests (i.e., interest similarity) has become a basic requirement for the organization and maintenance of vibrant OSNs. On the one hand, such information about users' interest similarity could be leveraged to support friend recommendation and social circle maintenance. For instance, the decision to recommend users who share many interests with each other to be friends could increase users' approval rate of recommendation, because people usually aggregate by their mutual interests [103]. On the other hand, knowing interest similarity between users also facilitates social applications and advertising. For example, instead of randomly hunting for clients, exploring those users with a high interest similarity with existing clients could efficiently enlarge client groups for application providers and businesses.

However, estimating interest similarity between two users is not a straight-forward issue since users do not always explicitly elaborate their interests. In our Facebook data set, 51.6% of users do not present any interests in their profiles; and among nine interest domains in the data set, except for movies, music and TV shows, less than a quarter of users reveal their interests in any of the other six interest domains (e.g., books, sports or games). Since such lack of users' interests occurs quite often in the real OSN environment, how to infer two users' interest similarity without complete information about their interests poses a challenge.

To deal with this problem, we investigate how two users' interest similarity relates to various social features in depth (e.g. profile overlap, geographic distance, and friend similarity) and further infer whether two users are alike/unalike in interest according to these learned relations. Existing studies have already demonstrated that friends share more interests than strangers [155] and verified that interest similarity strongly correlates to the trust between users [18]. However, the work to date has not address the issue of inferring users' interest similarity without complete information about users' interests. Furthermore, we carry out a comprehensive analysis on the correlations between users' interest similarity and diverse social features, and have unearthed additional relative factors that could enhance interest similarity prediction.

Particularly, we quantify interest similarity over an aggregation of user pairs by two metrics: **probability of sharing interest**, defined as the likelihood that two users have any mutual interests; and **degree of interest similarity**, which captures interest overlaps between two users based on the weighted cosine similarity. In addition, we extract social features (e.g. profile overlap, geographic distance, and friend similarity) from users' social information regarding three aspects: demographic information (age, gender, location, etc.), social relations (i.e., friendship), and obtainable users' interests. Specifically, we conduct the study in three interest domains, namely movies, music and TV shows.

We highlight our key findings captured from the wide variety of analysis — the homophily of interest similarity. Generally, homophily shows the level of homogeneity in people's social networks in relation to multiple sociodemographic, behavioral and intrapersonal characteristics [1]. Specifically, in this chapter, homophily

- reveals that people tend to be interested in the same movies, music and TV shows when they are similar in their demographic information, such as age, gender and location;
- implies that friends have higher interest similarity than strangers. Furthermore, the interest similarity increases if two users share more common friends;
- indicates that the individuals with a larger interest entropy are likely to share more interests with others. Note that we exploit interest entropy to quantify the characteristics of one user's

interests. A user's interest entropy is influenced by two factors: the total number of a user's interests and the popularity of these interests. The more interests a user presents, and the less popular the interests are, the more the user gains in interest entropy.

Based on the empirical studies, we propose a prediction model with a number of features (e.g. geographic distance, friend similarity and interest entropy). This prediction model can determine whether two users are similar or not in interest when one of the users does not provide his interests. The prediction result can be properly applied to various interest similarity based applications (e.g., recommendation system [156][157], friend prediction [155][104] and user evaluation system [158]). For instance, the model can help to address the *new user problem* in the typical collaborative recommendations [5][119]. Normally, a collaborative recommendation system recommends a user some items that are liked by the others with similar interests. Whereas, the recommendation may fail when it comes to a *new user u* not revealing his interests, as the system cannot determine which of its existing users may share interests with *u*. In this case, even without *u*'s interests, the proposed prediction model is able to find some existing users who are predicted being similar to *u* and recommend *u* some items according to their interests.

In summary, the main contributions of this chapter include:

- To the best of our knowledge, this is the first work to infer the interest similarity of two users where we do not know one of the user's interests. Owing to the frequent lack of users' interest in OSNs and the common requirement for applications of knowing the interest similarity between users, this research problem has a practical significance.
- We capture various social features depending on users' social information and investigate how interest similarity relates to these social features through a comprehensive perspective at a collective level. We uncover the homophily between these social features and users' interest similarity. Relying on a large data set crawled from Facebook, the analytical results can advance the collective knowledge of OSNs.
- We devise a practical interest similarity prediction model based on the learned social features, namely *InterestSim* model. We also introduce two baselines referred to *Friend* model and *DemoSim* model. These two baselines depends on users' friendships [5][159] and demographic similarity [119][160][161] respectively. The experiments show that *InterestSim* model outperforms *Friend* and *DemoSim* model by 12%-16% and 3%-4% respectively in terms of AUCs in different interest domains.
- We illustrate a use case where we leverage the proposed *InterestSim* model to practically address the *new user recommendation problem*. Compared with several state-of-the-art approaches, it turns out that our proposed *InterestSim* model can facilitate the *new user recommendation* with a higher precision.

6.2 Overview

We provide a brief overview to state the research problem, present an outline of a potential solution and introduce the empirical analysis framework, visualized in figure 6.1.

The goal of this chapter is *to estimate the interest similarity between two users without knowing one user's interest information*. To achieve this goal, we first distinguish two kinds of users, *Active Users* and *Passive Users*:

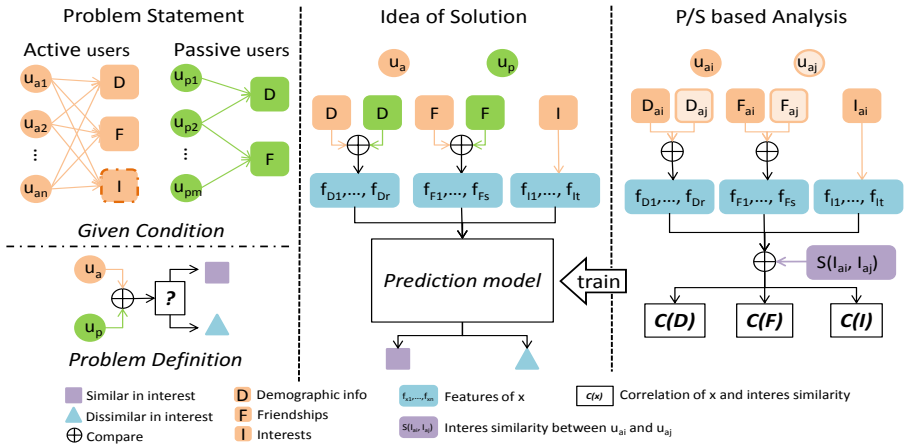


Figure 6.1: Overview of problem, proposed solution and the problem and solution (P/S) based analysis framework

- *Active Users* (i.e., u_a) explicitly present their demographic information (D), friendships (F) and interests (I), which can be denoted by a tuple of $u_a :< D_a, F_a, I_a >$;
- *Passive Users* (i.e., u_p) only report partial demographic information and/or friendships, but hide interests from the public; we denote a passive user as $u_p :< D_p, F_p >$.

On this basis, the fundamental problem becomes, given an active user u_a and a passive user u_p, to infer whether u_a and u_p are similar or dissimilar in interest. The problem also could be extended to select a subset of active users who probably share many interests with u_p, given a u_p and a set of active users (i.e., $C_{u_a} = \{u_a :< D_a, F_a, I_a >\}$).

Our solution for this problem is to train a prediction model which can infer the interest similarity between users relying on their obtainable social information. For instance, it might speculate that two users are more likely to share interests if they are friends. Consequently, we attempt to achieve the interest similarity prediction by two steps: (1) based on users' social information, we can capture several social features that may reflect users' interest similarity to some extent; and (2) based on the learned social features, we construct an interest similarity prediction model.

According to the proposed solution, the primary issue is to determine what specific social features correlate to the users' interest similarity. Therefore, we conduct extensive empirical analysis on interest similarity with respect to various social features derived from the users' social information. In particular, we perform the analysis through three perspectives:

- *Demographic-related features (f_D):* We extract the demographic features by comparing two users' demographic information (D) and investigate how they correlate to interest similarity. For example, we measure the geographic distance between users and examine how users' interest similarity varies regarding their geographic distance.
- *Friendship-related features (f_F):* We generate friendship features based on the friendships (F) of two users. For example, we define a feature of friend similarity by counting the mutual friends of two users and study its influence on interest similarity.
- *Interest-related feature (f_I):* Since we do not know the passive user's interests in the prediction problem, we tend to explore interest-related feature by capturing the interest characteristics

from the active user side (I). We expect that the users who exhibit certain characteristics on his interests would generally achieve a higher/lower interest similarity with others. In this chapter, we specially employ entropy to quantify a user's interests as the interest-related feature.

Furthermore, based on the learned social features, we exploit Support Vector Machines (SVM) [162][163] method to train the interest similarity prediction model.

6.3 Measurements for Interest Similarity

To study the properties of interest similarity among users, we define the measurement of interest similarity by two steps: (1) we first limit the computation of interest similarity between two users (i.e., a user pair); (2) we extend the computation to an aggregation of user pairs and obtain a measurement of collective interest similarity. The analysis regarding interest similarity in the following sections depends on the collective interest similarity. Consequently, we first introduce two ways to measure interest similarity between two users: **binary similarity** and **weighted cosine similarity**. Then, based on these two measurements, we define two metrics to evaluate interest similarity at an aggregated level, namely the **probability of sharing interests** and the **degree of interest similarity**.

6.3.1 Interest Similarity of Two Users

Binary similarity and weighted cosine similarity are the two measurements used to calculate interest similarity between two users. Note that user u's interests are denoted by an interest set I_u instead of a binary interest vector to avoid a very sparse interest vector.

Binary similarity measures whether or not two users are similar in terms of their interests. We assume that two users are similar in interest, as long as they have any mutual interests; otherwise, they are dissimilar, denoted as:

$$s_b(u,v) = \begin{cases} 1 & \text{if } I_{uv} \neq \varnothing \\ 0 & \text{if } I_{uv} = \varnothing \end{cases} \tag{6.1}$$

where I_{uv} represents the intersection of interests between user u and v. Binary similarity is defined to evaluate the probability of sharing interests.

Weighted cosine similarity estimates the extent to which two users are similar in interest. It is introduced by two steps. First, drawing on the general calculation of cosine similarity, the interest similarity between users u and v is then defined as the cosine distance between their interest sets: $s_c(u,v) = \frac{\|I_{uv}\|_1}{\|I_u\|_2 \cdot \|I_v\|_2}$ where $\|I_u\|_2 = \sqrt{l_u}$ (l_u is the number of interests of u) and $\|I_{uv}\|_1$ is the number of mutual interests of u and v. If either $l_u = 0$ or $l_v = 0$, $s_c(u,v)$ is undefined.

Moreover, as it seems easier for two users to share a very popular interest (e.g., the movie 'Harry Potter') than a rare one (e.g., the documentary 'La Dany'), we consider the interest similarity to be more significant if two users share a less popular interest. So, we introduce interest popularity into the calculation of cosine similarity. Specifically, we count the number of users who like an interest as its popularity and weight the cosine similarity according to the popularity of two users' mutual interests. The more an interest occurs, the less weight it is assigned. Thus we formulate the weighted cosine interest similarity as:

$$s_w(u,v) = \frac{\sum_{i \in I_{uv}} w(i)}{\|I_u\|_2 \cdot \|I_v\|_2} \tag{6.2}$$

in which $w(i)$ equals the inverse log N where N stands for the number of users who are interested in interest i, i.e., $w(i) = \frac{1}{\log N}$. Weighted cosine similarity is applied to compute the degree of interest similarity.

6.3.2 Collective Interest Similarity

Based on the above-introduced interest similarity metrics regarding two users, we further estimate the collective interest similarity over an aggregation of user pairs. We denote the aggregation of user pairs as C and average the interest similarities of the user pairs in C as its collective interest similarity.

In particular, we define **probability of sharing interests** (i.e., p) of user pairs in C as the mean binary similarity of the collective pairs as follows:

$$p = \frac{\sum_{(u,v)\in C} s_b(u,v)}{\|C\|} \tag{6.3}$$

In addition, we calculate the **degree of interest similarity** (i.e., s) of C as the average weighted cosine similarity of all the user pairs in C, denoted as:

$$s = \frac{\sum_{(u,v)\in C} s_w(u,v)}{\|C\|} \tag{6.4}$$

where $\|C\|$ stands for the number of pairs that are included in the pair set C. In the rest of this chapter, we use these two collective measurements to study how interest similarity varies depending on various social features.

6.4 Homophily of Interest Similarity

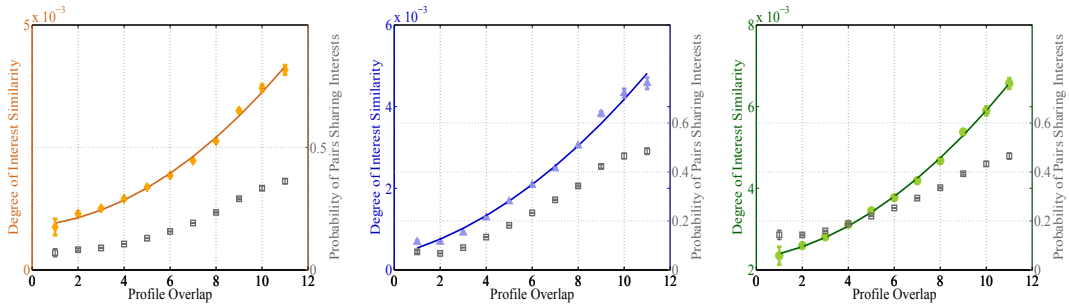
In this section, we examine the relations between interest similarity and various social features that emerge from the collective users. We investigate the changes of interest similarity with respect to demographic-related features, social relationships and interest-related feature subsequently. Note that, each empirical study is carried out on a specific social feature and a particular interest domain (i.e., movies, music and TV shows). Therefore, for each study, the pair set C is generated by considering two factors: (1) the related profile attribute and (2) the focused interest domain. For instance, to test the relation between gender and interest similarity in terms of movies, we construct a gender/movie set of pairs by coupling users who present both gender and movies. Note that we only consider the users who exhibit more than three items in the focused interest domain.

6.4.1 Interest Similarity by Demographics

We study how demographic information affects interest similarity from four perspectives, profile overlap, gender, location (geographic distance and country) and age (age distance and generation).

6.4.1.1 Interest Similarity by Profile Overlap

Profile overlap measures the number of the profile attributes where two users exhibit the same value. In particular, for each user, we generate a profile vector with 16 cells which corresponds to nine interest domains and seven demographic attributes (refer Section 3.2.1). Concerning a particular interest domain cell, if a user u presents any items in the interest domain, we say u is interested in



(a) Movies (b) Music (c) TV shows

Figure 6.2: Interest similarity with profile overlap. Standard error is estimated by bootstrap resampling throughout this chapter. The colorful left Y-axes stand for the degree of interest similarity and the right grey Y-axes indicate the probability of sharing interests.

this domain and denote the cell as 1; otherwise, it is set to 0. We directly put the users' demographic attributes into the corresponding cells.

We separately generate profile/interest sets for the three interest domains (i.e., movies, music and TV shows) with $1,000,000$ user pairs where the users present more than three interest items and at least one demographic attribute. Let $C_{d_q} = \{(q_u, q_v, p_{uv}, s_{uv}) : |q_u \bigcap q_v| = d_q\}$ denote a collection of user pairs where the profile overlap between the user pair (u and v) is d_q; q_u and q_v represent u and v' profile vectors; p_{uv} and s_{uv} are the probability of sharing interest and the degree of interest similarity between u and v respectively.

Figure 6.2 plots the interest similarity over profile overlap in movies, music, and TV shows respectively. As the number of user pairs with profile overlap beyond 11 is very small, we concentrate on the user pairs whose profile overlap falls between 1 and 11. The results reveal that both of the probability of sharing interests and the degree of interest similarity go up with the increase of profile overlap regardless of interest domains. This observation demonstrates that two users are more similar in their tastes if they share more common attributes in their profiles.

6.4.1.2 Interest Similarity by Gender

We produce gender/interest sets with $1,000,000$ randomly coupled user pairs where the users present their gender and more than three interest items (Movies, Music or TV shows). Let $C_{g_c} = \{(g_u, g_v, p_{uv}, s_{uv}) : g_u \bigcup g_v = g_c\}$ denote an aggregation of user pairs where two users are of gender combination g_c. Here, the gender combination of a user pair takes three possible values (i.e, g_c) as male-male, female-female and male-female.

Table 6.1 shows the probability of sharing interests and the degree of interest similarity according to the different gender combinations. We observe the homophily for gender that the pairs present higher interest similarities when they are in the same sex (i.e., male-male or female-female). In addition, we find that males are more similar on the interests of movies and music whereas females present higher interest similarity in TV shows.

This observation of homophily for gender here is different from the heterophily for gender in communication network reported in the previous work [92]. It demonstrates that people communicate more with the ones in the opposite gender. In other words, although people like to make connection with others of different sex, the pairs of cross-gender do not share interests highly. This suggests that we should exploit the gender property of the homophily or heterophily properly accord-

	Probability of sharing interests			Degree of interest similarity		
	Movies	**Music**	**TV shows**	**Movies**	**Music**	**TV shows**
Male & Male	0.164	0.179	0.209	0.0022	0.0019	0.0035
Female & Female	0.145	0.157	0.245	0.0020	0.0015	0.0042
Female & Male	0.118	0.151	0.176	0.0015	0.0014	0.0027

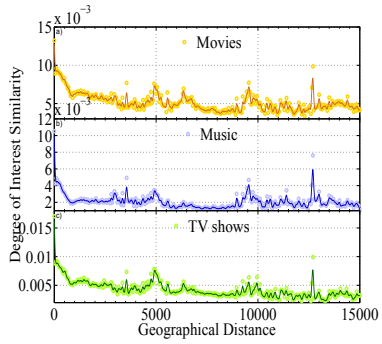
Table 6.1: Interest similarity by gender

ing to the specific applications. For instances, for some specific communication/dating applications, users in the opposite gender might take the priority to be considered; while the users of the same gender are supposed to be thought at the first place when it comes to enhancing the recommendation for interests.

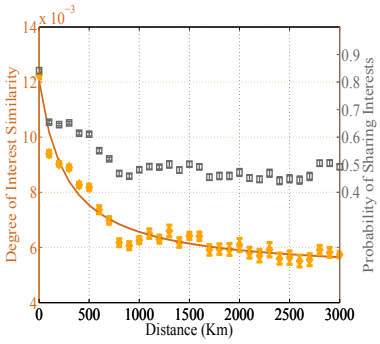
6.4.1.3 Interest Similarity by Location

We study how location affects interest similarity by geographic distance and country.

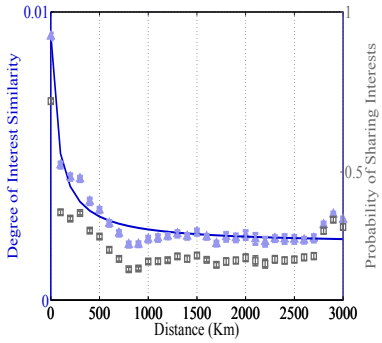
Interest similarity by geographic distance: denote a set of user pairs where the two users of a pair are apart of d_{uv} in the span of $[d_l, d_l+\nabla)$ by $C_{d_l} = \{(l_u, l_v, p_{uv}, s_{uv}) : distance(l_u, l_v) = d_{uv} \bigwedge d_{uv} \in [d_l, d_l+\nabla)\}$. l_u is the location of user u represented by its latitude and longitude and ∇ stands for an interval of distance.



