MINING USER-GENERATED GEO-SOCIAL DATA FOR
SEARCH AND RECOMMENDATION

A Dissertation in
Computer Science and Engineering
by
Mao Ye

© 2012 Mao Ye

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2012
The dissertation of Mao Ye was reviewed and approved by the following:

Wang-Chien Lee  
Associate Professor of Computer Science and Engineering  
Dissertation Advisor, Chair of Committee

Tom La Porta  
Professor of Computer Science and Engineering

Piotr Berman  
Associate Professor of Computer Science and Engineering

Xiaolong “Luke” Zhang  
Associate Professor of Information Science and Technology

Raj Acharya  
Head and Professor of Computer Science and Engineering

*Signatures are on file in the Graduate School.*
Abstract

With the increasing availability of GPS-enabled smart phones, rapid development of location-based services, and growing interests in on-line social networking, a number of location-based social networking (LBSN) services such as Facebook Places, Foursquare, Whrrl, Yelp, EveryTrail and TripAdvisor have emerged. These services allow users to explore places, write reviews and blogs, and share their locations and experiences with others. In this thesis, we propose to mine those user-generated geo-social data (e.g., places, reviews, blogs and social links) to enable better search and recommendation services through several innovative techniques.

First, we develop a travelogue service that discovers and conveys various travelogue (or trip blog) digests, in form of theme locations, geographical scope, traveling trajectory and location snippet, to users. In this service, theme locations in a travelogue are the core information to discover. Thus we aim to address the problem of theme location discovery to enable the above travelogue services. Due to the inherent ambiguity of location relevance, we perform location relevance mining (LRM) in two complementary angles, relevance classification and relevance ranking, to provide comprehensive understanding of locations. Furthermore, we explore the textual (e.g., surrounding words) and geographical (e.g., geographical relationship among locations) features of locations to develop a co-training model for enhancement of classification performance. Built upon the mining result of LRM, we develop a series of techniques for provisioning of the aforementioned travelogue digests in our travelogue system.

Second, we develop a semantic annotation technique for location-based social networks (e.g., Foursquare and Whrrl) to automatically annotate all places with category tags which are a crucial prerequisite for location search. Our annotation algorithm learns a binary support vector machine (SVM) classifier for each tag in the tag space to support multi-label classification. Based on the check-in behavior of users, we extract features of places from i) explicit patterns (EP) of individual places and ii) implicit relatedness (IR) among similar places. The features extracted from EP are summarized from all check-ins at a specific place. The features from IR are derived by building a novel network of related places (NRP) where similar places are linked by virtual edges. Upon NRP, we determine the probability of a category tag for each place by exploring the relatedness of places. Both EP and IR features are complementary with each other and beneficial for the proposed classification task.

Third, we provide a point-of-interests (POIs) recommendation service for the rapid grow-

1point-of-interests (POIs) are also referred as places in our context of location-based social networks.
ing location-based social networks (LBSNs), e.g., Foursquare, Whrrl, etc. The idea is to explore user preference, social influence and geographical influence for POI recommendations. In addition to deriving user preference based on user-based collaborative filtering and exploring social influence from friends, we put a special emphasis on geographical influence due to the spatial clustering phenomenon exhibited in user check-in activities of LBSNs. We argue that the geographical influence among POIs plays an important role in user check-in behaviors and model it by power-law distribution. Accordingly, we develop a collaborative recommendation algorithm based on naive Bayesian method, by incorporating geographical influence. Furthermore, We propose a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence.

Finally, social friendship has been shown beneficial for item recommendation for years. However, existing approaches mostly incorporate social friendship into recommender systems by heuristics. Here, we argue that social influence between friends can be captured quantitatively and propose a probabilistic generative model, called social influenced selection (SIS), to model the decision making of item selection (e.g., what book to buy or where to dine). Based on SIS, we mine the social influence between linked friends and the personal preferences of users through statistical inference. To address the challenges arising from multiple layers of hidden factors in SIS, we develop a new parameter learning algorithm based on expectation maximization (EM). Moreover, we show that the mined social influence and user preferences are valuable for group recommendation and viral marketing.