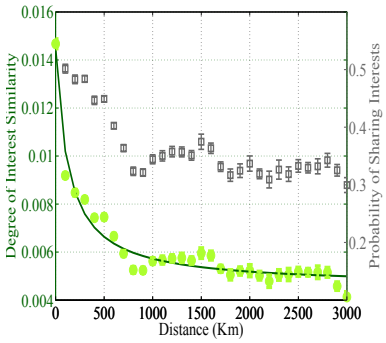
(a) Overview on degree of interest similarity



(b) Movies



(c) Music



(d) TV shows

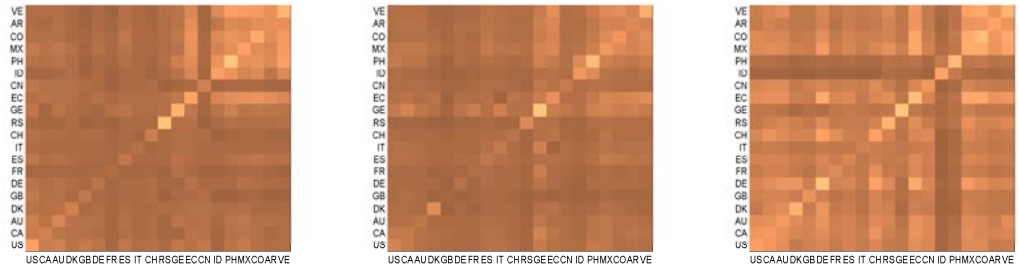
Figure 6.3: Interest similarity by geographical distance

Figure 6.3(a) reports the degree of interest similarity by a full view of distance range from 0 to 15000km with an interval of 100km. Although the results fluctuate at some points when the distances

are larger than 3000*km*, we see a decreasing trend of the degree of interest similarity by the distance. Furthermore, we zoom in the x-axes and show the interest similarity with distances in the range of 0 and 3000*km* in figure 6.3(b), 6.3(c) and 6.3(d). We observe that the interest similarity decreases quickly when the distance is small, and it gets steadily when the distance continuous increasing. This implies that the interest similarity correlates to the distance very sensitively only in a limited range of distance.

In addition, we look into a number of pair samples which might lead to the fluctuations at distances larger than 3000*km*. Taking the peak at 3500*km* as an example, we find that the two users at this distance are mostly from the east and west of the USA. Therefore, we speculate that such peaks may reveal some implicit connections (e.g., nationality, language, culture) between the specific geographic regions. Therefore, we further examine how interest similarity varies depending on the geographic region in terms of country.

Interest similarity by country: let $C_{t_{hk}} = \{(t_u, t_v, p_{uv}, s_{uv}) : t_u = h \bigwedge t_v = k\}$ denote the set of pairs in which the two users come from the countries (denoted by t_u and t_v) of h and k. We select users from 20 representative countries over six continents and randomly generate 200,000 pairs for each country combination (cross-country or same-country).



(a) Movies (b) Music (c) TV shows

Figure 6.4: Degree of interest similarity by country

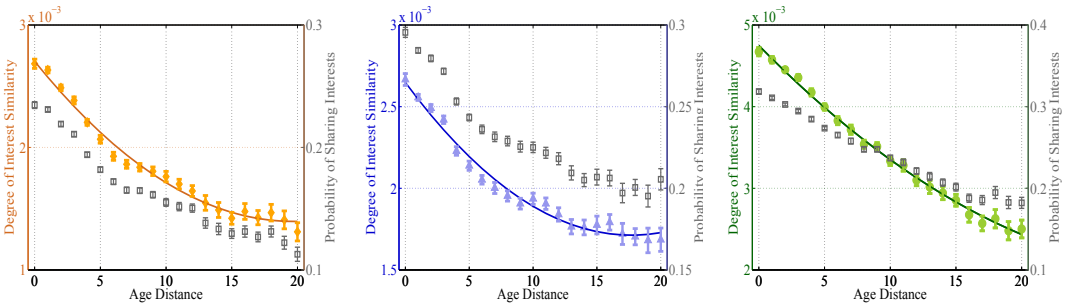
Figure 6.4 displays the heatmaps of the degree of interest similarity by country combination, where a brighter cell indicate that users from the corresponding countries (represented by the row and column) share more interests. Note that the cells on the secondary diagonal represents the interest similarity of pairs from the same country (i.e., native pairs).

We observe that the cells on the secondary diagonal is brighter than the other cells in the same row or column. This demonstrates that, compared to the pairs from two diverse nations (i.e., alien pairs), native pairs share more interests. Besides, we notice Chinese share less movies with Philippine and Indonesian, but report a high movie similarity with American. We also notice that users from South America countries share a lot of interest. This observation might imply that the different countries share interests with distinctions.

6.4.1.4 Interest Similarity by Age

How age distance and generation affect interest similarity are learned in this section.

Interest similarity by age distance: age distance measures the gap of two users in terms of age. Let $C_{d_a} = \{(a_u, a_v, p_{uv}, s_{uv}) : |a_u - a_v| = d_a\}$ denote a set of pairs whose ages differ at d_a. Note that the discussed age distance (i.e., d_a) varies from 0 to 20 years.

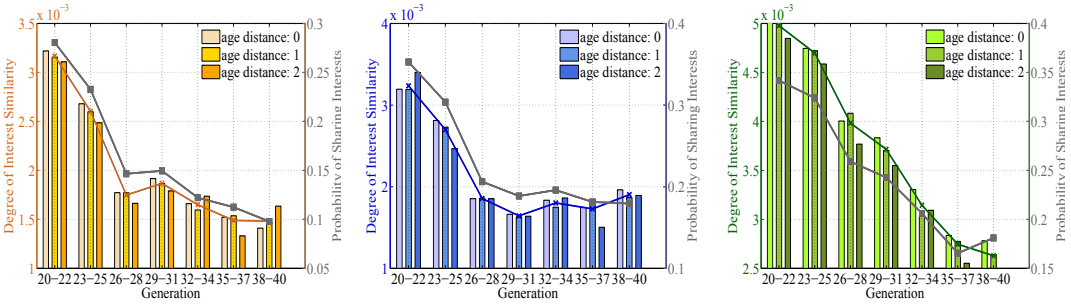


(a) Movies (b) Music (c) TV shows

Figure 6.5: Interest similarity by age distance

Figure 6.5 shows that the interest similarity decline as the age distance goes up. This observation demonstrates that users share more interests if they are closer at age. Moreover, we observe that the interest similarity drops fast when the age distance is small; and it gets to decline gradually as the age distance continues increasing.

Interest similarity by generation: Let $C_{g_a} = \{(a_u, a_v, p_{uv}, s_{uv}) : a_u \in g \bigwedge a_v \in g\}$ denote a set of user pairs where the two users are in the same generation g. Remind that we select 3 years as an age interval of one generation.



(a) Movies (b) Music (c) TV shows

Figure 6.6: Interest similarity by generation. The lines represent the interest similarity of each generation. Inside each generation, the grouped three histograms display the degree of interest similarity with age distance at 0, 1 and 2 respectively.

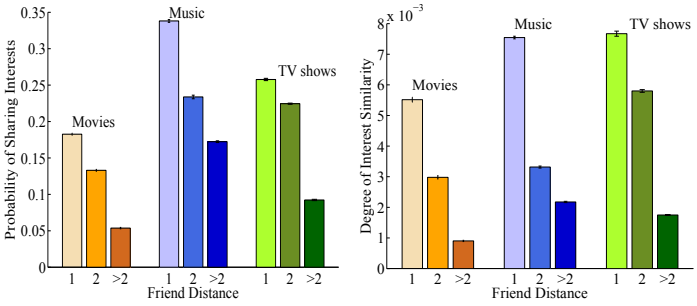
Figure 6.6 reveals that the younger generations present higher interest similarity than the middle-age generations. And comparing the interest similarity by age distance inside a generation, the results basically hold the rule that the interest similarity decrease with the increase of the age distance although several exceptions exist (e.g., 38-40 for movie).

6.4.2 Effects of Friendship

We examine interest similarity according to friendship through two perspectives: friend distance and friend similarity. Friend distance is computed by the connected hops between two users; friend similarity measures the common friends of two users.

6.4.2.1 Interest Similarity by friend distance

Let $C_{d_f} = \{(f_u, f_v, p_{uv}, s_{uv}) : D(f_u, f_v) = d_f\}$ denote a set of pairs where the friend distance of the two users u and v is d_f hops. Particularly, we take into account friendship in two-hop with three users pair groups: *direct-friend* pair — u and v connect to each other directly ($d_f = 1$); *indirect-friend* pair — u is a friend of v's friends but u and v are not direct-friend ($d_f = 2$); *stranger* pair — u and v's friend distance is larger than 2 ($d_f > 2$).



(a) Probability of sharing interests (b) Degree of interest similarity

Figure 6.7: Interest similarity by friend distance

Figure 6.7(a) and 6.7(b) report the probability of sharing interests and degree of interest similarity by friend distance respectively. These results reveal that the users with less friend distance share more interests: *direct-friend* pairs exhibit the highest interest similarity; and the *indirect-friend* pairs share more interests than the *stranger* pairs do.

6.4.2.2 Interest similarity by friend similarity

Friend similarity measures two users' common friends by cosine similarity, i.e., $f_{uv} = \frac{\|f_u f_v\|}{\|f_u\|\|f_v\|}$. Note that we only consider the user pairs who present at least one mutual friend where 95% of them show a friend similarity less than 0.02. So the studied friend similarity is in the range of $(0, 0.02]$. Let $C_{s_f} = \{(f_u, f_v, p_{uv}, s_{uv}) : \frac{\|f_u f_v\|}{\|f_u\|\|f_v\|} = f_{uv} \bigwedge f_{uv} \in [f_s, f_s + \nabla)\}$ denote a set of user pairs in which the two users exhibit a friend similarity in the range of $[f_s, f_s + \nabla)$. ∇ represents an interval of friend similarity.

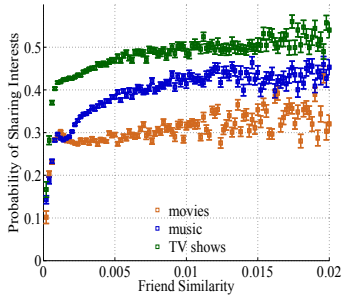
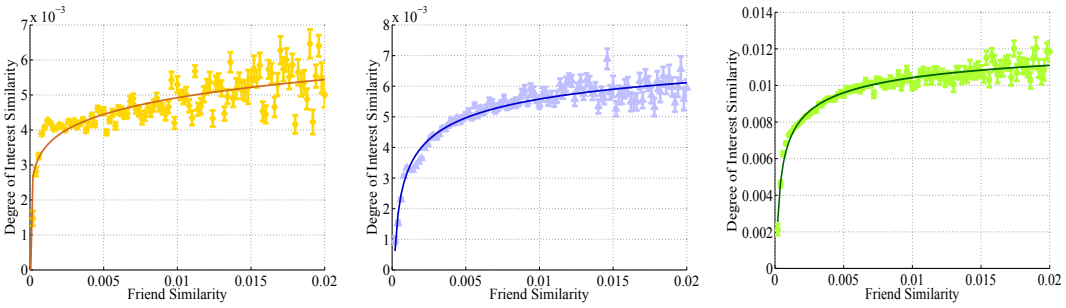


Figure 6.8: Probability of sharing interests by friend similarity

Figure 6.8 shows the change of the probability of sharing interests with friend similarity; figure 6.9(a), 6.9(b) and 6.9(c) display the relation between the degree of interest similarity and friend



(a) Movies (b) Music (c) TV shows

Figure 6.9: Degree of interest similarity by friend similarity

similarity with respect to movies, music and TV shows respectively. All these figures reveal that the user pairs generally share more interests if they obtain a higher friend similarity. In particular, we observe that the interest similarity goes up steeply when the friend similarity is less than 0.001, and hereafter it becomes steady with rise of friend similarity.

6.4.3 Effects of Interest Entropy

In this section, we are interested in looking at interest related feature. We employ entropy to capture a user's interest feature. Entropy quantifies the information amount of the user's interests by two elements of the interests: the number of interests and the weight of interests. Generally speaking, a user with many high weighted interests should be assigned with a large entropy. Using the natural log, we define interest entropy $H(I_u)$ as:

$$H(I_u) = -\sum_{x_i \in I_u} w(x_i) \log w(x_i)$$

Where $w(x_i)$ represents the weight of interests x_i (defined in Section 6.3). As 95% of users' interest entropy is less than 8, we discuss the interest similarity by entropy in $[0,8]$.

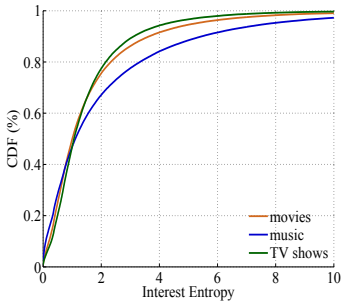
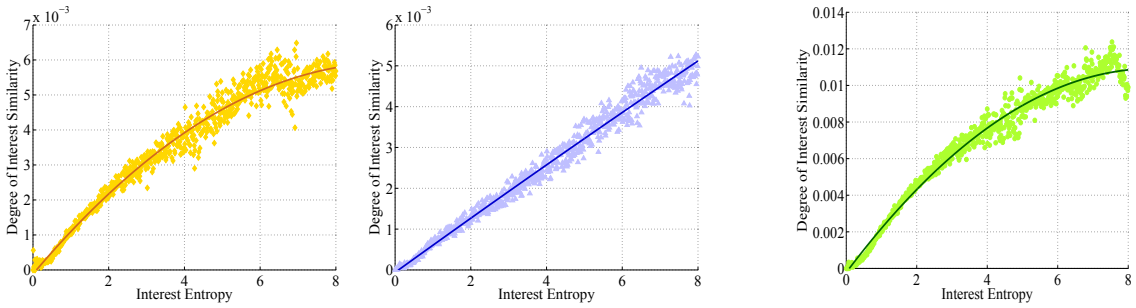


Figure 6.10: CDF of interest entropy

Let $C_{e_i} = \{(I_u, I_v, p_{uv}, s_{uv}) : H(I_u) = e_i \bigvee H(I_v) = e_i\}$ denote a set of pairs by users' interest entropy of e_i. Note that, in this set, only one user in a user pair is required to have an interest entropy of e_i. Because we tend to study whether the interest similarity would be influenced by one user' interest entropy in a pair.



(a) Movies (b) Music (c) TV shows

Figure 6.11: Interest similarity by interest entropy

Figure 6.10 displays the probability of sharing interest; Figure 6.11(a), 6.11(b) and 6.11(c) show degree of interest similarity. We observe that the interest similarity grows as the increase of interest entropy. And it particularly rises very quickly as the interest entropy is small.

6.5 Inferring Interest Similarity

In the previous section, we conducted extensive analysis of how various social features correlate to interest similarity of two users. The goal of this section is to design a prediction model for inferring whether two users are similar in interest (namely interest similarity between users) relying on these new learned correlations.

Let us consider many applications which directly exploit interest similarity between users to improve the performance [155][158][104]. Obviously, the interest similarity can be easily computed if both of two users' interests are known. However, as there are always some users not revealing their interests, for such applications, missing users' interests is indeed a practical obstacle to computing interest similarity directly (e.g., new user problem in recommendation system [119][5][161][160][159]). Therefore, it is appealing to infer two users' interest similarity for this case.

Besides, users' interests are normally desirable for personalized recommending or advertising [156][157]. For a number of passive users who do not explicitly reveal their interests (51% of users in our Facebook data set), if it is possible to capture some active users who not only expose their own interests but also are predicted to have similar interests as a given passive user, then we can infer the passive user's interests according to the similar active users' interests. In this case, how to predict users' interest similarity (i.e., to determine whether two users are similar or not in their interests) without knowing interests from one of the users becomes a meaningful problem.

Specifically, in this prediction, we consider two users: a passive user u who only presents some demographic information and social relationships with limited friends but does not reveal his interests (i.e., $u_p : \langle D_p, F_p \rangle$); and an active user v who has complete information including demographic attributes, friends as well as interests (i.e., $u_a : \langle D_a, F_a, I_a \rangle$). Then, ***the prediction task is to determine whether the passive user u and the active user v are similar or dissimilar regarding their interests***.

6.5.1 Interest Similarity Prediction Model

According to the prediction task itself, two possible results are expected: i) the given passive user u and active user v are similar regarding their interests (i.e., labeled as *interest-similar*); ii) u and

v are not similar (i.e., labeled as *interest-dissimilar*). To achieve the task, the basic idea is to train a prediction model to label u and v as either *interest-similar* or *interest-dissimilar* by learning their *social features*. Therefore, in this section, we introduce our prediction model in details from three aspects: (1) we clarify the criterion to determine whether two users are *interest-similar* or *interest-dissimilar*; (2) we illustrate the social features that are leveraged to train the prediction model; (3) by exploiting Support Vector Machines (SVM) method [163], we establish our interest similarity prediction model, namely *InterestSim* model.

Criterion: Given a pair of users u and v, whether they are similar or dissimilar is determined by their interest similarity and an established threshold. We compute u and v's interest similarity by the degree of interest similarity (i.e., $s_w(u,v)$) and compare the value to the established threshold (i.e., ε). We use z_{uv} to label the interest similarity between u and v. If the interest similarity is larger than ε, z_{uv} is labeled to 1, representing u and v are *interest-similar*; otherwise, z_{uv} is labeled to -1, indicating u and v are *interest-dissimilar*:

$$z_{uv} = \begin{cases} 1 & s_w(u,v) \geqslant \varepsilon \\ -1 & s_w(u,v) < \varepsilon \end{cases} \tag{6.5}$$

Social Features: Moreover, given a passive user u, an active user v and all of their obtainable social information (i.e., demographic information, friends and v's interests), we extract the following social features drawing on the studies in the previous section:

- *Profile Overlap* (PO_{uv}) computes the percentage of the same attributes that u and v share among the seven demographic attributes: age, gender, current city, hometown, high school, employer, and college.
- *Gender Combination* (GC_{uv}) takes three possibilities: 1 (male-male), -1 (female-female), and 0 (male-female).
- *Geographic Distance* (GD_{uv}) measures the distance between u and v's current city (refer to Section 6.4.1.3).
- *Binary Country* (BC_{uv}) is set to 1 if u and v come from the same country; otherwise it equals 0.
- *Age Distance* (AD_{uv}) calculates the absolute difference of u and v's ages.
- *Friendship Distance* (FD_{uv}) is set to 1 if two users are friends; otherwise, it equals 0.
- *Friend Similarity* (FS_{uv}) is calculated by cosine similarity (refer to Section 6.4.2.2).
- *Interest Entropy* (IE_v) is computed by the active user v's interests (refer to Section 6.2 and 6.4.3).

Note that we normalize *Geographic Distance*, *Age Distance*, *Friendship Distance* and *Interest Entropy* to ensure all the features belonging to $[-1,1]$. Thus, for the user pair u and v, we obtain a social feature vector: $\boldsymbol{x}_{uv} = \langle PO_{uv}, GC_{uv}, GD_{uv}, BC_{uv}, AD_{uv}, FD_{uv}, FS_{uv}, IE_v \rangle$.

SVM-based *InterestSim* model: So far, from each user pair (u,v) where u is a passive user and v is an active user, we can generate a tuple $\langle \boldsymbol{x}_{uv}, z_{uv} \rangle$. $\boldsymbol{x}_{uv}$ is the social features extracted from u and v's social information; z_{uv} is the label which stands for whether u and v are *interest-similar* or *interest-dissimilar*. To train the *InterestSim* model, we aggregate a number of user pairs where all the pairs are made of a passive user and an active user. Similarly, from all these user pairs, we can generate

a tuple collection where each tuple corresponds to a pair of users, denoted as $\mathscr{C}\{pair_i : (\boldsymbol{x}_i, z_i)\}$. Assume q stands for the total number of the user pairs and i denote the ith pair. Then constructing the SVM-based prediction model is solving the following optimization problem:

$$\begin{aligned} &\min L(w) = \tfrac{1}{2}\|w\|^2 + \delta \textstyle\sum_{i=1}^{q} \xi_i \\ &\text{subject to: } \begin{cases} \xi_i \geqslant 0 \\ z_i \langle w, \boldsymbol{x}_i \rangle \geqslant 1 - \xi_i \end{cases} \end{aligned} \tag{6.6}$$

where δ is a constant and $\xi_i, (i = 1, ..., q)$ are slack variables for optimization. Note that, for training the prediction model, we assume that u's interests are known to calculate u and v's interest similarity so as to determine the label (*interest-similar* or *interest-dissimilar*). However, when computing the social features, we think of u's interests as unavailable information in keeping with the prediction problem's pre-condition that u is a passive user.

Specifically, to train the proposed *InterestSim* model, we generate 150,000 user pairs by randomly coupling two users (u and v) where both u and v exhibit all the demographic information, friend lists as well as more than three interests in movies, music, or TV shows. Afterward, we split the whole 150,000 user pairs into ten subsets (i.e., 15,000 user pairs per subset) and do a ten-fold cross validation.

6.5.2 Evaluation of Prediction

In this section, we are going to evaluate the *InterestSim* model through two ways: (1) we leverage the 'leave-one-feature-out' approach to investigate the effects of various social features on the interest similarity predictions; (2) we evaluate the performance of *InterestSim* model and compare it with other two baseline approaches.

6.5.2.1 Leave-One-Feature-Out Evaluation

We carry out 'leave-one-feature-out' comparisons and train prediction models by excluding one of overall features. For instance, we train a *No Profile Overlap* model by taking out *Profile Overlap* from the social feature vector $\boldsymbol{x}_{uv}$. In addition, for some features originated from one attribute, we remove them as one integrated feature to train the 'leave-one feature-out' model. For example, we view Friendship Distance and Friend Similarity (both originated from friend lists) as an integrated feature, namely *Social Relation*; and also regard *Geographic Distance* and *Binary Country* as *Location*. In particular, we generated models without any one out of the six features of *Profile Overlap*, *Gender Combination*, *Age Distance*, *Location*, *Social Relation*, and *Interest Entropy*. In total, we obtain 18 'leave-one-feature-out' models with respect to the three interest domains of movies, music and TV shows (6×3).

Table 6.2 compares the 'leave-one-feature-out' models with the *InterestSim* model in terms of the areas under ROC curves (AUCs). From the table, we can see that our proposed *InterestSim* model, which infers interest similarity according to all the learned social features, outperforms the other models which miss one type of social features. It demonstrates that all the used social features are beneficial for the prediction. Note that a social feature (e.g. *Gender Combination*) would be more important if the AUC of a model trained without the feature (e.g., *No Gender Combination* model) is smaller. Therefore, from the results, we can say that *Profile Overlap*, *Gender Combination* and *Social Relation* are less sensitive in the predictions of interest similarity compared to the other attributes, such as *Interest Entropy*, *Age Distance*, and *Location*. In addition, we observe that the

Type of Model	AUC		
	Music	**Movies**	**TV shows**
No Profile Overlap	0.6201	0.6388	0.6825
No Gender Combination	0.6521	0.6410	0.6889
No Age Distance	0.5831	0.5943	0.6061
No Location	0.5490	0.5880	0.6550
No Social Relation	0.6491	0.6206	0.6727
No Interest Entropy	0.5171	0.5236	0.6047
***InterestSim* Model**	**0.6720**	**0.6644**	**0.7027**

Table 6.2: Comparison of effects on interest similarity prediction by different social features

impacts of the social features on the predictions in different interest domains exhibit their own properties. For instance, *Location* is more sensitive to music similarity prediction than movie similarity prediction, while *Social Relation* plays a more important role in movie similarity prediction than music similarity prediction.

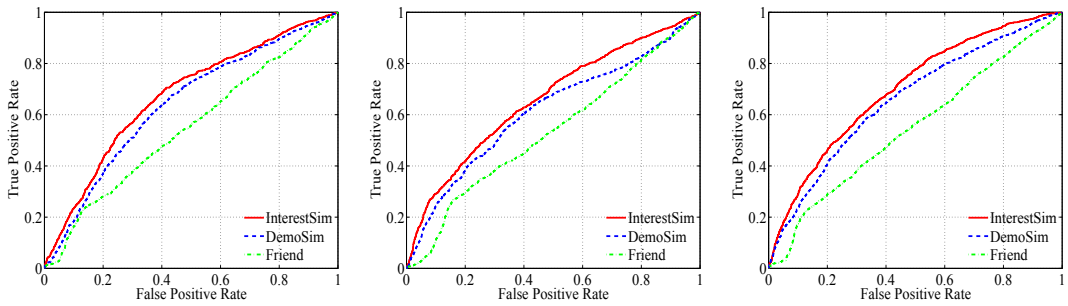
6.5.2.2 Prediction Performance Comparison

To the best of our knowledge, this is the first work aiming at inferring whether two users are similar or not in terms of their interests, without knowing one user's interests. Some existing work has pointed out several good features that can indicate similar interests between users. The friendship between two users is one of the most acknowledged feature that are used to infer a user's interests from the other's [5][164][159]. Additionally, in order to make accurate recommendations for new users without rating any items, demographic information is also explored to indicate that users with more common demographic information might share more interests [119][161][160]. Therefore, we draw on their main ideas on interest similarity indications and train two baseline prediction models respectively exploiting users' friendships and demographic information, namely *Friend* model and *DemoSim* model. In particular, we train *Friend* model by using two features: *Friend Distance* and *Friend Similarity*; and we construct the *DemoSim* model by applying *Profile Overlap*, *Age Distance*, *Gender Combination* and *Geographic Distance*.

Figure 6.12 plots the ROC curves for the three interest domains of movies, music, and TV shows, comparing the proposed *InterestSim* model to the *Friend* model and *DemoSim* model in the aspect of prediction capacity. Table 6.3 compares AUCs between the three sets of models. The ROC curves of *Friend* model almost approach to the secondary diagonal which represents the capability of random prediction. It indicates that we can hardly infer users' interest similarity merely with respect to their friendships. By considering four demographic features which involves in seven profile attributes, *DemoSim* model generates larger AUCs and performs better than *Friend* model. Even though, much of the area improvement under the ROC curves of *InterestSim* model has been shown in figure 6.12. From table 6.3, for movies, music and TV shows, we gain more than 3%-4% of improvement compared with *DemoSim* in terms of AUC.

	Friend	***Demo***	***InterestSim***
Music	0.5487	0.6411	**0.6720**
Movies	0.5335	0.6142	**0.6644**
TV shows	0.5478	0.6593	**0.7027**

Table 6.3: AUC comparisons among *Friend* model, *Demo* model and *InterestSim* model



(a) Movies (b) Music (c) TV shows

Figure 6.12: ROC curves of prediction

6.6 Case Study: Recommendation for New Users

Recommendation system recommends items to a user if these items are presumably preferred by the user. In order to make efficient recommendations, many existing approaches, which are categorized as *content-based recommendations*, *collaborative recommendations* and *hybrid recommendations*, need to acquire the users' interests. These approaches encounter a common and difficult problem — *new user problem* — when the recommendations are required for the new users who have no or very little information about their interests [156][165]. Fortunately, our proposed *InterestSim* model just can make a bridge between the new users and their interests via some existing active users who present interests in the recommendation system: we can recommend the interests of the existing active users who are predicted being similar in interest with the new users. For this reason, we leverage our proposed *InterestSim* model to address the *new user problem*. With this case study, we aim at demonstrating the practical use of our proposed prediction model.

6.6.1 Approaches

In this subsection, we briefly describe how to recommend items to a new user based on our proposed *InterestSim* model — namely *InterestSimPop* recommendation; we also introduce several state-of-the-art new user recommendation approaches to compare with:

- *InterestSimPop:* exploits *InterestSim* model to infer a number of users who are similar with the new user in interest; and then it recommends the new user the most popular items that liked by those similar users;
- *OverallPop:* For a new user without claiming his interests, a straightforward way is recommending the overall most popular items among all the existing users. Such a method, called *OverallPop* here, is often used as an intuitive baseline in the existing research about the new user problem [166];
- *FriendPop:* In [5][159], the authors indicate that using the friends' interests may facilitate the recommendation performance for a new user. We thus borrow the basic idea from these works to implement the *FriendPop* baseline method, which selects the most popular items among a new user's friends;
- *DemoSimPop:* Demographic information, such as age, location, gender, is another useful source to tackle the new user problem [119][160][161]. Following the idea in [119], *De-*

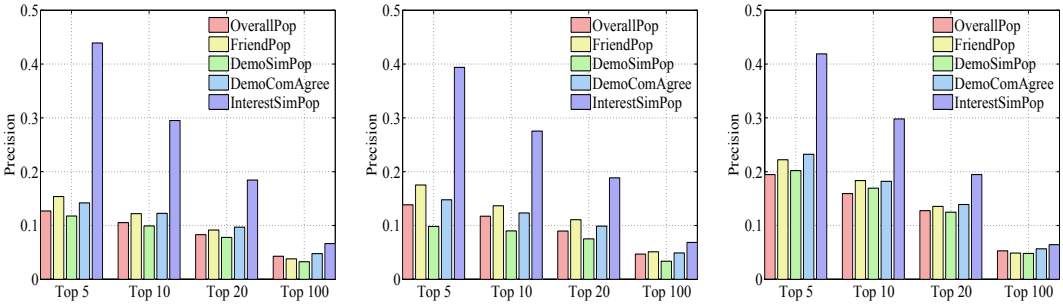
moSimPop first finds the users whose demographic attributes (e.g., gender, location, and age) are similar to the new user, and then selects the most popular items from those demographic-similar users;

- *DemoComAgree:* Based on α-community spaces model and 'level of agreement' of the community, the authors propose another way to use demographic information to improve the item recommendation for a new user [120]. Here, we also implement this method and call it as *DemoComAgree*.

6.6.2 Experiment Setup and Results

According to our data set, we randomly select 200 users who present demographic information (including age, gender, current city, hometown, high school, college and employer), friends and interests respectively in terms of movies, music and TV. We hide these users' interests and collect them into a new users set (i.e., U_{new}) to recommend items. In addition, we use the rest of users who present more than 3 movies, music or TV shows as the *existing active users*. By using the above-mentioned recommendation approaches, we generate recommendation item lists for the new users from the preferences of the *existing active users*, and eventually we compare the recommended items with the new users' real preferences.

To evaluate and compare the performance of the above-mentioned approaches, we respectively select the top 5, top 10, top 20 and top 100 items to generate the recommendation lists. We estimate the effectiveness of the recommendations by a quite commonly used metric — precision [156][157][120][166]. In fact, *precision* estimates how many percentage of recommendations are the users' real interests. Assume that a new user $u \in U_{new}$ has p_u specific preferences; we recommend q_u items to the u where r_u among these q_u items are u's real interests. Then, we have $precision = \frac{1}{N}\sum_{u \in U_{new}} r_u/q_u$, where N is the number of new users in U_{new}. By the definition of *precision*, a good recommendation approach should exhibit a large *precision*.



(a) Movies (b) Music (c) TV shows

Figure 6.13: Evaluation on recommendation precision

Figure 6.13 compares the *precision* of our proposed *InterestSimPop* recommendation to the other four baselines. We observe that our proposed *InterestSimPop* approach achieves the largest *precision* no matter what the interest domain refers to. This indicates that our proposed approach can improve effectiveness of recommendations for a new user. For instance, in Figure 6.13(a), the *precision* of *InterestSimPop* is around 0.45 for the top 5 recommendations, which means we can correctly recommend $2-3$ movies out of the top 5 recommendations to the new users on average; however the other approaches cannot ensure 1 correct movie recommendation.

6.7 Discussion

In this section, we further discuss two concerns: 1) social feature selection; 2) the practical use of the proposed interest similarity prediction model.

Social feature selection: To fully exploit the obtainable information in the prediction, besides demographic information and friendships, we handily use interest entropy to characterize the active user's interests and luckily find that two users' interest similarity correlates to interest entropy. Thus, we leverage the active user's interest entropy with other demographic and friendship features into the prediction model. The 'leave-one-feature-out' evaluation reveals the positive effects of interest entropy and all other social features. This just indicates all the studied social features can improve the prediction. For the future work, we may improve the prediction model if more social features could be obtained.

Use of the proposed prediction model: We have illustrated how to use our prediction model to enhance the recommendation for new users. We also believe that our proposed model can be easily used to other applications, like friend recommendation. Although several existing approaches may rely on mutual friends, colleague or classmate, we propose to recommend friends according to interest similarity for the following reasons: 1) as our proposed interest similarity prediction model exhaustively exploits the users' obtainable information, the interest similarity based friend recommendation may substitute for the existing approaches once their requisite information (e.g., friend, job or school) is missing; 2) The promising of the interest-based OSNs like Pinterest, CircleMe and Yaamo reveals that people like to connect other people with similar interests. It has also been proved that users who share certain interests are more likely to be friends [155][104][103]. Thus, a mixed solution, which includes all the approaches based on mutual friend, colleague, classmate and interest similarity, may be an alternative.

6.8 Summary

As users do not always explicitly elaborate their interests in OSNs, in this chapter, we address a practical problem for OSNs: How to infer two users' interest similarity when we cannot fully know their interests?

To solve this problem, from users' demographic information, friendships and their interests, we first attempt to identify some users' social features (e.g. geographic distance, friend similarity) that are strongly correlated to their interest similarity. In particular, we conduct a comprehensive empirical study on how users' interest similarity relates to various social features with a large Facebook data set in three interest domains (i.e. movies, music, and TV shows). The result reveals that people tend to exhibit more similar tastes if they have similar demographic information (e.g., age, location) or share more common friends. In addition, we also find that the individuals with a higher interest entropy would generally share more interests with the others. Finally, we identify several effective social features that are strongly correlated to users' interest similarity, including geographic distance, gender combination, age distance, friend similarity, interest entropy, etc.

Based on the above identified social features, we propose a user interest similarity prediction model that can determine whether two users are similar or not in an interest domain while interests cannot be obtained from one of them. The evaluation demonstrates that the prediction model integrating all the learned social features outperforms other models that lack some of those features.

Chapter 7

Information Relevance and Leakage: Predicting Location and Preserving Privacy

Contents

7.1 Introduction

Owing to the increase of potential violations such as advertising spam, online stalking and identity theft [167], in recent years, more and more users in OSNs start to concern their *privacy* and become reluctant to expose all their personal information to public [168]. Consequently, the users may not fill the privacy-sensitive attributes (e.g., location, age, or phone number) or hide them from strangers and merely allow such information visible to their friends.

While hiding the privacy-sensitive attributes, users usually expose some other information that seems less privacy-sensitive to them. It has been reported that Facebook users publicly reveal four attributes on average and 63% of them uncover their friends list [169]. Due to the correlations among various attributes, some of the self-exposed information may indicate the invisible privacy-sensitive attributes to some extent [170]. In such a case, whether the privacy-sensitive attribute that a user intends to hide is really undercover is in doubt.

In this work, exploiting users' location information as a representative, we attempt to understand what is the risk that a user's invisible information would be disclosed. There exist several reasons that lead us to conduct this study based on location information. First, among various kinds of information, location is usually one of the privacy-sensitive attributes for a user. In real-life OSNs, we notice that users are quite careful to reveal their location information: 16% of users in Twitter reveal home city [89] and 0.6% of Facebook users publish home address [171]. Moreover, for third-parties, location information is a valuable attribute that can be utilized for commercial purposes; this may tempt the third parties to infer users' hidden location information. Even worse, the location information might be misused by unscrupulous businesses to bombard a user with unsolicited marketing, or even lead to more severe harms such as stalking and physical attacks [172]. Therefore, protecting the hidden location information for a user becomes rather critical. In particular, as Facebook is the most popular OSN [173], we concentrate on the attribute of *current city* in Facebook and investigate the following issues:

1) *Is the private current city that a user expects to hide really undercover? In other words, if a user hides his current city but exposes some other information, can we predict a user's current city by using his self-exposed information?*

2) *For an individual user, can we help him to understand the actual risk (probability) that his private current city could be correctly predicted based on his self-exposed information? Furthermore, can we provide some countermeasures to increase the security of his hidden current city?*

To address the aforementioned issues, we first propose a current city prediction approach to predict users' hidden current city. Although many location prediction approaches have been developed for Twitter [121][122][123][174] and Foursquare [170][127], they cannot be appropriately leveraged to Facebook because of the different properties (e.g., obtainable information) in these OSNs. For Facebook, Backstrom et al. predict users' locations based on their friends' locations [171]. In

order to achieve high prediction accuracy in Facebook, we first conduct empirical studies to demonstrate that friends' locations, users' profile attributes, such as hometown, school and workplace, may indicate their current city to some extent. Furthermore, we devise a novel current city prediction approach by extracting location indications from integrated self-exposed information including profile attributes and friends list.

Second, based on the proposed prediction approach, we construct a current city exposure estimator to estimate the exposure probability that a user's invisible current city may be correctly inferred via his self-exposed information. The exposure estimator can also provide a user with some countermeasures to keep his hidden current city undercover. To the best of our knowledge, this is the first work that attempts to estimate a user's exposure probability of an invisible attribute by his self-exposed information.

It is a non-trivial task to construct either the current city prediction approach or the exposure estimator. We encounter the following challenges:

1) ***How to extract and integrate different location indications from a user's multiple self-exposed information?*** Since the proposed prediction approach tends to explore location indications from both profile attributes and friends list, two subproblems are considered. (i) A user probably reveals multiple attributes (e.g., hometown, workplace) which may indicate different locations; besides, a certain attribute might indicate several locations. For example, a user working in GOOGLE suggests that the user could probably live in any city where GOOGLE sets up an office e.g., CALIFORNIA, BEIJING or PARIS. (ii) The friends of a user, probably residing in different cities, may be close to or far away from the user. These strong or weak geographic relations may influence the significance of the friends' location indications. Thus, it is challenging to appropriately combine these various location indications into an integrated model, so as to determine probabilities to locations where the user may live.

2) ***How to predict a user's current city when we obtain the probabilities of the user being at various locations?*** By overcoming *challenge 1*, we can obtain a probability vector which indicates the probabilities that a user resides at certain locations. At first glance, with this probability vector, we might easily predict the location with the highest probability as the user's current city. However, this might not be the best option when concerning the locations' geographic relations. Assume the probability vector suggests that a user u has 40%, 35% and 25% probability of residing in BEIJING, PARIS and EVRY respectively. Then, u is more likely to live in the area around PARIS and EVRY than BEIJING, because PAIRS and EVRY are only 30km apart but they are thousands of kilometers away from BEIJING. Hence, a location selection method should be carefully designed for a current city prediction approach.

3) ***How to estimate the exposure risk of a user' hidden current city?*** To help a user understand the exposure risk of his hidden current city, a straight-forward method is providing the user with a predicted location; thus the user can decide whether his current city can be predicted correctly (risky) or incorrectly (secure). However, this method may not meet users' expectations. For a user whose location is correctly predicted, he may expect to know which of his self-exposed information primarily leads to the leakage of his private current city and how to increase its security. For another user whose hidden location is not predicted correctly, still some leakage of location leakage may exist. For example, a prediction approach may incorrectly infer a parisian living in LYON according to probabilistic results: 55% in LYON and 45% in PARIS; Even though the prediction result is incorrect, the user still leaks some location information. Therefore, how to estimate the current city exposure risk and help a user achieve his privacy intention is a challenging objective.

This work makes the following contributions:

1) **Profile and friend location indication model:** To properly reveal location indications from

users' self-exposed information, we construct an integrated probability model. We capture location indications from two types of information: *location sensitive attributes* and *friends list*. Location sensitive attributes are the profile attributes that can indicate one or multiple locations (e.g., hometown). For each location sensitive attribute, we set up a *location attribute indication matrix* from which we can index the locations and the corresponding probabilities that a certain attribute value indicates. Besides, considering a user and one of his friends who publish current city, we estimate their location similarity according to their attribute correlations, and assign a large weight to the friend if he has a high location similarity to the user. For a friend not revealing current city, we predict the friend's current city using his visible location sensitive attributes, and assign a very small weight to him. Eventually, based on our Facebook data set, we train an integrated model that can tell the probability for each potential city where a user may reside.