We conduct extensive empirical studies to evaluate the performance of our proposed approaches. The experiment results demonstrate the effectiveness of our approaches and their superiority over state-of-the-art approaches in corresponding domains.
# Table of Contents

List of Figures .......................... ix

List of Tables .......................... xi

Acknowledgments ........................ xii

Chapter 1

Introduction .................................. 1
  1.1 Research Challenges and Opportunities in LBSNs .................. 2
  1.2 Our Proposals ................................ 4
  1.3 The Organization of the Thesis ............................... 5

Chapter 2

Literature Review ............................ 6
  2.1 GPS Trajectory Data ............................ 8
    2.1.1 Location Extraction .......................... 8
    2.1.2 Transportation Model Inference .................. 9
    2.1.3 Human Mobility Model .......................... 9
    2.1.4 Location/Movement Prediction .................... 10
    2.1.5 Urban Planning ............................ 10
  2.2 Geo-associated Text Data .................... 11
    2.2.1 Location Entity Extraction .......................... 11
    2.2.2 Spatial Information Retrieval .................. 12
  2.3 Data from Emerging LBSNs .................. 12
    2.3.1 Spatio-Social Analysis .......................... 13
    2.3.2 User’s Location Detection .................. 14
    2.3.3 Location Type Classification .................. 14
Chapter 5
Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation

5.1 Overview ........................................... 68
5.2 Preliminaries ........................................ 71
  5.2.1 User-based Collaborative Filtering ............... 71
  5.2.2 Friend-based Collaborative Filtering .......... 71
  5.2.3 Related Work .................................. 73
5.3 Recommendation via Geographical Influence ............. 74
5.4 Unified Collaborative POI Recommendation .......... 77
  5.4.1 Fusion Framework ............................... 77
  5.4.2 Check-in Probability Score Estimation ........... 78
5.5 Empirical Evaluation ................................ 79
  5.5.1 Dataset Description ............................. 79
  5.5.2 Performance Metrics ............................ 81
  5.5.3 Evaluated Recommendation Approaches .......... 82
  5.5.4 Tuning Parameters .............................. 83
  5.5.5 Performance Comparison ......................... 83
  5.5.6 Study on Item-based CF .......................... 86
  5.5.7 Study on Social Influence ....................... 86
  5.5.8 Impact of Data Sparsity ........................ 88
  5.5.9 Test for Cold Start Users ....................... 89
5.6 Summaries ........................................... 89

Chapter 6
Exploring Social Influence for Recommendation - A Generative Model Approach

6.1 Overview ........................................... 92
6.2 Related Work ........................................ 93
6.3 Social Influenced Selection ........................... 95
  6.3.1 The SIS Model .................................. 96
  6.3.2 Parameter Learning Algorithm .................... 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Applications of Social Influence</td>
<td>101</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Group Recommendation</td>
<td>101</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Viral Marketing</td>
<td>104</td>
</tr>
<tr>
<td>6.5</td>
<td>Performance Evaluation</td>
<td>105</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Dataset Description</td>
<td>106</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Item Recommendation</td>
<td>107</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Topic Analysis</td>
<td>109</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Social Influence Study</td>
<td>110</td>
</tr>
<tr>
<td>6.5.5</td>
<td>User Profile Analysis</td>
<td>112</td>
</tr>
<tr>
<td>6.5.6</td>
<td>Group Recommendation</td>
<td>113</td>
</tr>
<tr>
<td>6.5.7</td>
<td>Viral Marketing</td>
<td>115</td>
</tr>
<tr>
<td>6.6</td>
<td>Summaries</td>
<td>116</td>
</tr>
</tbody>
</table>

Chapter 7

Conclusions and Future Works | 118 |

Bibliography | 121 |
List of Figures

2.1 A process of data mining ........................................... 8

3.1 A paragraph in a travelogue for Siesta Key. Relevant locations are in bold, partial relevant locations are in teletype, and irrelevant locations are in italic. 19

3.2 Dilemma in relevant location classification ........................................ 23

3.3 Partonomy hierarchy of locations mentioned in the travelogue shown in Figure 3.1 ........................................ 25