2) **Current city prediction approach:** To address *Challenge 2*, we aggregate locations into clusters by considering the locations' geographic relations. Then, based on the proposed *profile and friend location indication model*, we predict a user's invisible current city by two steps: (i) cluster-selection: for each cluster, we sum up the probabilities of locations inside the cluster; then we select the cluster with the highest probability; (ii) location-selection: we determine a best location within the selected cluster as the user's current city. The evaluation results demonstrate that our proposed prediction approach achieves less error distance and higher accuracy than the state-of-the-art approaches. Furthermore, for the users who reveal their 'Hometown' and 'Work and Education', our proposed approach can predict current city with an accuracy of 90%.

3) **Current city exposure estimator:** We define some measurements to describe the characteristics of users' self-exposed information. Based on these measurements, we analyze how the users' self-exposed information affects the probability that users' current city may be correctly inferred (i.e., current city exposure probability). Furthermore, we use a regression method to model the current city exposure probability and construct a current city exposure estimator. Given a user's self-exposed information, the proposed exposure estimator provides two estimators — *Exposure Probability* and *Risk Level* — to quantify the current city exposure risk. The exposure estimator can also estimate the exposure risk assuming that the user hides some of his self-exposed information. Consequently, the user can easily decide which information he should hide to satisfy his privacy intention.

7.2 Empirical Studies on Location Correlation

In this section, concentrating on current city, we tend to explore and display how location sensitive attributes can indicate a user's current city from two perspectives. First, we examine how a user's current city (i.e. CC) correlates to the location of her hometown (i.e. HT), high school (i.e. HS) and employer (i.e. EM). Then, we investigate how the location information of a user's friends correlates to the user's location information. We conduct the studies on both city-level and country-level.

7.2.1 Location Correlation between User's Attributes

Concerning a certain location sensitive attribute ($i.e., a_i(u)$), we compare a user's current city (i.e., $c(u)$) and the corresponding city of $a_i(u)$: if $c(u)$ and $a_i(u)$ are equal in city, then we denote $F(c(u), a_i(u)) = 1$; otherwise $F(c(u), a_i(u)) = 0$. We calculate the percentage of users who have identical location on $c(u)$ and $a_i(u)$ as the Average Correlation between $c(u)$ and $a_i(u)$, denoted as:

$$\overline{Cor_{(c,a_i)}} = \frac{\sum_{i=1}^{M} F(c(u), a_i(u))}{M} \tag{7.1}$$

where M represents the number of users who indicate both $c(u)$ and $a_i(u)$.

Figure 7.1 plots the Average Correlation between current city and the location sensitive attributes in terms of HT, HS and EM. We can see that around 60% of people live in the same city as their hometown. On the contrary, employer location does not match current city with a high probability as our expectation. One possible reason could be many large companies have branches all over the world but only indicate the address of the headquarters on their web sites. However, we still find 56% of users have the same employer city as their current city. While 42.8% of users present their current city as same as the city of their high school. At country level, we note that more than 80% of users stay in the country of their hometown, employer and high school.

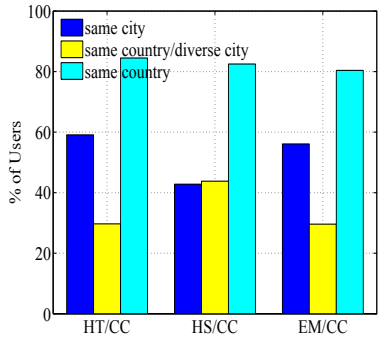


Figure 7.1: User-centric correlation

We further investigate the location change from hometown to current city by continents. The results are shown in figure 7.2 where we use AF, AS, EU, NA, AU and SA represent Africa, Asia, Europe, North America and South America. According to the results, almost 70% people in Australia leave their hometown and move to a new place, therefore it is much harder to infer Australians current location from their hometown. However, more than 60% of people in Asia and South America stay in their hometown. The correlations of current city and hometown are much higher in these continents.

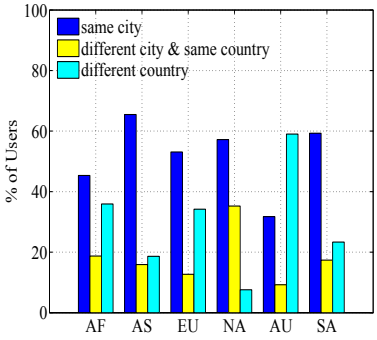


Figure 7.2: Correlation of current location and hometown by continent

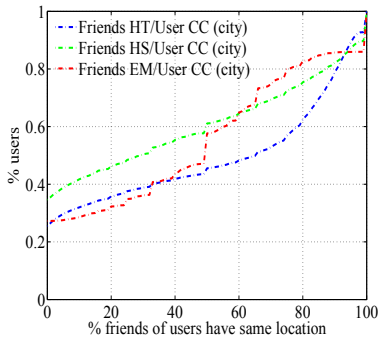
7.2.2 Location Correlation between Friends

People usually make friends with their classmates who go to the same school, colleagues who work in the same company, and others who participate in a same off-line activity. Therefore, we expect that the location information from a user's friends may give indications to where the user is.

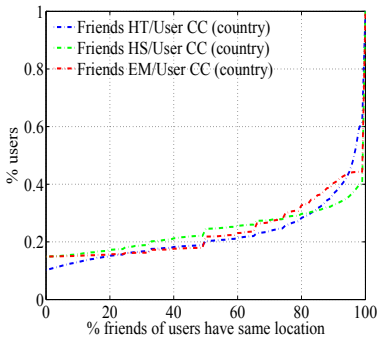
Given a location sensitive attribute i of user u and a location sensitive attribute j of her friend f_k^u, denoted respectively as $a_i(u)$ and $a_j(f_k^u)$, we compare their location: if $a_i(u)$ and $a_j(f_k^u)$ are identical in location, then $F(a_i(u), a_j(f_k^u))) = 1$; otherwise $F(a_i(u), a_j(f_k^u)) = 0$. On this basis, we define the correlation between users on attribute i to their friends on attribute j as the percentage of u's friends whose have $a_j(f_k^u)$ be identical to $a_i(u)$ in location, denoted as:

$$Cor_{(a_i(u),a_j(f^u))} = \frac{\sum_{k=1}^{N} F(a_i(u), a_j(f_k^u))}{N} \tag{7.2}$$

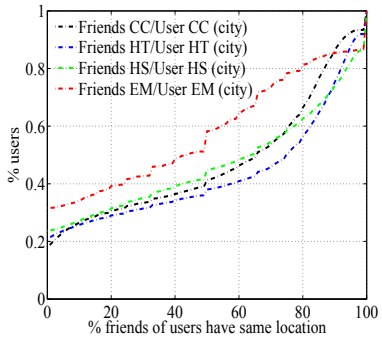
where N represents the number of u's friends who publish attribute j.



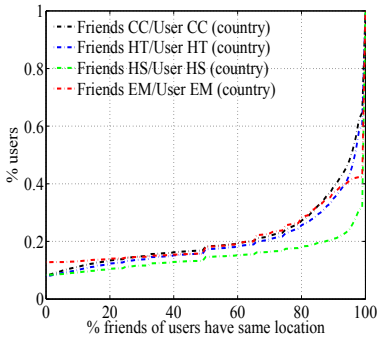
(a) Correlation between users' current city and friends' location (city)



(b) Correlation between users' current country and friends' location (country)



(c) Correlation between corresponding location (city)



(d) Correlation between corresponding location (country)

Figure 7.3: Location correlation between users and their friends

First we compute the interrelation of users' current location and their friends' hometown, employer, and high school respectively, shown in figure 7.3(a) and figure 7.3(b). In city level, more than 50% of users are in the city that 60% of their friends come from; and around 40% of users live in the same city that 50% of their friends work in or where they went to high school. While at country level, about 70% of users live in the same country as 80% of their friends.

We also calculate what percentage of a users' friends have the same location (current location,

hometown, employer, high school) as the users' corresponding locations. At city level(figure 7.3(c)), we have around 50% of users who work in the same city as more than 50% of their friends; around 60% of users went to the same school as more than 50% of their friends; and 60% of users come from the same city as more than 60% of their friends. At country level, the number of location correlations goes even higher – more than 70% of users are in the same country as 80% their friends (figure 7.3(d)).

Based on the above preliminary analysis and observations of location correlation, we conclude that: (1) users' current city correlate to their hometown, high school and employer locations to a certain degree; and (2) friends' explicit and implicit location attributes reflect users' current location to some extent. Moreover, these observations suggest that we may predict a user's current city from her self-exposed information including profile attributes and friends list.

7.3 Predicting Current City: Problem Statement

In this section, we formulate the current city prediction problem. Facebook, as a social network containing location information, can be viewed as an undirected graph $\mathscr{G} = (\mathscr{U}, \mathscr{E}, \mathscr{L})$, where $\mathscr{U}$ is a set of users; $\mathscr{E}$ is a set of edges $e\langle u, v\rangle$ representing the friend relationship between users u and v, where u and $v \in \mathscr{U}$; $\mathscr{L}$ is a candidate locations list composed of all the user-generated locations.

Typically, a user u in Facebook might contribute various information, e.g., his basic profile information, friends, comments, photos. The core information of u in this chapter is his current city, denoted as $l(u)$. According to the accessibility of users' current city, the users are classified into two sets: current city available users (LA-users) and current city unavailable users (LN-users). We, respectively, use $\mathscr{U}^{LA}$ and $\mathscr{U}^{LN}$ to denote the sets of LA-users and LN-users, where $\mathscr{U} = \mathscr{U}^{LA} \cup \mathscr{U}^{LN}$.

To predict users' current city, we tend to exploit the users' location sensitive attributes and friends list. Assume that there exist m types of location sensitive attributes, denoted as $\mathscr{A} = \{a_1, a_2, \cdots, a_m\}$. Specifically, we denote a user u's location sensitive attributes as $\mathscr{A}(u) = \{a_1(u), a_2(u), \cdots, a_m(u)\}$. The users may also have a friends list, denoted as $\mathscr{F}(u)$, where $\mathscr{F}(u) = \{f \in \mathscr{U} \wedge e\langle u, f\rangle \in \mathscr{E}\}$. Therefore, we use a tuple to represent a user as $u : \langle l(u), \mathscr{A}(u), \mathscr{F}(u)\rangle$.

Additionally, we attempt to denote a location with a tuple of its unified *ID* (l_{id}), latitude and longitude coordinate. Therefore, a location can be written as a tuple: $l : \langle l_{id}, lat, lon\rangle$ and the candidate locations list can be denoted as a set of location tuples: $\mathscr{L} = \{l : \langle l_{id}, lat, lon\rangle\}_N$, where lat and lon respectively stand for the latitude and longitude of a location, and N is the number of candidate locations in the list.

Thus, the ***current city prediction problem*** can be formally stated as: *Given, (i) a graph* $\mathscr{G} = (\mathscr{U}^{LA} \cup \mathscr{U}^{LN}, \mathscr{E}, \mathscr{L})$*; (ii) the public location* $l(u)$ *for LA-users* $u \in U^{LA}$*; (iii) the location sensitive attributes* $\mathscr{A}(u)$ *and the friends list* $\mathscr{F}(u)$ *for all the users* $u \in (\mathscr{U}^{LA} \cup \mathscr{U}^{LN})$*, we predict current city* $\hat{l}(u)$ *for each LN-user* $u \in \mathscr{U}^{LN}$*, so as to make* $\hat{l}(u)$ *close to the user's real current city.*

Note that the current city of a user's friends can be either available ($f \in U^{LA}$) or unavailable ($f \in U^{LN}$). Thus, we introduce two notations to represent the two groups of friends: current city available friends (LA-friends) and current city unavailable friends (LN-friends). Let denote a user's LA-friends as $\mathscr{F}^{LA}(u)$ and LN-friends as $\mathscr{F}^{LN}(u)$, where $\mathscr{F}(u) = \mathscr{F}^{LA}(u) \cup \mathscr{F}^{LN}(u)$.

7.4 Overview of Current City Prediction

The goal of current city prediction is to correctly infer a coordinate point with latitude and longitude for a LN-user, given the candidate locations list $\mathscr{L}$ and the user's self-exposed information including

his location sensitive attributes and friends list. To achieve this goal, the basic idea is first to train a unified location indication model by extracting and integrating location indications from the given self-exposed information. This trained model is expected to estimate the probability of the given LN-user being at each location in the candidate locations list. Based on the candidate locations and the corresponding probabilities that are suggested by the model, a prediction approach is then proposed to properly select a location to be the predicted current city.

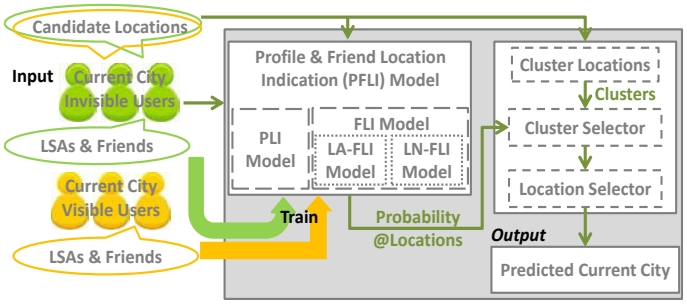


Figure 7.4: Framework of current city prediction (LSAs: location sensitive attributes)

Figure 7.4 illustrates the framework of our proposed current city prediction solution. To train the integrated model, we first separately consider the location indications from location sensitive attributes and friends, and consequently obtain two sub-models: profile location indication (*PLI*) model and friend location indication (*FLI*) model. Both *PLI* model and *FLI* model calculate a probability vector in which the element stands for the probability of a user being at a certain candidate location. Note that, *FLI* model leverages the location indications from both LA-friends and LN-friends. By integrating the probability vectors generated by *PLI* and *FLI* model with proper parameters, a unified profile and friends location indication (*PFLI*) model is derived.

For determining a current city for a LN-user based on the probabilities of all the candidate locations obtained by *PFLI* model, we use a two-step location selection strategy: cluster selection and location selection. Specifically, we first aggregate the nearby locations into a location cluster and obtain a set of location clusters. Then, we calculate the probability of a user being in a cluster by summing up the probabilities of all the candidate locations belonging to this cluster; the cluster with the highest probability is picked out as a candidate cluster. Finally, we try to select the 'best' location from the candidate cluster as the predicted current city.

In the next two sections, we will introduce how we set up the *PFLI* model and devise the current city prediction approach in detail.

7.5 Profile and Friend Location Indication Model

In this section, we describe the design of the probabilistic models which can suggest the probabilities of users being at all candidate locations. We first introduce the *profile location indication* (*PLI*) model; it estimates the probability merely relying on a user's location sensitive attributes. Then, we describe the *friend location indication* (*FLI*) model, which considers the location indications from a user's friends. Eventually, we integrate these two models together and obtain the integrated *profile and friend location indication* (*PFLI*) model.

7.5.1 Profile Location Indication Model

According to *Challenge 1* in Sec. 7.1, two problems should be concerned to construct *PLI* model. First, a certain value of a location sensitive attribute may indicate several locations. Therefore, for each attribute value, we need to consider its all possible location indications with the corresponding probabilities. Second, multiple location sensitive attributes (e.g., hometown, workplace, college) can be captured from a user's profile. This requires us to integrate various location indications extracted from different location sensitive attributes.

In order to capture the multiple possible location indications from one attribute value, we define a *location-attribute indication matrix* for each (k-th) location sensitive attribute $a_k \in \mathscr{A}$, denoted as $\mathscr{R}_k$. The rows of this matrix represent the candidate locations ($l \in \mathscr{L}$), while the columns stand for the possible values of a_k. We use l_i to represent the i-th candidate location and a_{k_j} to denote the j-th possible value of a_k. Then, a cell σ_k^{ij} in the matrix calculates the *indication probability* of a_{k_j} to l_i — the probability that a user, whose k-th location sensitive attribute a_k equals a_{k_j}, currently lives in the city l_i. Specifically, the indication probability equals the number of users who live in l_i and have a value of a_{k_j} divided by the total number of users who have a value of a_{k_j}. For instance, considering workplace, if 10 out of 100 employees from TELECOM SUDPARIS state that they live in EVRY in the whole data set, then the indication probability of TELECOM SUDPARIS to EVRY is 0.1. Note that, the j-th column of the $\mathscr{R}_k$ represents the multiple location indications of a_{k_j}.

Assume that a_k refers to M possible values except 'null'; N is the total number of the candidate locations. The k-th location-attribute indication matrix can be written as:

$$\mathscr{R}_k = \{\sigma_k^{ij}\}_{N\times M} = \{p(l(u) = l_i | a_k(u) = a_{k_j})\}_{N\times M}$$

Based on the location-attribute indication matrix ($\mathscr{R}$), we model the probability of a user's current city at l_i by combining all the user's available location sensitive attributes in his profile:

$$\begin{aligned} p_{Prof}(u, l_i) &= \sum_{a_k \in \mathscr{A}, a_k(u) \neq null} \alpha_k p(l(u) = l_i | a_k(u) = a_{k_j}) \\ &= \sum_{a_k \in \mathscr{A}, a_k(u) \neq null} \alpha_k \sigma_k(u, l_i) \end{aligned} \tag{7.3}$$

Where $\sigma_k(u, l_i)$ can be easily obtained by indexing the corresponding location-attribute indication matrix ($\mathscr{R}_k$) according to u's value of a_k ($a_k(u) = a_{k_j}$) and the given location (l_i), namely σ_k^{ij}; α_k is a parameter to adjust the significance of the different location sensitive attributes.

As we discussed in Sec. 7.3, not all a user's the attributes can be observed by public. Therefore, in Eq. 7.3, we merely consider the location sensitive attributes where the user publishes a value ($a_k(u) \neq null$). That means the indication probability of a user at any location is equal to zero if the user's attribute $a_k(u)$ is invisible. If all the location sensitive attributes are invisible for a user, we rely on the other information (e.g., his friends) to infer his current city, which we will discuss in the next section.

7.5.2 Friend Location Indication Model

Besides a user's location sensitive attributes, a plenty of location indications can be extracted from the user's friends. The existing work points out that around 92% of the crawled users from Twitter whose locations are also revealed in their relationships [89]; We find 87% of users' current city from

their friends' locations in our crawled Facebook data set. Hence, we exploit location indications from users' friends to construct the *FLI* model.

Since a user's friends can be either LA-friends (current city available) or LN-friends (current city unavailable), we take into account the location indications from both LA-friends and LN-friends to design *FLI* model. On one hand, we build up *FLI* model primarily depending on the location indications from LA-friends. On the other hand, we consider the location indications from LN-friends as a small regulator to modulate *FLI* model. Accordingly, *FLI* model contains two components: LA-friends location indication (*LA-FLI*) model and LN-friends location indication (*LN-FLI*) model.

7.5.2.1 LA-FLI Model

LA-FLI model differentiates the weights of a user's LA-friends and estimates the probability that he lives in a certain location (l_i) depending on the weights of the friends living in l_i. *LA-FLI* model attempts to assign a LA-friend with high weight if he is more likely to be in the same city to the user. However, *LA-FLI* model cannot directly determine which friends live in the same city to the user since the user's city is unknown. Therefore, to differentiate the friends' weights, *LA-FLI* model assesses the location similarity between a user and his friends according to the correlation between their location sensitive attributes.

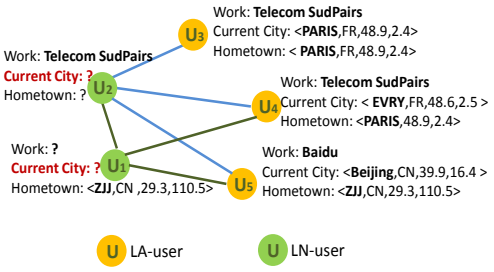


Figure 7.5: An example of social relations and profile information

We illustrate an example in Figure 7.5 and show that the location sensitive attributes can be used to distinguish the weights among various LA-friends. Focusing on LN-user u_2 and his LA-friends u_3, u_4 and u_5, we notice that u_2 and u_3, u_4 work in the same institute, while u_5 works in another company which is far away from u_2's workplace. In this case, it is natural to infer that u_2 is more likely to be living in the same city with u_3 and u_4 than with u_5; then u_3 and u_4 should be assigned with higher weights than u_5 because of the location similarity indicated by their workplace.

Inspired by the example, we construct an *attribute-based location similarity matrix* ($\mathscr{W}_k$) to estimate the location similarity between two users by each (k-th) location sensitive attribute ($a_k \in \mathscr{A}$). The rows and columns in the matrix are the possible values regarding a_k. The cells in the matrix w_k^{ij} calculate the *location similarity* of two users — the probability that the two users live in the same city — when they respectively have values of a_{k_i} and a_{k_j}. Specifically, we compute the total number of friend pairs where one user has a value of a_{k_i} and the other has a value of a_{k_j}, denoted as $|\{a_k(u) = a_{k_i} \wedge a_k(v) = a_{k_j}\}|$; Among these friend pairs, we further count the number of friend pairs where the two users live in the same city, denoted as $|\{l(u) = l(v) \wedge a_k(u) = a_{k_i} \wedge a_k(v) = a_{k_j}\}|$. Then, the attribute-based location similarity matrix is defined as:

$$\begin{aligned}\mathscr{W}_k &= \{w_k^{ij}\}_{M\times M} \\ &= \{p(l(u)=l(v)|a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j})\}_{M\times M} \\ &= \{\frac{|\{l(u)=l(v)\wedge a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j}\}|}{|\{a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j}\}|}\}_{M\times M}\end{aligned}$$

where M is the number of possible values of attribute a_k.

For a certain attribute a_k, assume that u and his LA-friend v have a value of a_{k_i} and a_{k_j} respectively. Then, the *location similarity* between u and v on a_k can be easily obtained by indexing the i-th row and j-th column of $\mathscr{W}_k$, denoted as $w_k(u,v) = w_k^{ij}, v \in F^{LA}(u)$. If the user u or his LA-friend v does not expose attribute a_k, $w_k(u,v) = 0$.

On the basis of the location similarity on a certain attribute, we combine all the location similarities on multiple location sensitive attributes with a set of trained parameters (β) to measure the weight of the LA-friend v. This combined attribute-based weight describes the probability that u and v live in the same city concerning all of their location sensitive attributes (e.g., work, hometown). v will be assigned to a large weight if he has a high probability to be in the same city with u.

Then, *LA-FLI* model calculates the probability of u living in l_i by integrating all the weights of u's LA-friends who live in l_i:

$$p_{LA-F}(u,l_i) = \sum_{v\in\mathscr{F}^{LA}(u)}\sum_{a_k\in\mathscr{A}} \beta_k w_k(u,v) p_{LA-U}(v,l_i) \tag{7.4}$$

where $p_{LA-U}(v,l_i)$ represents whether or not the LA-friend v living in l_i. It equals 1 if v states his current city is l_i; otherwise, it is 0:

$$p_{LA-U}(v,l_i) = \begin{cases} 1 & if\ l(v)=l_i \\ 0 & otherwise \end{cases}$$

7.5.2.2 LN-FLI Model

Before introducing LN-FLI model, we inspect the potential benefit of a user's LN-friends for his current city prediction with another example shown in Figure 7.5. We observe that u_2, being as a LN-friend of u_1, does not expose his current city; whereas, the workplace of u_2, TELECOM SUDPARIS, indicates two cities — PARIS and EVRY — according to the current cities of the users u_3 and u_4 who are also the employees of TELECOM SUDPARIS. Thereby, a user's LN-friends can also reveal some location indications in their exposed attributes, which may help the prediction.

Therefore, for a LN-friend (v), we first rely on his exposed location sensitive attributes and use *PLI* model (Sec. 7.5.1) to predict his current city, as:

$$p_{Prof}(v,l_i) = \sum_{a_k\in\mathscr{A}, a_k(v)\neq null} \alpha_k p(l(v)=l_i|a_k(v)=a_{k_j})$$

Taking all the LN-friends as equal, *LN-FLI* model integrates LN-friends' location indications and computes the probability that u lives in $l_i \in \mathscr{L}$ as follows:

$$p_{LN-F}(u,l_i) = \sum_{v\in F^{LN}(u)} p_{Prof}(v,l_i) \tag{7.5}$$

7.5.2.3 FLI Model

Eventually, primarily relying on *LA-FLI* model and being adjusted by *LN-FLI* model with a very small regulator parameter λ, *FLI* model estimates the probability that u currently lives at l_i by:

$$p_F(u,l_i) = p_{LA-F}(u,l_i) + \lambda p_{LN-F}(u,l_i) \tag{7.6}$$

7.5.3 Integrated Profile and Friend Location Indication Model

So far, we have introduced *PLI* model and *FLI* model, which abstract the probabilities of a user at various candidate locations, respectively, from his own location sensitive attributes and friends list. Then, we integrate them into a unified probabilistic location indication model, so as to capture complete location indications from two sides. Specifically, *PFLI* model calculates the probability of u living in $l_i \in \mathscr{L}$ as:

$$p(u,l_i) = \theta_P p_{Prof}(u,l_i) + \theta_F p_F(u,l_i) \tag{7.7}$$

7.5.3.1 Parameter Computation

To obtain a set of good parameters for the model, we first rewrite the model as:

$$\begin{aligned} p(u,l_i) &= \theta_P p_{Prof}(u,l_i) + \theta_F p_F(u,l_i) \\ &= \sum_{a_k \in \mathscr{A}} \theta_P \alpha_k \sigma_k(u,l_i) \\ &+ \sum_{a_k \in \mathscr{A}} \theta_F \beta_k \sum_{v \in F^{LA}(u)} w_k(u,v) p_{LA-F}(v,l_i) \\ &+ \sum_{a_k \in \mathscr{A}} \lambda \theta_F \sum_{v \in F^{LN}(u)} \alpha_k \sigma_k(v,l_i) \\ &= \sum_{a_k \in \mathscr{A}} \{[\mu_k \sigma_k(u,l_i) + \nu_k \delta_k(u,l_i)] + [\lambda_\alpha \eta_k(u,l_i)]\} \end{aligned} \tag{7.8}$$

where

- $\mu_k = \theta_P \alpha_k$; $\nu_k = \theta_F \beta_k$; $\lambda_\alpha = \lambda \theta_F$
- $\delta_k(u,l_i) = \sum_{v \in F^{LA}(u)} w_k(u,v) p_{LA-F}(v,l_i)$
- $\eta_k(u,l_i) = \sum_{v \in F^{LN}(u)} \sigma_k(v,l_i)$

As λ is a regulator parameter of very small value, we take the location indications extracted from a user's location sensitive attributes and his LA-friends as primary indications, while the part captured from the LN-friends as a micro-regulating indication. Thus, we compute the parameters by two separate steps. We first optimize the parameters μ_k and ν_k together for the main indication. We also try to find a set of local optimal parameters for *PLI* model to have a good regulating value from the LN-friends.

We train a good set of parameters μ_k and ν_k on a group of LA-users. For each LA-user, we consider all the locations l_i that indicated by either the user's location sensitive attributes or friends. In other words, the probability that the user lives in l_i is larger than zero, i.e., $\sum_{a_k \in \mathscr{A}} [\sigma_k(u,l_i) + \delta_k(u,l_i)] > 0$. According to each indicated location l_i, we generate an independent $\langle$label: features$\rangle$ item as $\langle \text{label}(l_i) : \sigma_1(u,l_i), \cdots, \sigma_m(u,l_i), \delta_1(u,l_i), \cdots, \delta_m(u,l_i) \rangle$ .In particular, we classify the indicated locations into

two groups according to their distances to u's actual location. We label an indicated location (l_i) as a *far* location (label$(l_i) = 0$), if its distance to u's actual location is larger than a pre-defined threshold; otherwise, we regard it as a *close* location (label$(l_i) = 1$); in addition, $\sigma_k(u,l_i) = \sigma_k^{ij}$ and $\delta_k(u,l_i) = \sum_{v \in F^{LA}(u)} w_k(u,v) p_{LA-F}(v,l_i)$, where $k \in [1,m]$. Based on the generated items, we use a logistic regression method to train the model in the following format:

$$f(y|\mathbf{x};\sigma_1,\cdots,\sigma_m,\delta_1,\cdots,\delta_m) = h_{\sigma,\delta}(\mathbf{x})^y(1-h_{\sigma,\delta}(\mathbf{x}))^{1-y}$$

where y is the label of the indicated location, $\mathbf{x}$ stands for the features and $h_{\sigma,\delta}(\mathbf{x})$ is the hypothesis function. Then we can apply gradient descent method to maximize $f(y|\mathbf{x};\sigma,\delta)$ and compute the parameters. Similarly, we can obtain a set of parameters for *PLI* model which works for the LN-friends.

7.6 Current City Prediction Approach

Based on the results of *PFLI* model — the corresponding probabilities of a user currently living in all the candidate locations, we devise a current city prediction approach in this section. Recall the example we illustrated in *Challenge 3* of Sec. 7.1. Assuming the *PFLI* model suggests that a user u has a probability of, respectively, 40% in BEIJING, 35% in PAIRS and 25% in EVRY which is very close to PARIS. In this case, u might live in the area around PAIRS and EVRY with a larger probability than BEIJING since the aggregated probability of u in the area around PAIRS and EVRY are higher than BEIJING. Therefore, rather than directly deal with the problem on a single-city [89][171][122][125], we aggregate the candidate locations which are very close to each other into a location cluster. We attempt to predict a user's current city by two steps: cluster selection and location selection. Refer to Figure 7.4, this prediction approach has been implemented with several main functions, including *Candidate Locations Cluster*, *Cluster Selector* and *Location Selector*. We are going to explain the main functions and illustrate how the prediction approach performs.

7.6.1 Candidate Locations Cluster

We draw on hierarchical clustering method [175] to generate location clusters. The hierarchical clustering method arranges all the candidate locations in a hierarchy with a treelike structure based on the distance between two locations, and successively merges the closest locations into clusters. Specifically, we first treat all the candidate locations as an independent location cluster and calculate the distance between any two candidate locations (*Step 1*). We find the closest pair of the location cluster and merge them into a new location cluster (*Step 2*). Then, we compute the average distance between the new cluster and each of the old ones (*Step 3*). We repeat the *Step 2* and *Step 3* until all the candidate locations are organized into one cluster tree. Eventually, we choose an ideal distance threshold (i.e., the average distance between any of two locations in the neighboring location clusters) to cut the cluster tree into clusters (*Step 4*).

Figure 7.6 illustrates an example of clusters on the user-generated candidate locations that locate in the area with latitude in $47°N \sim 49°N$ and longitude in $1°W \sim 6°E$. There exist 154 candidate locations in this area mentioned by users in our Facebook data set. With the hierarchical clustering method, we divide them into 7 location clusters which are marked in different color. We note several properties of our candidate locations clusters. First, all the regions are formed by clustering the user-generated locations according to their distances, instead of dividing areas with equal-sized grid cells [176] [177]. In our clustering, the areas that the users do not mention are out of consideration.

Second, the density inside the clusters are different; however, the average distances between all the candidate locations in any two neighboring clusters are equal ($100km$ in Figure 7.6). Third, the complexity of the *Step 2* and *Step 3* is $O(k^3)$ and the complexity of the *Step 4* is $O(2^k)$. Although the computation of location clustering is expensive, it can be preprocessed and only needs to be run once.

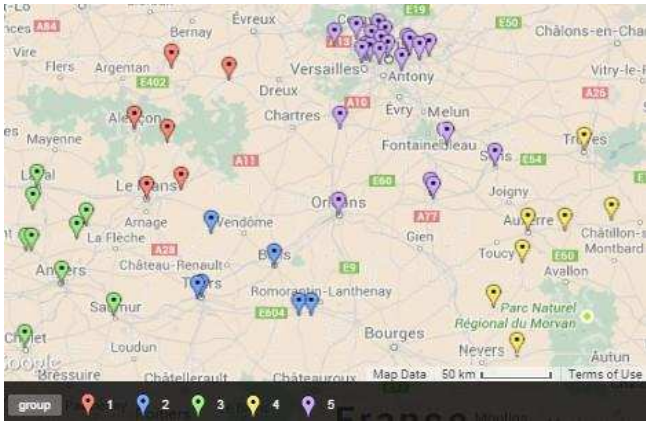


Figure 7.6: An example of candidate locations cluster

7.6.2 Cluster Selector

Cluster selector selects the best cluster for a user where the user may reside with the highest probability. We leverage the proposed *PFLI* model to obtain the user's probability living at each candidate location. The probability of a user locating in a cluster is equal to the aggregated probability of all the candidate locations inside the cluster. Therefore, for each cluster, we sum up the probabilities of its candidate locations and select the cluster with the highest probability.

7.6.3 Location Selector

Eventually, we tend to select a best point from the selected cluster for the user. Three alternatives are considered here. First, we select the point with the highest probability (i.e., *point of highest probability*) inside the selected cluster as the best point. Second, we consider the *geographic centroid* of the selected cluster as the user's best point. The geographic centroid is the average coordinate for all the points in a cluster while the probability of each point is considered as its weight. Third, we calculate the *center of minimum distance* which minimizes the overall distance form itself to all the rest of locations in a cluster. We will further discuss and compare the three methods in the experiment.

7.6.4 Implementation of Prediction Approach

Practically, each user only associates with a very limited number of locations compared to the total number of the candidate locations. Hence, according to the *PFLI* model, we do not calculate the probability for each candidate location and simplify the computation by three steps.

First, we initiate a probability vector (i.e., $\mathbf{p}_0(u)$) of candidate locations for a user, which merely includes the location indications from the user's location sensitive attributes and his LN-friends. Recall that the j-th column in the *location-attribute indication matrix* $\mathscr{R}_k$ stands for the probabilities

ALGORITHM 2: Current city prediction

Input: A LN-user u's location sensitive attributes;
u's friends list and friends' location sensitive attributes;
Location clusters set $\mathscr{C} = \{c_1, c_2, \cdots, c_s\}$ (s is the number of clusters);
Output: Predicted current city for u: $\langle lat, lon \rangle$;
Initiate the probability vector $\mathbf{p}_0(u)$ for u ***(Eq. 7.9)***;
Obtain all of LA-friends' current city $\mathscr{L}_{LA-F}$;
$\mathbf{p}(u) = \mathbf{p}_0(u)$;
for $l_i \in \mathscr{L}_{LA-F}$ **do**
  $\mathbf{p}(u, l_i) \leftarrow \mathbf{p}(u, l_i) + p_{LA-F}(u, l_i)$;
end
for $c_x \in \mathscr{C}$ **do**
  $p(u)_{c_x} = \sum_{l \in c_x} p(u, l)$
end
Cluster selection: c_h where $p(u)_{c_h} \geq p(u)_{c_x}, \forall c_x \in \mathscr{C}$;
Location selection from c_h ***(Sec. 7.6.3)***;
The predicted current city of u: $\langle lat, lon \rangle$

that a user lives in the corresponding locations if the user present a value of a_{k_j} in terms of the attribute a_k. We rewrite the indication matrix as $\mathscr{R}_k = [R_{\cdot k_1}, R_{\cdot k_2}, ..., R_{\cdot k_M}]$, where M is the number of possible values of a_k. Thus, we obtain the following initial probability vector:

$$\begin{aligned} \mathbf{p}_0(u) &= \sum_{a_k \in \mathscr{A}} [\mu_k \mathbf{p}(a_k(u) = a_{k_j}) + \lambda_\alpha \sum_{v \in F^{LN}} \mathbf{p}(a_k(v) = a_{k_j})] \\ &= \sum_{a_k \in \mathscr{A}} (\mu_k R_{\cdot k_j} + \lambda_\alpha \sum_{v \in F^{LN}} \alpha_k R_{\cdot k_j}) \end{aligned} \tag{7.9}$$

Second, we look at the location indications from the user's LA-friends. In fact, such indications correspond to two factors: LA-friends' current city and their weight. The current city of the user's LA-friends are aggregated into a set, denoted as $\mathscr{L}_{LA-F}$. According to the *LA-FLI* model, we can compute the probabilities that the user live in $l_i \in \mathscr{L}_{LA-F}$. As the number of the locations in $\mathscr{L}_{LA-F}$ is much smaller than the total number of candidate locations in $\mathscr{L}$, we can dramatically improve the computation rate.

Third, we add the location indications from LA-friends (i.e., $p_{LA-F}(u, l_i)$, where $l_i \in \mathscr{L}_{LA-F}$) to the right positions in the initial probability vector by indexing l_i in $\mathscr{L}$. Then, we have a current city probability vector (i.e., $\mathbf{p}(u)$) for the user, which is used to calculate the clusters' probabilities.

Eventually, obtaining the probabilities of u in all possible locations, we can easily compute the aggregated probability of u in each cluster and select the cluster with the highest probability. Furthermore, we predict a current city for u from the selected cluster by exploiting the *location selector*.

We summarize the current city prediction approach in Algorithm 2. In this algorithm, we assume that we have acquired the trained *PFLI* model with parameters $(\mu_k, \nu_k, \lambda_k)$, the candidate locations ($\mathscr{L}$), the location attribute indication matrices ($\mathscr{R}_k$), the attribute-based location similarity matrix ($\mathscr{W}_k$) and the location clusters set ($\mathscr{C}$). Then we try to predict the user u's current city according to u's self-exposed information. Finally, we obtain a location with its latitude and longitude.

7.7 Evaluation for Current City Prediction

In this section, we evaluate our proposed approach on a data set crawled from Facebook. We first introduce this data set, the compared approaches and the measurement. Then, we report the experiment results.

7.7.1 Experiment Setup

We are going to introduce the experiment setup through three aspects: data description, evaluation approaches and the measurements.

7.7.1.1 Data description

Among all 479,048 users in our data set, 173,027 users publicly report their current city (LA-users) and the rest 306,021 users do not reveal their current city (LN-users). In our evaluation, we use the 126,101 LA-users as the train and test set for the prediction. All the LN-users' information is also involved in the experiments, as the proposed approach considers the integrated location indications not only from a user' location sensitive attributes and LA-friends but also from LN-friends.

We extract a user's latest work or education experience as a location sensitive attribute, named 'Work and Education'; we also exploit a user's 'Hometown' as another location sensitive attribute. In our data set, 87,708 LA-users show their 'Hometown' and 54,097 LA-users expose 'Work and Education' to the public. In addition, 85,923 of the LA-Users publish their friends list.

7.7.1.2 Approaches

Based on our current city prediction model and the two-step location selection strategy (cluster selection; then location selection), we propose three cluster based prediction approaches with different location selectors. We tend to compare the performance of these approaches and determine a good location selector to obtain a prediction approach with high prediction accuracy. We also propose a non-cluster prediction approach based on our current city prediction model to evaluate the effectiveness of location cluster. Specifically, these model based approaches can be denoted as:

- $PFLI_{prob}$ is a cluster based approach which selects the *point of highest probability* from the selected cluster as the predicted location.
- $PFLI_{cent}$ is a cluster based approach which selects the *geographic centroid* from the selected cluster as the best prediction.
- $PFLI_{dist}$ is a cluster based approach which selects the *center of minimum distance* from the selected cluster.
- $PFLI_{noclst}$ is a non-cluster approach which selects the *point of highest probability* from all candidate locations as the predicted location.

Besides, we compare the model based approaches with several state-of-the-art prediction approaches:

- $Base_{dist}$ predicts a user's location based on the observation that the likelihood of friendship between two persons is decreasing with the distance [171].