3.4 Co-training framework for location relevance classification .............. 26

3.5 Travelogue service ........................................................................ 31

3.6 Location relevance classification ....................................................... 36

3.7 Challenge in location relevance classification ........................................... 37

3.8 LLE with varied λ ......................................................................... 38

3.9 Improvement with different feature groups ............................................. 38

3.10 Theme location ............................................................................... 40

3.11 Geographical scope ......................................................................... 40

3.12 Test on TS ...................................................................................... 42

3.13 Evaluation on location snippet ............................................................. 42

4.1 Users and places in an LBSN ............................................................... 45

4.2 Distr. of # of visitors ....................................................................... 49

4.3 Distr. of Max # of check-ins by a single visitor .................................. 49

4.4 Distr. of check-in time (day) ............................................................... 50

4.5 Distr. of check-in time (hour) ............................................................. 50

4.6 Entropy distribution ........................................................................ 52

4.7 Graph representations of LBSN data .................................................... 52

4.8 Performance comparison .................................................................... 57

4.9 Impact of β ..................................................................................... 60

4.10 Impact of η .................................................................................... 61

4.11 Impact of k .................................................................................... 62

4.12 Impact of mark-off rate, x% ............................................................... 64

4.13 Discrete Hours Vs. Smoothed Hours ................................................. 65
5.1 Graph representation of user-user friendship and user-location check-in activity in a LBSN ................................................. 68
5.2 Geographical influence probability distribution ......................... 75
5.3 Tuning parameters .................................................................. 80
5.4 Performance comparison .......................................................... 81
5.5 Fusion of U and L ................................................................. 84
5.6 Fusion of L and G ................................................................. 84
5.7 Social tie and its influence implication ........................................ 85
5.8 Impact of Data Sparsity ............................................................ 87
5.9 Performance for cold start users ................................................ 88

6.1 The SIS model .................................................................... 96
6.2 Decompose an arbitrary group into a set of two-member groups. ........ 103
6.3 Testing topic sizes (last.fm) .................................................... 107
6.4 Testing topic sizes (whrrl.com) ............................................... 107
6.5 Performance comparisons (last.fm) .......................................... 108
6.6 Performance comparisons (whrrl.com) ....................................... 108
6.7 Example topics learned from whrrl.com dataset, based on SIS .......... 109
6.8 Example topics learned from last.fm dataset, based on SIS ............ 110
6.9 Social influence analysis .......................................................... 111
6.10 Example user profiles and social influence learned from last.fm dataset, by using SIS ................................................................. 112
6.11 Example user profiles and social influence learned from whrrl.com dataset, by using SIS ................................................................. 113
6.12 Group recommendation (whrrl.com) .......................................... 114
6.13 A subgraph of social network labeled with social influence and user preference. 115
List of Tables

2.1 Different Categories of Data ................................................. 7
3.1 Features and their categories ............................................. 24
3.2 Location snippet extraction for each individual travelogue. .......... 41
4.1 Categories and their percentages (z\%) .................................. 56
4.2 Optimal $\beta$ settings under $\eta = 0.2$ and $k = 5$ .................. 59
5.1 Optimal parameter settings .................................................. 83
6.1 Datasets statistics ......................................................... 106
6.2 Comparison of viral marketing strategies ............................... 115
Acknowledgments

I would like to express my sincere gratitude to Dr. Wang-Chien Lee, Associate Professor in the Department of Computer Science and Engineering, who has been my advisor since the beginning of my PhD study. He provided a lot of invaluable advice and guidance in my research, and gave help and encouragement when I encountered difficulties during the course of my studies.

Besides, I would like to take this chance to thank Dr. Xiaolong Luke Zhang, Dr. Piotr Berman, and Dr. Tom La Porta for their time and effort serving as my PhD thesis examination committee members and for important comments that helped improve the quality of this thesis. In addition, I appreciate having the honor to work with Dr. Bernardo A. Huberman, Dr. Thomas Sandholm, Dr. Christina Aperjis, Dr. Xing Xie, Dr. Rong Xiao, Dr. Dik Lun Lee, Dr. Ken C.K. Lee, Dr. Krzysztof Janowicz, Dr. Xueyan Tang, Dr. De-Nian Yang, Dr. Jianliang Xu, Mr. Chunyan Wang, Mr. Peifeng Yin, Mr. Xingjie Liu, Mr. Greg Ference, Mr. Dong Shou, Mr. Christoph Milligann, Miss Yuan Tian, Mr. Huajing Li, and Mr. Congxing Cai in those years, sharing research ideas and conducting high quality research together.

My special appreciation is to my family who always encouraged me and believed in me. Last but not least, my special thanks goes to my wife, Li Wang for her love and continuous support. Without her encouragement, many tasks of mine including this thesis would not have been accomplished.
Chapter 1

Introduction

Rapid technological advances towards “Web 2.0” have led to the growth of user-generated content [1, 2] in recent years. This trend has appeared in a wide range of new web applications such as Wikis [3, 4, 5], forums [6, 7, 8], blogging sites [9, 10, 11], and content sharing services [12, 13, 14, 15]. Among user-generated data, e.g., blogs, commentaries, photos and videos, many of them contain location information, explicitly or implicitly, along with other rich semantic information. For example, Flickr [12] allows users to semantically and geographically tag the photos and organize them according to locations. EveryTrail [16] allows travelers to share photos and journals along with GPS trajectories. Yelp [17] lets users exchange reviews of point-of-interests (POIs), such as merchants and restaurants. Recently, a growing number of location-based social networking services (LBSNs)\(^1\), e.g., Foursquare [18], Gowalla [19], Whrrl [20], etc, have also emerged. They allow users to connect with friends, explore places (e.g., restaurants, stores, theaters, etc), share their locations, and upload photos, videos, and blogs. Different from conventional social networking services that connect people merely in the cyber world, these LBSNs bring people together via both cyber connections and “physical” interactions with places. More importantly, the interactions between users and places have been captured in the user-generated geo-social data in these services.

The aforementioned user-generated geo-social data represents a wealth of rich information resources, which are valuable for scientific research and innovative applications. As mentioned earlier, many systems and sites allow users to establish cyber links to their friends or other users, create tags, post comments and tips to share with other users. For example, OpenStreetMap provides rich information about road map and POIs; Foursquare and Gowalla con-

\(^1\)These services are often referred as location based social networks and thus abbreviated as LBSNs.
tain plenty of footprints of socially connected users; and Yelp provides a good platform for users to share reviews and ratings of places. In addition to the rich location information and social connections noted above, one important trend is the sharing of personal views, opinions, observations, and experiences expressed in various forms (e.g., liked/disliked votes, ratings, commentary articles, travelogues\(^2\) and photographs). The collective intelligence embedded in the shared individual views is a very valuable resource for many realistic applications in our life and scientific studies. For example, many personalized and location-aware search and recommendation services could be provisioned:

- a tour planning site (e.g., TripAdvisor [21]) may facilitate Bob to find useful travelogues to plan the trip of visiting the Museum of Modern Art in New York;
- a location-based social networking site (e.g., Yelp [17]) may provide a search interface for Alice to find a Chinese restaurant near the Fifth Avenue in New York;
- a restaurant reservation site (e.g., OpenTable [22]) may recommend a French restaurant to Bob and Cindy for a romantic Friday evening dinner;
- a local group organization portal (e.g., Meetup.com [23]) may suggest a community park or event for a group of pug lovers.

### 1.1 Research Challenges and Opportunities in LBSNs

Generally speaking, the aforementioned services aim to provide search or recommendation for places or things associated with places (i.e., location-dependent items) to individuals or groups. Compared with conventional Web 1.0 data, searching or recommending user-generated geo-social data is definitely more challenging, which is mainly attributed to the following factors:

- **Unstructured and low quality data.** User-generated geo-social data is generated by volunteered users, which means the quality of data is inherently not guaranteed. For example, due to the nature of human language presentation, locations mentioned in the travelogue are not necessarily theme locations of the travelogue. In order to facilitate travelogue retrieval and mine the knowledge of (1) “whereabout” of the travelogue, and

\(^2\)They record travel experiences, on weblogs, forums and social communities for travels
(2) “how-about” of the mentioned attractions from the travelogue, it is important to automatically detect and tag the theme locations for a given travelogue. In addition, based on our analysis of data collected from Whrrl and Foursquare, about 30% of all places are lacking any meaningful textual descriptions, which are crucial for assisting users in searching and exploring new places as well as for developing recommendation services [24, 25, 26]. Thus, techniques representing data better, e.g., data tagging, are urgently needed in order to provide high quality user-generated geo-social data search services.

- **Rich and heterogeneous data.** Recent developments of location-based social networking systems introduce new dimensions (such as social and geographical dimensions) into the user-generated data. These rich social and geographical information provide good opportunities to better understand users’ personal interests, which are beneficial in provision of location-aware services such as recommendation in LBSNs. Furthermore, novel techniques that can harvest the information sources such as user’s previous transactional history, intelligence from crowd source, influence from friends and content of items to enable effective personalized search and recommendation services are highly desired.