Table 7.1: Prediction results (*AED*) for users with LA-Friends

Approach	$Base_{dist}$	$Base_{ann}$	$Base_{freq}$	$Base_{knn}$	$PFLI_{noclst}$	$PFLI_{dist}$	$PFLI_{cent}$	$PFLI_{prob}$
AED@60%	8.6	5.7	5.9	10.8	2.5	49.5	5.6	**2.1**
AED@80%	85.0	64.3	91.8	100.0	40.1	77.4	38.0	**36.9**
AED@100%	1288.5	1129.0	1160.5	1397.6	874.0	885.9	855.3	**854.4**

Table 7.2: Prediction results (*AED*) for overall users

Approach	$Base_{dist}$	$Base_{ann}$	$Base_{freq}$	$Base_{knn}$	$PFLI_{noclst}$	$PFLI_{dist}$	$PFLI_{cent}$	$PFLI_{prob}$
AED@60%	102.8	6.7	73.9	119.5	3.5	50.6	6.3	**3.1**
AED@80%	1368.8	74.7	1257.2	1429.6	52.5	88.2	50.2	**49.1**
AED@100%	2671	1204.0	2523.5	2698.5	981.0	989.9	960.8	**960.0**

- $Base_{ann}$ maps any location sensitive attribute value to a certain location and apply artificial neural network to train a current city prediction model.

- $Base_{freq}$ infers a user's location according to the location frequency extracted from his friends' location-specific tweets [121][122]. We borrow the idea of counting the frequency of locations that emerge in a user's friends and predict his current city by the most frequent location.

- $Base_{knn}$ also relies on the frequency idea for Twitter; however, it merely counts on a user's k closest friends who have the most common friends with him to compute the most frequent location [124][125].

Among the above approaches, $Base_{dist}$ and $Base_{ann}$ are originally devised for Facebook; while $Base_{freq}$ and $Base_{knn}$ for Twitter. We leverage the main ideas from $Base_{freq}$ and $Base_{knn}$, and modify them to fit our data set. By comparing our approach to $Base_{dist}$, $Base_{freq}$ and $Base_{knn}$ which mainly depend on friendships, we attempt to reveal the advantage of our approach: integrating location sensitive attributes. Using $Base_{ann}$, we verify the newly introduced one-attribute/multiple-locations mapping method.

7.7.1.3 Measurements

We exploit the same measurements that used in the existing work [121][122][126]: *Average Error Distance* (*AED*) and *Accuracy within K km* (*ACC@K*).

Error Distance of a user u's predicted result (i.e., $ErrDist(u)$) is defined as the distance in kilometers between the user's real location and his predicted location. *AED* averages the *Error Distances* of the overall evaluated users, denoted as $AED = \frac{\sum_{u \in U} ErrDist(u)}{|U|}$. In addition, we rank the users by their *Error Distance* in descending order and report *AED* of the top 60%, 80% and 100% of the evaluated users in the ranking list, denoted as *AED*@60%, *AED*@80% and *AED*@100% respectively [126].

Accuracy within K km reveals the percentage of users being predicted with an *Error Distance* less than K km. It can be represented as $ACC@K = \frac{|\{u|u \in U \wedge ErrDist(u) < K\}|}{|U|}$. $ACC@K$ shows the predication capability of an approach at a specific required *Error Distance*.

7.7.2 Experiment Results

Many relationship-based methods (e.g., $Base_{dist}$, $Base_{freq}$ and $Base_{knn}$) heavily rely on users' LA-friends whose locations are exposed. In general, such methods can work well for the users who have a certain number of LA-friends; but when they are applied to the overall users (either have or have not LA-friends), the performance decreases notably. We evaluate the prediction performance respectively on two user sets: *users with LA-friends* and *overall users*. And we report the evaluation results on *AED* and *ACC@K* subsequently.

7.7.2.1 Evaluation on *AED*

Table 7.1 and Table 7.2 show the *AED*s of all the compared approaches for two user sets respectively. We use bold font to highlight the shortest *AED* in the tables. From the results, we observe that the approaches based on our proposed *PFLI* model perform much better than all the other baselines. Among the model-based approaches, $PFLI_{prob}$, which selects the point with highest probability from the selected cluster, generates less *AED* than the other approaches with different location selectors; whereas the differences are quite small among all these model-based approaches. For instance, comparing $PFLI_{prob}$ and $PFLI_{cent}$, the differences of *AED@100%* are only 0.9 and 0.8 km for two user sets respectively. However, $PFLI_{prob}$ reduces the *AED* significantly compared to $Base_{ann}$ — the best baseline. This observation demonstrates that our integrated probabilistic model can better describe users' location than the other compared models.

Comparing the results of *AED@60%*, *AED@80%* and *AED@100%*, we notice that we can predict the top 60% and the top 80% of the users' current city at relatively small *AEDs* by our proposed approaches; While the *AEDs* increase by 10-23 times when we consider all of the users (*AEDs@100%*). It is similar when it comes to the other approaches: *AED@100%* is much larger than *AED@60%* and *AED@80%*. From the perspective of the approaches' capacity, this observation demonstrates that the approaches can predict most of the users' current city with a small *Error Distance*. While from the perspective of privacy, it implies that many users may be not security enough to hide their current city. We will discuss it further in the next section.

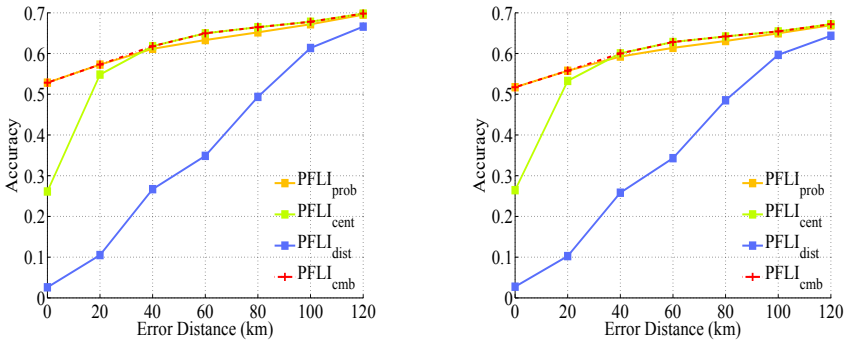
In addition, we notice that the *AED*s of $Base_{dist}$, $Base_{freq}$ and $Base_{knn}$ for the *overall users* are almost 2 times the values for *users with LA-friends*. However, for the *PFLI* model based approach, the *AED*s differ slightly for two user sets. For example, the *AED* of $PFLI_{prob}$ for *overall users* is only 74.4 km larger than the result for *users with friends*. Based on the evaluation comparisons on two users set, we can tell that, with the integrated location indications from users' profile and friends, our proposed prediction approaches is not constrained to users' LA-friends. Even for some users without knowing LA-friends in the *overall users*, our proposed approaches can still predict their location based on their profile and LN-friends.

Lastly, we compare the *AEDs* of the approaches using our proposed *PFLI* model. First, we compare $PFLI_{noclst}$ and $PFLI_{prob}$. $PFLI_{noclst}$ directly selects the location of the highest probability from the probability vector generated by *PFLI* model; while, relying on a cluster strategy, $PFLI_{prob}$ successively takes a cluster selection and a location selection which selects the location of the highest probability inside a selected cluster. The experiment results demonstrate that the cluster based approach outperforms the non-cluster based approach. Second, we investigate the cluster based approaches with different location selection solutions. From the results, $PFLI_{dist}$ generates the largest *AEDs* and $PFLI_{prob}$ achieves the smallest ones. This may suggest us a good solution — selecting the point with the highest probability — to select a location inside a cluster. We will further compare these three approaches on *ACC@K* and determine a good location selection solution to achieve a prediction approach with high accuracy in the next section.

7.7.2.2 Evaluation on *ACC@K*

In this section, we first study *ACC@K* of the three proposed prediction approaches with different location selectors, attempting to understand their strengths. Based on this study, we will develop a *combined-approach strategy* by combining the best prediction approaches under certain conditions, so as to obtain better performance than solely using any one of them.

Figure 7.7 compares the three proposed prediction approaches and plots *ACC@K*s at different *Error Distances* for two user sets in two subfigures. In both subfigures, we observe that the accuracy of $PFLI_{prob}$ goes up steadily with the increase of *Error Distance*. Compared to $PFLI_{prob}$, $PFLI_{cent}$ may lead very low accuracy when the required *Error Distance* is quite small; but it can achieve higher accuracy than $PFLI_{prob}$, when the *Error Distance* is larger than 40 km. It reveals the properties of these two prediction approaches: $PFLI_{cent}$ selects the geographic centroid of a cluster, which generates a short average *Error Distance* to all the locations in the cluster but loses chance to pick the user's exact coordinate once it is not the centroid; while $PFLI_{prob}$ might produce a large *Error Distance* if the location of the highest probability is not the user's real location. Besides, $PFLI_{dist}$ is not competitive with the other two approaches.



(a) Users with friends (b) Overall users

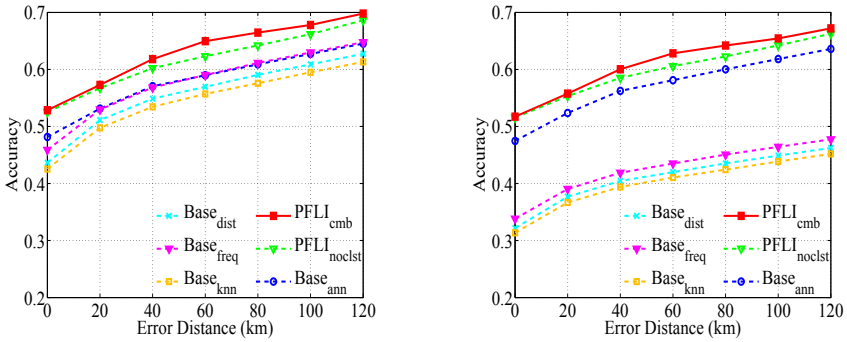
Figure 7.7: *ACC@K* of different location selectors

In this case, we propose a combined-approach strategy which uses $PFLI_{prob}$ when the required *Error Distance* is smaller than 40 km and otherwise applies $PFLI_{cent}$. We believe the combination is reasonable and practical. Because if third parties want to identify users according to their locations, they usually expect to identify users in a city or an area which allows certain *Error Distance*. Then, if a third party can tolerate a larger *Error Distance*, we can exploit $PFLI_{cent}$. Otherwise, we apply $PFLI_{prob}$. We also plot the combination line in Figure 7.7, named $PFLI_{cmb}$.

Figure 7.8 compares $PFLI_{cmb}$ to various baseline methods in terms of prediction accuracy. We observe that the proposed $PFLI_{cmb}$ outperforms all the compared baselines with the highest accuracy for both user sets. Compared to $PFLI_{noclst}$, $PFLI_{cmb}$ increases around 1.5% and 1.2% of accuracy on average respectively for *users with LA-friends* and *overall users*. It proves the effectiveness of the cluster strategy with successive cluster selection and location selection.

Comparing the results respectively for *users with LA-friends* and *overall users*, we observe a huge accuracy gap for $Base_{freq}$, $Base_{dist}$ and $Base_{knn}$. These approaches severely depend on friends' locations which lead to dramatic fall of performance when they are applied for users who do not have LA-friends. However, our proposed approaches integrating location sensitive attributes and friends (including our previous work $Base_{ann}$) can almost hold the prediction effectiveness for the *overall users*.

To summarize, first, we propose to combine $PFLI_{prob}$ and $PFLI_{cent}$ into a $PFLI_{cmb}$ approach



(a) Users with friends (b) Overall users

Figure 7.8: *ACC@K* of the proposed approach and other baselines

inspired by the experiment observations. $PFLI_{cmb}$ can flexibly change the prediction approach according to their performance under different required *Error Distances*. Second, our proposed approach outperforms the other compared baselines. Especially for the *overall users*, our proposed approach could gain 20% higher of accuracy than $Base_{dist}$ which is also a city prediction approach on Facebook.

7.8 Estimating Current City Exposure: Problem Statement

In this section, we pay attention to estimating the exposure probability of current city for a user who hides his current city. We formally formulate the current city exposure estimation problem as: *Given, (i) a graph* $\mathscr{G} = (\mathscr{U}^{LA} \cup \mathscr{U}^{LN}, \mathscr{E}, \mathscr{L})$*; (ii) the public location* $l(u)$ *for LA-users* $u \in U^{LA}$*; (iii) the location sensitive attributes* $\mathscr{A}(u)$ *and the friends list* $\mathscr{F}(u)$ *for all the users* $u \in (\mathscr{U}^{LA} \cup \mathscr{U}^{LN})$*; (iv) a required Error Distance K km, we forecast the current city exposure probability within K km and report exposure risk level for each LN-user* $u \in U^{LN}$.

To solve this problem, we run the proposed prediction approach on an aggregation of users and conduct analysis on the aggregated prediction results. Furthermore, we apply a regression method to construct the exposure model according to the analysis observations. Relying on this model, we devise a current city exposure estimator to tell a user the current city *Exposure Probability within K km* and *Exposure Risk Level*.

The *Exposure Probability within K km (EP@K)* represents the probability that a user' current city could be inferred correctly if the required *Error Distance* is *K* km. As it has the similar concept as the metric of *ACC@K*, we compute it by the same formula: $\frac{|\{u|u \in U \wedge ErrDist(u) < K\}|}{|U|}$.

Additionally, we set up 5 *Exposure Risk Levels* according to value of *Exposure Probability*, shown in Table 7.3. We regard *Level 5* as the most risky level which indicates an *Exposure Probability* larger than 0.9, while *Level 1* as the safe one which represents a small *Exposure Probability* less than 0.25.

Next, we first show some observations of inspections on the prediction for an aggregated users. Then we introduce the current city exposure model and the model based estimator. Finally, we illustrate some case studies to show the use of our proposed exposure estimator. We also summarize some guidelines to reduce the exposure risk.

Table 7.3: Risk level vs. exposure probability

Exposure Probability	$[0.9, 1]$	$[0.75, 0.9)$	$[0.5, 0.75)$	$[0.5, 0.25)$	$[0.25, 0]$
Risk Level	Level 5	Level 4	Level 3	Level 2	Level 1

Table 7.4: User categories by visible attributes combination

User's Visible Attributes	**Abbreviation**
'Hometown'	'HT'
'Work and Education'	'WE'
'Friends'	'F'
'Hometown' and 'Work and Education'	'HT+WE'
'Hometown' and 'Friends'	'HT+F'
'Work and Education' and 'Friends'	'WE+F'
'Hometown', 'Work and Education' and 'Friends'	'HT+WE+F'

7.9 Current City Exposure Inspection

Assume that we have run the proposed prediction approach on an aggregation of users whose current city is visible. We then obtain a collection of prediction results including users' self-exposed information, predicted current city and actual current city. We also develop some measurements to describe the characteristics of users' self-exposed information. Based on these prediction results, we can learn the correlation between the current city exposure probability and the measurable characteristics of users' self-exposed information.

First, we classify users into diverse categories with respect to the combinations of visible/invisible properties of their location sensitive attributes and friends list. Table 7.4 lists the obtained seven *User Categories*. *User Category* measures the types and amount of users' self-exposed information.

Figure 7.9 inspects the *Exposure Probabilities* for various *User Categories*. From this figure, we observe that different types of self-exposed information may divulge users' current city to different extent. For instance, users in 'WE' category are normally more dangerous to disclose their current city than users in 'HT' or 'F' category. We also find that the users who publish their 'WE' (in category 'WE', 'HT+WE', 'WE+F' and 'HT+WE+F') exhibit a high *Exposure Probability*. This means that 'WE' is a very risky attribute to leak users' current city. The results also reveal that 'HT' is more sensitive to disclose current city than 'F', although 'F' is generally regarded as a significant location indication.

Besides, generally speaking, Figure 7.9 displays that a user's current city could be predicted with a larger probability if the user exposes more information. For example, users who expose 'HT+F' exhibit a higher exposure probability than users only revealing either 'HT' or 'F'. Note that, for a user who exposes 'WE+HT', his current city exposure probability can be up to 90% which approaches to the exposure probability of users who expose 'HT+WE+F'. In other words, merely exposing 'WE+HT' but not 'F' can almost lead to the leakage of current city.

According to the results displayed in Figure 7.9, we conclude that *User Category*, distinguishing users by the types and amount of their self-exposed information, relates to *Exposure Probability*.

Apart from *User Category*, we define a new metric named *Exposure Coefficient*. It estimates the ratio of the probabilities of candidate locations in the selected cluster c_h to the overall probabilities of all the candidate locations (equal 1), calculated as follows:

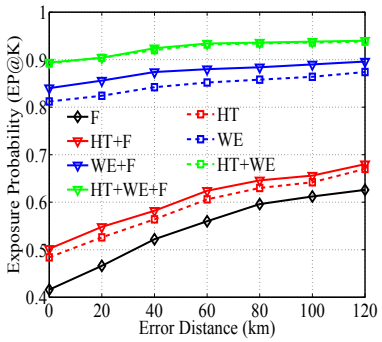
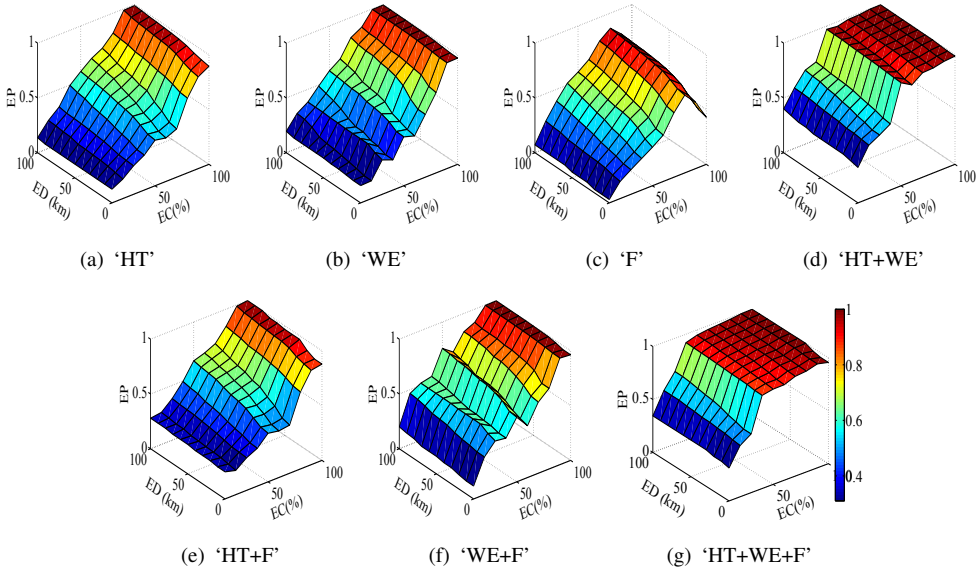


Figure 7.9: Current city exposure probability by user category



(a) 'HT' (b) 'WE' (c) 'F' (d) 'HT+WE'

(e) 'HT+F' (f) 'WE+F' (g) 'HT+WE+F'

Figure 7.10: Exposure probability by exposure coefficient in different user categories

$$EC(u) = \frac{\sum_{l \in c_h} p(u,l)}{\sum_{l \in \mathscr{L}} p(u,l)} = \sum_{l \in c_h} p(u,l)$$

Exposure Coefficient represents the centrality of the users' location indications. For example, *Exposure Coefficient* with a value of 100% means that all of a user's location indications point to an exclusive location cluster. We further look into the change of exposure probability according to *Exposure Coefficient* for each *User Category*.