In this thesis, we tackle the above challenges to provide search and recommendation services upon user-generated geo-social data.

First, regarding the problem of theme location detection for travelogues, we consider the information such as where and how frequent the location entity is mentioned in the travelogue, and whether the location entities mentioned in the travelogue are spatially-clustered are important for theme location detection. Intuitively, theme locations of travelogue are usually mentioned in the title and more frequently, and spatially-clustered as well. We consider user check-in behavior in location-based social networks such as Foursquare and Whrrl has patterns. For example, college places are usually checked in by users between Monday and Thursday; while drink places are usually visited by people between Friday and Sunday. Intuitively, the check-in temporal distribution for a place is useful in determining the semantic type of the places, which can facilitate place data cleaning and tagging.

Second, in our daily life, we may turn to our friends for opinions on books, movies or restaurants. Therefore social influence among friends is beneficial for search and recommendation. Furthermore, users tend to visit nearby places; thus the factor of geographical proximity is important for location-aware services as well. Therefore, for a given user, a place which is
suggested by friends and also nearby would be a good choice.

1.2 Our Proposals

First, to facilitate travelogue retrieval, we develop a travelogue service that discovers and conveys various travelogue digests, in the form of theme locations, geographical scope, traveling trajectory and location snippet, to users. In this service, theme locations in a travelogue are the core information to discover. Thus we aim to address the problem of theme location discovery to enable the above travelogue services. Due to the inherent ambiguity of location relevance, we perform location relevance mining (LRM) in two complementary angles, relevance classification and relevance ranking, to provide comprehensive understanding of locations. Furthermore, we explore the textual (e.g., surrounding words) and geographical (e.g., geographical relationship among locations) features of locations to develop a co-training model for enhancement of classification performance. Built upon the mining result of LRM, we develop a series of techniques for provisioning of the aforementioned travelogue digests in our travelogue system.

Second, we develop a semantic annotation technique for location-based social networks (e.g., Foursquare and Whrrl) to automatically annotate all places with category tags which are a crucial prerequisite for location search, recommendation services, or data cleaning. Our annotation algorithm learns a binary support vector machine (SVM) classifier for each tag in the tag space to support multi-label classification. Based on the check-in behavior of users, we extract features of places from i) explicit patterns (EP) of individual places and ii) implicit relatedness (IR) among similar places. The features extracted from EP are summarized from all check-ins at a specific place. The features from IR are derived by building a novel network of related places (NRP) where similar places are linked by virtual edges. Upon NRP, we determine the probability of a category tag for each place by exploring the relatedness of places. Both EP and IR features are complementary with each other and beneficial for the proposed classification task.

Third, we provide a point-of-interests (POI) recommendation service for the rapid growing location-based social networks (LBSNs), e.g., Foursquare, Whrrl, etc. The idea is to explore user preference, social influence and geographical influence for POI recommendations. In addition to deriving user preference based on user-based collaborative filtering and exploring social influence from friends, we put a special emphasis on geographical influence due to the
spatially-clustering phenomenon exhibited in user check-in activities of LBSNs. We argue that the geographical influence among POIs plays an important role in user check-in behavior and model it by a power-law distribution. Accordingly, we develop a collaborative recommendation algorithm based on geographical influence based on naive Bayesian. Furthermore, We propose a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence.

Finally, social friendship has been shown to be beneficial for item recommendation for years. However, existing approaches mostly incorporate social friendship into recommender systems by heuristics. Here, we argue that social influence between friends can be captured quantitatively and propose a probabilistic generative model, called social influenced selection (SIS), to model the decision making of item selection (e.g., what books to buy or where to dine). Based on SIS, we mine the social influence between linked friends and the personal preferences of users through statistical inference. To address the challenges arising from multiple layers of hidden factors in SIS, we develop a new parameter learning algorithm based on expectation maximization (EM). Moreover, we show that the mined social influence and user preferences are valuable for group recommendation and viral marketing.

1.3 The Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 surveys existing research related to user-generated geo-social data. Chapter 3 addresses the problem of theme location discovery to enable the travelogue services that discovers and conveys various travelogue digests, in form of theme locations, geographical scope, traveling trajectory and location snippet to users. Chapter 4 develops a semantic annotation technique for location-based social networks to automatically annotate all places with category tags which are a crucial prerequisite for location search, recommendation services, or data cleaning. Chapter 5 provides a point-of-interests (POI) recommendation service for the rapid growing location-based social networks (LBSNs), by exploring user preference, social influence and geographical influence. Chapter 6 proposes a probabilistic generative model, called social influenced selection (SIS), to model the decision making of item selection. Finally, Chapter 7 concludes the thesis and states our future research directions.
Chapter 2

Literature Review

In this chapter, we review the research literature on mining user-generated geo-social data, where user-generated geo-social data includes GPS trajectory data, geo-associated text data and data generated from emerging LBSNs as discussed below.

GPS Trajectory Data. Geographic information has spawned many novel Web applications. Among them, global positioning system (GPS) plays an important role in bridging the applications and end users. As part of this Web 2.0 trend, various social and scientific activities, e.g., reality mining [27, 28], geo-life [29] and sociallight [30], have started to make impacts on scientific studies and our society. GPS trajectory data collected from the above projects are usually obtained by using global positioning system (GPS) to record user’s movement. A GPS Trajectory consists of a series of GPS sampling points, where each point logs the user’s current location and corresponding time stamp.

Geo-associated Text Data. Continued flourish of World Wide Web (WWW) and advancement of transportation systems have made travel, an integral part of our life, very easy. Particularly, with the advantages of Web 2.0 technology, many people are willing to share their travelogues, which record travel experiences, on weblogs, forums and social communities for travels, (e.g., TravelPod\(^1\), IgoUgo\(^2\) and TravelBlog\(^3\)). Travelogues, which usually contain rich travel information such as the tours, lodging, meals, expenses, weather conditions, and so on, are highly valuable to a trip planner. Thus, many people turn to on-line travelogue archives or discussion forums to review travelogues before taking their trips. Here geo-associated text data refers a

\(^{1}\)http://www.travelpod.com
\(^{2}\)http://igougo.com
\(^{3}\)http://www.travelblog.com
text document which contains spatial information or location entities, such as the words “New York”.

**Data from Emerging LBSNs.** Recently, a growing number of location-based social networking services (LBSNs), e.g., Facebook Place, Foursquare, Flickr, Yelp, etc, have emerged. They allow users to connect with friends, explore places (e.g., restaurants, stores, theaters, etc), share their locations, and upload photos, videos, and blogs. Different from conventional social networking services that connect people merely in the cyber world, these LBSNs bring people together via both cyber connections and “physical” interactions with places. More importantly, the interactions between users and places have been nicely captured in the data generated from these services. Particularly, data from emerging LBSNs include social graph data, tweets from twitter, place data from Foursquare, photo data from Flickr, and so on.

In Table 2.1, we compare the above three categories of data in accordance with their content types: user profile information, social network, geo-coordinates (i.e., longitude and latitude) and text. For example, geo-associated text data is usually in a form of a document, where text is the main part of data content. Another example, in emerging LBSNs such as Foursquare, we collect data including the information such as users information, social networks, check-in places and comments to the corresponding places.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Social-Geo Content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>user profile</td>
</tr>
<tr>
<td>Geo-associated Text Data, e.g., travelogues and world news</td>
<td>maybe</td>
</tr>
<tr>
<td>GPS Trajectory Data, e.g., reality mining [27, 28], geo-life [29] data</td>
<td>maybe</td>
</tr>
<tr>
<td>Data from Emerging LBSNs, e.g., Foursquare and Twitter</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Table 2.1.** Different Categories of Data

The process of mining those user-generated geo-social data usually consists of four (or five) steps as shown in Figure 2. With the collected raw data, we first perform data preprocessing (e.g., data cleaning, sampling and normalization) to transform data to a better form. Then we apply data mining algorithms over the transformed data to uncover the patterns and hidden structure in the data. With techniques such as visualization and validation, we interpret the
results and discover the knowledge. Besides knowledge discovery, we can also develop applications such as search and recommendation. As we can see, different kinds of data may need different data preprocessing methods to transform data. In addition, different kinds of data correspond to different kinds of application scenarios, where different application scenarios have their own research problems which require us to devise different data mining algorithms. In the following, we review the related literature which is classified based on categories of data types used in research (as listed in Table 2.1).

![Figure 2.1. A process of data mining](image)

2.1 GPS Trajectory Data

As aforementioned, global positioning systems (GPS) play an important role in bridging the applications and end users. Learning knowledge from users’ raw GPS trajectory data can provide rich context information for both geographic and mobile applications.