Figure 7.10 reveals how *Exposure Probability* varies with diverse *Exposure Coefficient* and *Error Distances* in different *User Categories*. In this figure, each subfigure represents one *User Category*; the X, Y and Z axes in each subfigure are *Exposure Coefficient* (EC), *Error Distances* (ED) and *Exposure Probability* (EP) respectively. We observe that the *Exposure Probability* normally grows up when the *Exposure Coefficient* gets larger. When the *Exposure Coefficient* equals 100%, the *Exposure Probability* surpasses 90% within a required *Error Distance* of 20 km almost for all *User Categories*. This observation indicates that the current city is more dangerous to be predicted when a user's location indications are more likely to point to one city or to multiple cities that are in the

same cluster. In other words, a user's current city is easy to disclose if the centrality of the user's self-exposed information is high.

Note that, there exists an exception for the users only exposing their 'F': the decline of *Exposure Probability* when the *Exposure Coefficient* is larger than 0.9. One reasonable explanation is that only the users with an extremely small number of friends (e.g., only 1 friend) can have an *Exposure Coefficient* higher than 0.9, which might reduce the risk of current city exposure due to the limited information.

7.10 Estimating Current City Exposure Risk

In the previous section, we observe that the current city *Exposure Probability* for a user is influenced by three factors: *Error Distance*, *User Category* and *Exposure Coefficient*. According to the observation which are shown in Figure 7.10, we try to use a polynomial multiple regression method to model the relation among the current city *Exposure Probability*, *Exposure Coefficient* and *Error Distance* for each *User Category*. We can denote the model as: $y = fun_{x_1}(x_2, x_3)$, where x_1, x_2 and x_3 represent a user's *User Category*, *Exposure Coefficient* and *Error Distance* respectively; $fun_{x_1}(x_2, x_3)$ represents a polynomial function of *Exposure Coefficient* x_2 and *Error Distance* x_3 given the *User Category* x_1. y is the computed *Exposure Probability*.

By exploiting the proposed current city exposure model, we construct an exposure estimator to forecast the exposure risk of a user's private current city. Figure 7.11 illustrates the framework of current city exposure estimator. The exposure estimator contains three main function modules: user information handler, current city exposure model and exposure risk level decision. The inputs of the exposure estimator include a user's self-exposed information and a pre-established *Error Distance*. Given the user's self-exposure information, user information handler determines *User category* and computes *Exposure Coefficient*. Based on the pre-established *Error Distance*, the obtained *User category* and *Exposure Coefficient*, the exposure model calculates the current city exposure probability for the user. Exposure risk module determines a risk level according to the exposure probability. Eventually, the exposure estimator provides two risk measurements of current city: *Exposure probability* and *Risk Level*.

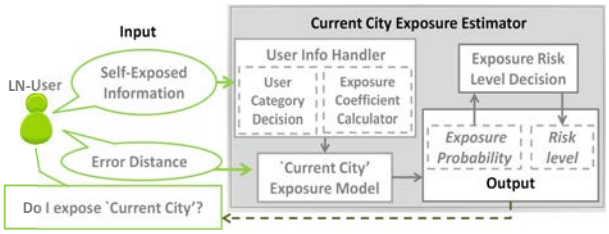


Figure 7.11: Framework of current city exposure estimator

7.11 Case Study: Exposure Estimator and Privacy Protection

Table 7.5 illustrates several use cases, where we estimate the *Exposure Probability* and *Risk Level* for some LN-users. In this study, we observe that some of the LN-users are not really safe to hide their current city if they leave some other information visible. For instance, considering $U7$, even only publishing 'EM', his current city is almost leaked with an extremely high *Exposure Probability* of 0.987 within an *Error Distance* of 20 km. In addition, for users in the same *User Category*, the

Table 7.5: Exposure estimator cases study

User	User Category	Exposure Coefficient	Error Distance	Exposure Probability	Risk Level
$U1$	'HT+WE+F'	0.491	100km	0.93	Level 5
$U1$	'HT+WE+F'	0.491	20km	0.88	Level 4
$U2$	'F'	0.905	100km	0.796	Level 4
$U3$	'F'	0.125	100km	0.128	Level 1
$U4$	'WE+F'	0.54	20km	0.461	Level 2
$U5$	'HT+F'	0.694	20km	0.683	Level 3
$U6$	'HT'	0.191	100km	0.254	Level 2
$U7$	'WE'	1	20km	0.987	Level 5

Table 7.6: Exposure guidelines for $U1$: the exposure risks if he adjusts some privacy configurations with an *Error Distance* of 100km

$U1$	**'HT+WE+F'**	**Hide 'WE'**	**Hide 'F'**	**Hide 'WE+F'**
Exposure Probability	0.93	0.46	0.906	0.436
Risk Level	Level 5	Level 2	Level 5	Level 2

ones who exist a higher *Exposure Coefficient* are more likely to divulge his current city. Looking at $U2$ and $U3$ who are both in 'F' category, the current city of $U2$ who exhibits an extremely high *Exposure Coefficients* is much more dangerous than $U3$'s current city.

In addition, the exposure estimator can give some countermeasures on privacy configuration to against information leakage. Assume users hide some part of their exposed information, the exposure estimator estimates and reports the corresponding *Exposure Probability* and *Exposure Risk Level*. Then users can decide a new privacy configuration accordingly. We take $U1$ as an example and list some possible exposure risks assuming that he adjusts his privacy configuration. The results shown in Table 7.6 reveal that the privacy could increase obviously if $U1$ hides his 'WE' or 'WE+F'. The results also point out that merely hiding 'F' could not protect $U1$'s current city privacy.

Eventually, according to the studies on the current city exposure risk, we summarize the following pieces of general suggestions:

- As all the location indications may expose the hidden current city, close all of location sensitive information including 'WE', 'F' and 'HT' so as to achieve a high current city security.

- Hide the most sensitive exposed information (e.g., 'WE') if users want to publicly share some personal information (e.g., 'F'), since the most sensitive information can independently lead to a quite high *Exposure Probability*. For example, 'WE' alone can lead an *Exposure Probability* higher than 80%.

- According to the centrality principle which refers to the *Exposure Coefficient*, Hide 'F' if most friends indicate the same place where the user lives. For instance, $U2$ in Table 7.5 is necessarily suggested hiding his 'F'.

7.12 Summary

This chapter starts with two open questions regarding the security of users' hidden privacy-sensitive attributes. To answer these questions, we first propose a novel current city prediction approach to infer users' current city by leveraging users' self-exposed information including location sensitive attributes and friends list. We validate the new prediction approach on our Facebook data set and the results reveal that the users' hidden current city may be dangerous to be predicted. Then we apply the proposed prediction approach to predict users' current city and model the exposure probability by *Exposure Coefficient* at different *Error Distances* for each *User Category*. Based on the exposure model, we propose a current city exposure estimator to measure the exposure probability and risk level of a user's hidden current city according to his self-exposed information. The exposure estimator can also help users to adjust their privacy configuration to satisfy their privacy intention. Note that, although this work studies the potential risk of users' privacy-sensitive attributes with a representative attribute of current city in Facebook, the proposed idea and approach could be extended to other attributes and utilized by other OSNs.

Chapter 8

Conclusion and Future Work

Contents

8.1 Conclusion

Since the largest OSN in the world, Facebook, was founded in 2004, OSNs have become a *de facto* portal for accessing the Internet for millions of users. Nowadays, there exist hundreds of OSNs all over the world, supporting a spectrum of practices, interests, and users, and attracting a lot of researchers to leverage the huge amount of digital information in OSNs to conduct research works across various areas including computer science, social science, economics, etc. With tens of millions of users worldwide, OSNs appear to be a new venue of innovation with many challenging research problems.

To gain new insights from OSNs, this dissertation focuses on investigating how user similarity can be leveraged comprehensively to facilitate a variety of applications, in order to address a wide range of critical issues in OSNs, containing interested content discovery, item recommendation, friend prediction, privacy protection, etc. Specifically, the contributions of this dissertation cover the whole life-cycle of the research process, including data collection, analysis and applications.

First, in order to establish concrete research results, this dissertation tries to obtain a large sample of OSN users from Facebook, the biggest OSN platform worldwide currently. In chapter 3, this dissertation presents the various web data collection approaches and describes the extracted 500K Facebook users' public profiles. Each user's profile includes three kinds of information: demographics, social relationships, and interests. According to the author's knowledge, our crawled data set is one of the largest Facebook data sets including rich user profile information up to date.

Based on this data set, from various perspectives, this dissertation then tends to study how to apply the knowledge of user similarity into a diversity of real-life applications to serve OSN users.

- Chapter 4 addresses the link prediction problem for new users who have not created any link, which can be used to recommend friends for new users to initialize their social networks. The basic idea is leveraging the users' profile attributes (e.g., workplace, high school and hometown) that can be easily obtained during the new users' sign up procedure to tackle the new-user link prediction problem. Based on the limited sign-up information, along with the attributes and links from existing users in OSN, three kinds of social features are identified: basic feature, derived feature and latent relation feature. By studying how the similarity of users regarding these features would indicate their friendship, a new user link prediction model is proposed by using all these learned social features. Extensive experimental results on our Facebook data set show that every learned social feature can effectively improve the link prediction performance for new users.

- Chapter 5, aiming at helping users to find their interested content in P2P network, leverages users' OSN profile information, including both their interests and social relationships, to improve the content discovery mechanism in P2P network. By connecting P2P network with OSN to establish a social P2P network model, a user can turn to his friends to query his interested resource, which probably has higher success rate than querying a stranger as friends tend to share more interests. More specifically, a user's different friends are assigned with different weights using random walk with restart, by considering the friend's own knowledge and the similarity with the user; thus, higher-weighted friends tend to have more probability to successfully answer a user's content discovery request. Experimental results show that considering the knowledge and similarity of a user's friend actually contributes to the content discovery in P2P network.

- Chapter 6, to verify the *homophily* that similar people will tend to have similar interests, carries out a comprehensive study on investigating the correlation between interest similarity and other attribute similarity in users' profiles, e.g., age, city. The social relationship between two users (i.e., whether they are friends) is also taken into account to see if it would affect the two users' interest similarity. This study confirms that the homophily does exist. Then, based on these analysis results, for a user whose interests are unknown, a practical prediction model is developed to identify some users who may share some interests with the given interest-unknown user. Finally, shown as a use case, this prediction model is applied into a item recommendation system to improve the recommendation performance for new users who have only little public information.

- Chapter 7 turns to protect the users' privacy-sensitive information. Specifically, it exploits users' location information, *current city*, as a concrete example of sensitive information to conduct this work. First, based on our data set, it observes that usually two users with similar profile attributes (e.g., work) and close social relationships (e.g., friends) would live in similar cities (i.e., same city or nearby cities). Inspired by this observation, it designs a prediction model to estimate a user's current city considering her other information. The estimation results of our proposed prediction model indicate that if a user exposes certain other information, then her hidden current city might still be inferred accurately by a third-party using her self-exposed information. Then, to help users to realize such privacy risk, some measurable characteristics are extracted from their self-exposed information, and an *exposure estimator* which can notify a user of her exposure risk level of the privacy-sensitive information (i.e., current city) is developed, due to the self-exposed information. Eventually, this chapter describes some use cases about how to leverage our exposure estimator to protect the users' sen-

sitive information (lower the risk level), via tuning the privacy settings of their self-exposed information.

To sum up, user similarity plays important roles in all the above four research applications. This dissertation comprehensively investigates the potential correlations between different user attributes and reveals that similar users tend to be similar in their interests, approximate in their geo-distance, or close in their social relationships. Relying on these observations and the obtainable information, this dissertation designs models and algorithms to address various critical issues in OSNs, such as item/friend recommendation for new users and privacy-sensitive location information protection.

8.2 Future Work

This dissertation has extensively studied the effects of user similarity in our Facebook data set and further introduces these effects to effectively enhance four specific applications and services, whereas our work is still subject to several limitations. In the following, some promising future research directions are discussed.

8.2.1 Data Fusion for Empirical Study

First of all, so far, this dissertation only relies on one data set of static information about user profile from Facebook. Apart from this information, Facebook contains a great deal of knowledge, e.g., user actions such as post, comment, and joining group, which can be used to more precisely measure user similarity and more efficiently augment the social applications and services. Besides, many other OSN platforms are flourishing and accumulating data with their own specialties. For instance, Twitter spread information in a 140-character short message with the location where it has been sent; Foursquare shares and gathers a great number of check-ins. Such massive information expanding within digital footprint can represent different facets of human behavior or characters. A variety of natural laws regarding similarity may be captured or the knowledge about homophily principle may be reinterpreted or enriched from different perspectives. Therefore, integrating the extensive data to study, analysis and model would be more interesting and conductive for the applications and services. Although data fusion can enrich our data sources, it also presents challenges with respect to how to collect and unify a user's information from multi-platforms, how much data to store, how much this will cost, and whether the users' expected privacy will be preserved during data integration.

8.2.2 Scalability of the Approach

In reality, the total number of users in OSNs is beyond 1 billion, let alone the amount of user-generated connections, interactions and content. However, this dissertation has not taken into account the practical scalability issue and only evaluated our proposed algorithm and approaches on a 500K user profile data set. How to apply our approach to billions of users' profiles might meet many challenging problems, probably needing the techniques from other research areas such as distributed system and parallel computing. Therefore, another direction for our future research is to adapt our proposed approaches to the scalable approaches that can be leveraged in real-life OSN platforms consisting of an enormous number of users.

8.2.3 Enriching other OSN Applications with Similarity

Inspired by the findings about similarity and homophily principle from social science, this dissertation has successfully explored and leveraged the effects of similarity to advance four specific applications and services in OSNs. While the four works cover certain important OSN issues such as item recommendation, friend prediction and privacy protection, there still exist a broad spectrum of other applications that may be enriched stemming from the intensive study of similarity. On one hand, novel OSN applications are emerging all the time, most of which could be facilitated by the investigation of user similarity. For instance, in the rapid-growing crowdfunding platforms, by studying the similarity between a project initiator with other successful/failed project initiators, some valuable insights could be obtained for the potential project supporters to determine whether the rewards of funding this project is deserved. On the other hand, some applications can be benefited by the similarity beyond two users. For example, by abstracting the similarity between users upto cities or even nations, it affords an opportunity to leverage OSN big data to automatically sketch a global culture map (which is traditionally created by time-consuming sample surveys) to reflect the commonality and distinction between different regions worldwide.

Appendix A

Parameter Optimization for Social P2P Network Model

To optimize the parameters of the social P2P network model, we minimize equation 5.7 with respect to the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The parameters are represented uniformly as $\boldsymbol{a}$ instead in this section. Therefore we calculate the derivative of equation 5.7 as:

$$\frac{\partial F(\boldsymbol{a})}{\partial \boldsymbol{a}} = 2\boldsymbol{a} + \sum_{k,r} \frac{\partial h(p_r - p_k)}{\partial \boldsymbol{a}} = 2\boldsymbol{a} + \sum_{k,r} \frac{\partial h(p_r - p_k)}{\partial (p_r - p_k)} (\frac{\partial p_r}{\partial \boldsymbol{a}} - \frac{\partial p_k}{\partial \boldsymbol{a}}) \tag{A.1}$$

Applying the commonly used hinge-loss function, i.e., $h(p_r - p_k) = [1 - (p_r - p_k)(\boldsymbol{a}^T(\boldsymbol{w}_{rv} - \boldsymbol{w}_{kv}))]_+$; thus we have $\frac{\partial h(p_r - p_k)}{\partial (p_r - p_k)} = [\boldsymbol{a}^T(\boldsymbol{w}_{rv} - \boldsymbol{w}_{kv})]_+$. To calculate $\frac{\partial p_u}{\partial \boldsymbol{a}}$, we obtain the initial probability vector at step 0 by sending queries to all the friends of the starting node and calculating the success rate. We denote the initial probability vector as $\boldsymbol{p^{(0)}}$. According to equation 5.5, we can iteratively compute the final probability given by:

$$\boldsymbol{p} = (1 - \delta)\boldsymbol{A'}\boldsymbol{p^{(0)}} \tag{A.2}$$

where $\boldsymbol{A'}$ is the final random walk transition probability matrix. Note that $\boldsymbol{p}$ is the principal eigenvector of matrix $\boldsymbol{A'}$. A.1 can be rewritten as $p_u = \sum_i p_i \boldsymbol{A'}_{iu}$. Therefore the derivative of p_u with respect to $\boldsymbol{a}$ equals:

$$\frac{\partial p_u}{\partial \boldsymbol{a}} = \sum_j A'_{iu} \frac{\partial p_i}{\partial \boldsymbol{a}} + p_i \frac{\partial A'_{iu}}{\partial \boldsymbol{a}} \tag{A.3}$$

By recursively employing the chain rule to A.3, we can compute the derivative of p_u iteratively [58] [178] [179] [180].

Eventually, we apply the gradient descent method to minimize $F(a)$ directly:

$$\boldsymbol{a} := \boldsymbol{a} - \mu \frac{\partial F(\boldsymbol{a})}{\partial \boldsymbol{a}}$$

Bibliography

[1] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[2] A. Schwering, "Approaches to semantic similarity measurement for geo-spatial data: A survey," *Transactions in GIS*, vol. 12, no. 1, pp. 5–29, 2008.

[3] J.-B. Lei, J.-B. Yin, and H.-B. Shen, "Gfo: a data driven approach for optimizing the gaussian function based similarity metric in computational biology," *Neurocomputing*, vol. 99, pp. 307–315, 2013.

[4] G. Tsebelis, "Decision making in political systems: Veto players in presidentialism, parliamentarism, multicameralism and multipartyism," *British journal of political science*, vol. 25, no. 03, pp. 289–325, 1995.

[5] D. H. Lee and P. Brusilovsky, "Social networks and interest similarity: The case of citeulike," in *HyperText*. ACM, 2010, pp. 151–156.

[6] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.

[7] G. Kossinets, "Effects of missing data in social networks," *Social networks*, vol. 28, no. 3, pp. 247–268, 2006.

[8] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *SIGSPATIAL*. ACM, 2010, pp. 458–461.

[9] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Stanford InfoLab, Tech. Rep. 2006-10, April 2006.

[10] C. Chelmis and V. K. Prasanna, "Social link prediction in online social tagging systems," *ACM Transactions on Information Systems*, vol. 31, no. 4, p. 20, 2013.

[11] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *WSDM*. ACM, 2012, pp. 703–712.

[12] X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver, "The wisdom of the few: A collaborative filtering approach based on expert opinions from the web," in *SIGIR*. ACM, 2009, pp. 532–539.

[13] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," in *ICDM*. IEEE Computer Society, 2010, pp. 971–976.

[14] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Intelligent Data Engineering and Automated Learning - IDEAL*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8206.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*. ACM, 2001, pp. 285–295.

[16] M. Goeksel and C. P. Lam, "System and method for utilizing social networks for collaborative filtering," mar 2010, uS Patent 7,689,452.

[17] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in *SIGCHI*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 210–217.

[18] C.-N. Ziegler and J. Golbeck, "Investigating interactions of trust and interest similarity," *Decision Support Systems*, vol. 43, no. 2, pp. 460–475, mar 2007.

[19] B. Markines and F. Menczer, "A scalable, collaborative similarity measure for social annotation systems," in *HyperText*, ser. HT '09. ACM, 2009, pp. 347–348.

[20] C. Cattuto, D. Benz, A. Hotho, and G. Stumme, "Semantic grounding of tag relatedness in social bookmarking systems," in *The Semantic Web - ISWC*. Springer Berlin Heidelberg, 2008, vol. 5318, pp. 615–631.

[21] R. Bunescu and Y. Huang, "A utility-driven approach to question ranking in social qa," in *Coling*. Association for Computational Linguistics, 2010, pp. 125–133.

[22] D. Lin, "An information-theoretic definition of similarity," in *ICML*. Morgan Kaufmann Publishers Inc., 1998, pp. 296–304.

[23] D. Hindle, "Noun classification from predicate-argument structures," in *28th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1990, pp. 268–275.

[24] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme, "Evaluating similarity measures for emergent semantics of social tagging," in *WWW*. ACM, 2009, pp. 641–650.