2.1.1 Location Extraction

Location extraction from users’ GPS trajectories help profile users spatial behaviors, which is beneficial for location-based services. Some research works [46, 47, 48, 49] consider the re-occurrence of GPS readings around the same location and thus propose to use clustering algorithms to group GPS points to get locations. On the other hand, in [43, 44, 45] each GPS
point in the trajectory data is treated as a location. Thus, these works have no need to cluster
the GPS points into locations. In addition to location extraction, Zhou et. al. [47] propose to
mine personal important locations by labeling locations with “home” or “work office” tags;
Zheng et. al. [48] propose to mine interesting locations and travel sequences from GPS data,
and applied a HITS-based inference model to rank locations; and Cao et. al. [49] consider
semantics to enhance the location extraction process and also propose new models for ranking
locations, which are capable of better exploiting the features of GPS trajectories, such as the
relationships between locations, the distances between locations, and the durations of visits.

2.1.2 Transportation Model Inference

The transportation mode, such as walking, driving, etc., implied in a user’s GPS trajectory
data can provide us valuable knowledge to understand the user. In [50, 51], Patterson et.
al., propose a Bayesian model based particle filters to infer the transportation mode based
on raw GPS readings. In [52], for transportation model inference, Zheng et. al. propose an
approach consisting of three parts: a change point-based segmentation method, an inference
model and a post-processing algorithm based on conditional probability. The change point-
based segmentation method is compared with two baselines including uniform duration based
and uniform length based methods. Meanwhile, four different inference models including
Decision Tree, Bayesian Network, Support Vector Machine (SVM) and Conditional Random
Field (CRF) are studied in the experiments.

2.1.3 Human Mobility Model

The spatial characteristics of user movement has been studied for years, even before the emer-
gence of online social networking systems. In [31], a dataset collected from circulation of
bank notes in United State is employed to study human traveling statistics. The analytical
result shows that the distribution of traveling distances follows a power law, indicating that
trajectories of bank notes are scale free random walks known as Levy flights. In [32], Gonzale-
lez et. al. propose to investigate the human mobility patterns by studying the trajectories of
100,000 anonymized mobile phone users whose positions are tracked over a six-month period.
In contrast with the random trajectories predicted by the prevailing Levy flight [31], human tra-
jectories show a high degree of temporal and spatial regularity. In other words, each individual
has a significant probability to return to a few highly frequented locations, which has also been
confirmed by [33].

2.1.4 Location/Movement Prediction

Location or movement prediction is an important part of many ubiquitous computing systems. In [50, 53, 51], Patterson et al. propose to apply particle filter techniques to predict users’ positions based on their previous GPS movement traces. In [54], Akoush et al. present a hybrid Bayesian neural network model for predicting locations on cellular networks, and compare many standard neural network techniques such as back-propagation, Elman, Resilient, Levenberg-Marqudat, and One-Step Secant models for the implementation. In [46], Ashbrook et al. propose to use Markov model to investigate the relationship among different locations, by learning the transition probability from one location to another location. The transition probability among locations can be employed for the movement prediction. In [55], Yavas et al. propose a frequent pattern mining based algorithm for predicting the next inter-cell movement of a mobile user in a cellular network. In the algorithm, mobility patterns are mined from the mobile user trajectories, then mobility rules are extracted from those patterns, and finally movement prediction is accomplished by using those mobility rules. Similarly, in [56], the frequent movement patterns are extracted from history data, namely Trajectory Patterns. A decision tree, named T-pattern Tree, is built to learn the Trajectory Patterns in a certain area, which can be utilized for movement predication. In [57], Eagle et al. identify the structure inherent in daily behaviors by the technique of principal component analysis (PCA), i.e., an individual’s behavior over a specific day can be approximated by a weighted sum of his or her primary characteristic vectors, namely “eigenbehaviors”. When these weights are calculated halfway through a day, they can be used to predict the day’s remaining behaviors. In [58], Song et al. investigate the predictability of human movement by studying the mobility patterns of anonymized mobile phone users. By measuring the entropy of each individual’s trajectory, they find a 93% potential predictability in user mobility across the whole user base, despite the significant differences in the travel patterns for individuals.