[25] J. Caverlee and S. Webb, "A large-scale study of myspace: Observations and implications for online social networks." in *ICWSM*, 2008.

[26] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, p. 056117, Nov 2009.

[27] N. Nguyen, T. Dinh, Y. Xuan, and M. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *INFOCOM*, April 2011, pp. 2282–2290.

[28] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," *SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 363–374, aug 2010.

[29] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *IJCAI - Volume 1*. Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.

[30] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," in *ECAI*, vol. 16. Citeseer, 2004, p. 1089.

[31] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 297–303, 2011.

[32] G. Pitsilis and W. Wang, "Harnessing the power of social bookmarking for improving tag-based recommendations," *CoRR*, vol. abs/1410.5072, 2014.

[33] Y. Xu, X. Guo, J. Hao, J. Ma, R. Y. K. Lau, and W. Xu, "Combining social network and semantic concept analysis for personalized academic researcher recommendation," *Decision Support Systems*, vol. 54, no. 1, pp. 564–573, dec 2012.

[34] A. O. Alves, F. Rodrigues, and F. C. Pereira, "Tagging space from information extraction and popularity of points of interest," in *Second International Conference on Ambient Intelligence*. Springer-Verlag, 2011, pp. 115–125.

[35] B. Markines, L. Stoilova, and F. Menczer, "Social bookmarks for collaborative search and recommendation," in *AAAI*, vol. 2006, 2006.

[36] V. Belák and V. Svátek, "Ontopolis. net: Social-semantic web application for participative e-democracy," *9th Znalosti Conference*, 2010.

[37] K. Yi, "An empirical study on the automatic resolution of semantic ambiguity in social tags," *American Society for Information Science and Technology*, vol. 48, no. 1, pp. 1–10, 2011.

[38] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Transactions on the Web*, vol. 6, no. 2, pp. 9:1–9:33, jun 2012.

[39] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic detection of semantic similarity," in *WWW*. ACM, 2005, pp. 107–116.

[40] L. Weng, A. Flammini, A. Vespignani, and F. Menczer, "Competition among memes in a world with limited attention," *Scientific Reports*, vol. 2, 2012.

[41] L. Weng and F. Menczer, "Emergent semantics from game-induced folksonomies," in *First International Workshop on Crowdsourcing and Data Mining*. ACM, 2012, pp. 1–9.

[42] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[43] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *SIGKDD*. ACM, 2011, pp. 1100–1108.

[44] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici, "Link prediction in social networks using computationally efficient topological features," in *PASSAT and SocialCom*, Oct 2011, pp. 73–80.

[45] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," in *INFOCOM*, April 2011, pp. 1647–1655.

[46] M. Makrehchi, "Social link recommendation by learning hidden topics," in *RecSys*. ACM, 2011, pp. 189–196.

[47] V. Vasuki, N. Natarajan, Z. Lu, and I. S. Dhillon, "Affiliation recommendation using auxiliary networks," in *RecSys*. ACM, 2010, pp. 103–110.

[48] H.-N. Kim and A. El Saddik, "Exploring social tagging for personalized community recommendations," *User Modeling and User-Adapted Interaction*, vol. 23, no. 2-3, pp. 249–285, 2013.

[49] P. Symeonidis and C. Perentis, "Link prediction in multi-modal social networks," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 147–162.

[50] L. Dong, Y. Li, H. Yin, H. Le, and M. Rui, "The algorithm of link prediction on social network," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[51] M. Franceschet, "Pagerank: Standing on the shoulders of giants," *Communications of the ACM*, vol. 54, no. 6, pp. 92–101, jun 2011.

[52] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *WWW*. Elsevier Science Publishers B. V., 1998, pp. 107–117.

[53] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *CIKM*. ACM, 2003, pp. 556–559.

[54] H. H. Song, B. Savas, T. W. Cho, V. Dave, Z. Lu, I. S. Dhillon, Y. Zhang, and L. Qiu, "Clustered embedding of massive social networks," in *SIGMETRICS*. ACM, 2012, pp. 331–342.

[55] V. Leroy, B. B. Cambazoglu, and F. Bonchi, "Cold start link prediction," in *SIGKDD*. ACM, 2010, pp. 393–402.

[56] D. Quercia and L. Capra, "Friendsensing: recommending friends using mobile phones," in *RecSys*. ACM, 2009, pp. 273–276.

[57] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel, "Co-authorship networks in the digital library research community," *Information processing & management*, vol. 41, no. 6, pp. 1462–1480, 2005.

[58] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM*. ACM, 2011, pp. 635–644.

[59] M. Sirivianos, K. Kim, and X. Yang, "Socialfilter: introducing social trust to collaborative spam mitigation," in *INFOCOM*. IEEE, 2011, pp. 2300–2308.

[60] H. Zhang, C. Xu, and J. Zhang, "Exploiting trust and distrust information to combat sybil attack in online social networks," in *Trust Management VIII*. Springer, 2014, pp. 77–92.

[61] D. Ramage, A. N. Rafferty, and C. D. Manning, "Random walks for text semantic similarity," in *Workshop on Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics, 2009, pp. 23–31.

[62] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies: Search and ranking," in *3rd European Conference on The Semantic Web: Research and Applications*, 2006, pp. 411–426.

[63] C. Cattuto, D. Benz, A. Hotho, and G. Stumme, "Semantic analysis of tag similarity measures in collaborative tagging systems," *CoRR*, vol. abs/0805.2045, 2008.

[64] F. Göbel and A. Jagers, "Random walks on graphs," *Stochastic Processes and Their Applications*, vol. 2, no. 4, pp. 311–336, 1974.

[65] H. Tong, C. Faloutsos, and Y. Koren, "Fast direction-aware proximity for graph mining," in *SIGKDD*. ACM, 2007, pp. 747–756.

[66] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, sep 1999.

[67] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, mar 2007.

[68] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *SIGKDD*. ACM, 2010, pp. 243–252.

[69] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *SIGCOMM*. ACM, 2009, pp. 267–278.

[70] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *SIGCOMM*. ACM, 2009, pp. 322–335.

[71] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *ICDM*. IEEE Computer Society, 2006, pp. 613–622.

[72] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 4, no. 3, pp. 173–184, dec 2010.

[73] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *SIGKDD*. ACM, 2002, pp. 538–543.

[74] L. Cao, B. Cho, H. D. Kim, Z. Li, M.-H. Tsai, and I. Gupta, "Delta-simrank computing on mapreduce," in *1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. ACM, 2012, pp. 28–35.

[75] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing web search using social annotations," in *WWW*. ACM, 2007, pp. 501–510.

[76] J. Yu, X. Jin, J. Han, and J. Luo, "Collection-based sparse label propagation and its application on social group suggestion from photos," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 2, pp. 12:1–12:21, feb 2011.

[77] L.-j. NING and H.-y. DUAN, "An algorithm for friend-recommendation of social networking sites based on simrank and ant colony optimization," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, pp. 79–87, 2014.

[78] D. B. Thi and T.-A. N. Hoang, "Features extraction for link prediction in social networks," in *ICCSA*. IEEE, 2013, pp. 192–195.

[79] K. Zolfaghar and A. Aghaie, "Evolution of trust networks in social web applications using supervised learning," *Procedia Computer Science*, vol. 3, pp. 833–839, 2011.

[80] R. Jin, V. E. Lee, and H. Hong, "Axiomatic ranking of network role similarity," in *SIGKDD*. ACM, 2011, pp. 922–930.

[81] T. Maehara, M. Kusumoto, and K.-i. Kawarabayashi, "Efficient simrank computation via linearization," in *SIGKDD*. ACM, 2014, pp. 1426–1435.

[82] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *WWW*. ACM, 2010, pp. 61–70.

[83] "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4772–4778, 2010.

[84] Z. Li and H. Shen, "Social-p2p: An online social network based p2p file sharing system," in *ICNP*, 10 2012, pp. 1–10.

[85] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *WOSN*, 2009, pp. 37–42.

[86] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *SIGKDD*. ACM, 2011, pp. 1082–1090.

[87] L. Liu, N. Antonopoulos, and S. Mackin, "Managing peer-to-peer networks with human tactics in social interactions," *Journal of Supercomputing*, vol. 44, no. 3, pp. 217–236, 6 2008.

[88] C. Suen, S. Huang, C. Eksombatchai, R. Sosic, and J. Leskovec, "Nifty: A system for large scale information flow tracking and clustering," in *WWW*, 2013, pp. 1237–1248.

[89] R. Li, S. Wang, and K. C.-C. Chang, "Multiple location profiling for users and relationships from social network and content." *PVLDB*, vol. 5, no. 11, pp. 1603–1614.

[90] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *IMC*. ACM, 2007, pp. 29–42.

[91] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *CoRR*, vol. abs/1111.4503, 2011.

[92] J. Leskovec and E. Horvitz, "Planetary-scale views on a large instant-messaging network," in *WWW*. ACM, 2008, pp. 915–924.

[93] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[94] H. Shen, L. Zhao, H. Chandler, J. Stokes, and J. Li, "Toward p2p-based multimedia sharing in user generated contents," in *INFOCOM*. IEEE, 2011, pp. 667–675.

[95] T. Chen and L. He, "Collaborative filtering based on demographic attribute vector," in *ICFCC*, June 2009, pp. 225–229.

[96] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 2, pp. 179–192, 2010.

[97] M. Krishna Ramanathan, V. Kalogeraki, and J. Pruyne, "Finding good peers in peer-to-peer networks," in *Vehicle Navigation and Information Systems Conference*, Oct 1993, p. 8.

[98] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, jun 2005.

[99] R. Lambiotte, V. D. Blondel, C. de Kerchove, E. Huens, C. Prieur, Z. Smoreda, and P. Van Dooren, "Geographical dispersal of mobile communication networks," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 21, pp. 5317–5325, 2008.

[100] Y. Jia, Y. Wang, X. Jin, and X. Cheng, "Tsbm: The temporal-spatial bayesian model for location prediction in social networks," in *International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 2. IEEE, 2014, pp. 194–201.

[101] J. Leskovec and E. Horvitz, "Geospatial structure of a planetary-scale social network," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 3, pp. 156–163, 2014.

[102] Y. Volkovich, S. Scellato, D. Laniado, C. Mascolo, and A. Kaltenbrunner, "The length of bridge ties: Structural and geographic properties of online social interactions." in *ICWSM*, 2012.

[103] K. Lewis, M. Gonzalez, and J. Kaufman, "Social selection and peer influence in an online social network," *National Academy of Sciences*, vol. 109, no. 1, pp. 68–72, 2012.

[104] L. Gou, F. You, J. Guo, L. Wu, and X. L. Zhang, "Sfviz: Interest-based friends exploration and recommendation in social networks," in *VINCI*. ACM, 2011, pp. 15:1–15:10.

[105] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks." *ICWSM*, vol. 11, pp. 329–336, 2011.

[106] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Transactions on the Web*, vol. 5, no. 1, pp. 5:1–5:44, Feb. 2011.

[107] L. Ge and A. Zhang, "Pseudo cold start link prediction with multiple sources in social networks." in *SDM*. SIAM, 2012, pp. 768–779.

[108] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *INFOCOM*, vol. 3, Mar 2003, pp. 2166–2176.

[109] Y. Zhang, G. Shen, and Y. Yu, "Lips: Efficient p2p search scheme with novel link prediction techniques," in *ICC*, june 2007, pp. 1875–1880.

[110] P. A. Chirita, A. Damian, W. Nejdl, and W. Siberski, "Search strategies for scientific collaboration networks," in *workshop on Information retrieval in peer-to-peer networks*. ACM, 2005, pp. 33–40.

[111] M. Sanchez-Artigas and B. Herrera, "Socialhelpers: Introducing social trust to ameliorate churn in p2p reputation systems," in *P2P*, Sept 2011, pp. 328–337.

[112] Z. Wang, L. Sun, S. Yang, and W. Zhu, "Prefetching strategy in peer-assisted social video streaming," in *MM*. ACM, 2011, pp. 1233–1236.

[113] H. Wang, F. Wang, J. Liu, C. Lin, K. Xu, and C. Wang, "Accelerating peer-to-peer file sharing with social relations," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 66–74, September 2013.

[114] A. Fast, D. Jensen, and B. N. Levine, "Creating social networks to improve peer-to-peer networking," in *SIGKDD*. ACM, 2005, pp. 568–573.

[115] G. Ruffo and R. Schifanella, "A peer-to-peer recommender system based on spontaneous affinities," *ACM Transaction on Internet Technology*, vol. 9, no. 1, pp. 4:1–4:34, Feb 2009.

[116] C. Doulkeridis, K. Nørvåg, and M. Vazirgiannis, "Peer-to-peer similarity search over widely distributed document collections," in *LSDS-IR*. ACM, 2008, pp. 35–42.

[117] "Fault-tolerant peer-to-peer search on small-world networks," *Future Generation Computer Systems*, vol. 23, no. 8, pp. 921–931, 2007.

[118] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *WSDM*, ser. WSDM '11, 2011, pp. 287–296.

[119] T. Chen and L. He, "Collaborative filtering based on demographic attribute vector," in *ICFCC*, 2009, pp. 225–229.

[120] A.-T. Nguyen, N. Denos, and C. Berrut, "Improving new user recommendations with rule-based induction on cold user data," in *RecSys*, pp. 121–128.

[121] S. Chandra, L. Khan, and F. Muhaya, "Estimating twitter user location using social interactions: A content based approach," in *SocialCom*, 2011, pp. 838–843.

[122] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating twitter users," in *CIKM*, pp. 759–768.

[123] Y. Ikawa, M. Vukovic, J. Rogstadius, and A. Murakami, "Location-based insights from the social web," in *WWW Companion*, 2013, pp. 1013–1016.

[124] S. Abrol and L. Khan, "Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining," in *SocialCom*, 2010, pp. 153–160.

[125] S. Abrol, L. Khan, and B. Thuraisingham, "Tweecalization: Efficient and intelligent location mining in twitter using semi-supervised learning," in *CollaborateCom*, 2012, pp. 514–523.

[126] 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 *SIGKDD*, 2012, pp. 1023–1031.

[127] T. Pontes, M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida, "We know where you live: privacy characterization of foursquare behavior," in *UbiComp*, 2012, pp. 898–905.

[128] L. Lu and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[129] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM'06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[130] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 437–452.

[131] M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1872–1892, 2011.

[132] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *WWW*, 2010, pp. 591–600.

[133] J. Zhang, X. Kong, and P. S. Yu, "Predicting social links for new users across aligned heterogeneous social networks," in *ICDM*, 2013, pp. 1289–1294.

[134] M. Yan, J. Sang, T. Mei, and C. Xu, "Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge," in *ICME*, 2013, pp. 1–6.

[135] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location hertory," *ACM Transactions on the Web*, vol. 5, no. 1, p. 5, 2011.

[136] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Transactions on the Web*, vol. 6, no. 2, p. 9, 2012.

[137] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[138] D. Caragea, V. Bahirwani, W. Aljandal, and W. H. Hsu, "Ontology-based link prediction in the livejournal social network." in *SARA*, vol. 9, 2009.

[139] J. McAuley and J. Leskovec, "Discovering social circles in ego networks," *TKDD*, vol. 8, no. 1, pp. 4:1–4:28, 2014.

[140] P. A. Networks, "The application usage and risk report," 2012.

[141] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web," in *STOC*. ACM, 1997, pp. 654–663.

[142] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM*. ACM, 2001, pp. 149–160.

[143] P. Maymounkov and D. Mazi?res, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*, ser. Lecture Notes in Computer Science, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Springer Berlin Heidelberg, 2002, vol. 2429, pp. 53–65.

[144] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys Tutorials*, vol. 7, no. 2, pp. 72–93, quarter 2005.

[145] Gnutella, "Gnutella protocol development," 2003.

[146] S. Ferretti, "Gossiping for resource discovering: An analysis based on complex network theory," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1631–1644, 2013.

[147] Y. Zhang and L. Liu, "Distance-aware bloom filters: Enabling collaborative search for efficient resource discovery," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1621–1630, 2013.

[148] L. Liu, J. Xu, D. Russell, P. Townend, and D. Webster, "Efficient and scalable search on scale-free p2p networks," *Peer-to-Peer Networking and Applications*, vol. 2, no. 2, pp. 98–108, 2009.

[149] S. Milgram, "The small world problem," *Psychology Today 1*, pp. 61–67, 1967.

[150] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna, "Four degrees of separation," *CoRR*, vol. abs/1111.4570, 2011.

[151] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM*. ACM, 2011, pp. 635–644.

[152] J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. V. Bonilla, E. Abbasnejad, and N. Della Penna, "New objective functions for social collaborative filtering," in *WWW*. ACM, 2012, pp. 859–868.

[153] H. Bao and E. Y. Chang, "Adheat: an influence-based diffusion model for propagating hints to match ads," in *WWW*. ACM, 2010, pp. 71–80.

[154] L. Liu, N. Antonopoulos, and S. Mackin, "Social peer-to-peer for resource discovery," in *PDP*, feb. 2007, pp. 459–466.

[155] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, jul 2003.

[156] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[157] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012.

[158] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *WSDM*. ACM, 2012, pp. 703–712.

[159] Z. Wen and C.-Y. Lin, "On the quality of inferring interests from social neighbors," in *SIGKDD*, 2010, pp. 373–382.

[160] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[161] S. Loh, F. Lorenzi, R. Granada, D. Lichtnow, L. K. Wives, and J. P. M. de Oliveira, "Identifying similar users by their scientific publications to reduce cold start in recommender systems," in *WEBIST*, 2009, pp. 593–600.

[162] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[163] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[164] P. Singla and M. Richardson, "Yes, there is a correlation:-from social networks to personal behavior on the web," in *WWW*, 2008, pp. 655–664.

[165] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *SIGIR*, 2002, pp. 253–260.

[166] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: learning new user preferences in recommender systems," in *IUI*, 2002, pp. 127–134.

[167] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *WPES*, 2005, pp. 71–80.

[168] R. Dey, Z. Jelveh, and K. Ross, "Facebook users have become much more private: A large-scale study," in *PERCOM Workshop*, 2012, pp. 346–352.

[169] R. Farahbakhsh, X. Han, Á. Cuevas, and N. Crespi, "Analysis of publicly disclosed information in facebook profiles," in *ASONAM*, 2013, pp. 699–705.

[170] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida, "Beware of what you share: Inferring home location in social networks," in *ICDM Workshop*, 2012, pp. 571–578.

[171] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *WWW*, 2010, pp. 61–70.

[172] M. Duckham and L. Kulik, "Location privacy and location-aware computing," *Dynamic & mobile GIS: investigating change in space and time*, vol. 3, pp. 35–51, 2006.

[173] M. A. STELZNER, "How marketers are using social media to grow their businesses," 2014.

[174] K. Ryoo and S. Moon, "Inferring twitter user locations with 10 km accuracy," in *WWW Companion*, 2014, pp. 643–648.

[175] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*.

[176] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, "Geographic routing in social networks," *Proceedings of the National Academy of Sciences*, vol. 102, no. 33, pp. 11 623–11 628, 2005.

[177] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the gap between physical location and online social networks," in *UbiComp*, 2010, pp. 119–128.

[178] A. L. Andrew, "Iterative computation of derivatives of eigenvalues and eigenvectors," *IMA Journal of Applied Mathematics*, vol. 288, pp. 209–218, 1979.

[179] Y. Jin, Y. Matsuo, and M. Ishizuka, "Ranking learning entities on the web by integrating network-based features," in *Mining and Analyzing Social Networks*, ser. Studies in Computational Intelligence, I.-H. Ting, H.-J. Wu, and T.-H. Ho, Eds. Springer Berlin Heidelberg, 2010, vol. 24(2), pp. 107–123.

[180] A. L. Andrew, "Convergence of an iterative method for derivatives of eigensystems," *Journal of Computational Physics*, vol. 26, pp. 107–112, jan 1978.