2.1.5 Urban Planning

In addition to GPS traces collected by mobile users, GPS trajectories collected from taxis have also attracted a lot of attentions these years. In [59, 60], Yuan et al. present techniques that compute customized and practically fast driving routes for an end user, by incorporating
the knowledge learned from (historical and real-time) traffic conditions and driver behavior. Furthermore, they learn the knowledge of passengers’ mobility patterns and taxi drivers pick-up behaviors from the GPS trajectories of taxicabs, to recommend taxi drivers where to pick up passengers quickly, and recommend people where to find vacant taxis [61]. Besides the location-based services for individuals, studies on GPS trajectories are beneficial for urban planning as well. Liu et. al. propose a non-density-based approach called mobility-based clustering to identify hot spots of moving vehicles in an urban area [62] and further develop a visual analytics system for metropolitan transportation [63]. Zheng et. al. propose to use the GPS trajectories of taxicabs traveling in urban areas to examine the current urban planning, e.g., to detect pairs of regions with salient traffic problems and find out linking structure as well as correlation among different regions [64]. Finally, besides of GPS traces from taxicabs, GPS trajectories from buses have been studied as well. For example, in [65], the regularity of scheduled vehicle traveling time has been explored to enable the services about predicting bus arrival time at some specific stop/station.

2.2 Geo-associated Text Data

In this section, we introduce research works on location entity extraction, followed by the discussion of the related works about spatial information retrieval.

2.2.1 Location Entity Extraction

Location entity extraction from text documents, as one of the applications for named entity extraction [66, 67, 68, 69], has attracted more and more attentions in recent years. The specific task of determining which place is meant by a particular occurrence of a place name, known as grounding (also referred to as localization), has been gaining attention. Clearly, grounding requires general-world knowledge and cannot rely completely on information found in the text or even in the whole corpus. This general knowledge is provided in a gazetteer, which traditionally lists the names of all places in an atlas. Most of work do not employ machine learning, but are rather based on various NLP heuristics. The reason is that machine learning algorithms are more expensive and require training data that is not readily available. Thus most of location extraction work are based on the knowledge provided by gazetteer [70, 71, 72, 73, 74, 75]. However, even if the help of gazetteer is available, it is still challenging to identify
grounding of a place name due to two types of ambiguities: (1) geo/non-geo ambiguity - term which is a place name in a given text, is commonly used as non-geo sense term, e.g., “Of” (Turkey); (2) geo/geo ambiguity - different geographical places may share the same place name, such as Paris in France and Paris in Texas, US.

2.2.2 Spatial Information Retrieval

With advances in location extraction techniques [74, 76], spatial interests expressed in text can be learned to support web services for users looking for spatial information [77, 74, 78, 79]. In [80], Wang et. al. consider three different types of evidence to determine the geographic location(s) associated with a document, including provider location, content location, and serving location. In [78], Wang et. al. propose a notion of query dominant location (QDL) to detect the locations truly searched for by queries. On the contrary, Ding et. al. propose to study the geographical extent of spatial web resources that their creators intend to reach [77]. Particularly, Ding et. al. introduce techniques based on the textual content of the resources, as well as on the geographical distribution of hyperlinks to automatically compute the geographical scope of web resources. Amitay et. al. propose to assign to each page a (or a few) “geographic focus” in [74]. They identify the focus of a given page by assigning different confidence score for different locations and use hierarchical structure of location terms to summarize locations with high confidence. Silva et. al. [81] propose to use a graph-based approach akin to PageRank to determine a single scopes for each document, based on a geographic knowledge repository. In [79], Aencar et. al. describe strategies for document classification into geography-related categories, using evidence extracted from Wikipedia. Recently, Hao et. al. aiming to provide travelogue services based on summarization of a travelogue corpus, propose to mine location-specific knowledge from a large collection of travelogues using a probabilistic generative model [82]. In our work [83], through theme location detection, we enable the provisioning of travelogue digests in our travelogue system, and provide new views for users to understand a travelogue.

2.3 Data from Emerging LBSNs

Recently, a growing number of location-based social networking services (LBSNs), e.g., Facebook Place, Twitter, Foursquare, Flickr, Yelp, etc, have also emerged. They allow users to connect with friends, explore places (e.g., restaurants, stores, theaters, etc), share their loca-
tions, and upload photos, videos, and blogs. The data we are talking about in this section include such as tweets from twitter, place data from Foursquare, photo data from Flickr, and so on.

2.3.1 Spatio-Social Analysis

Recently, the emergence of location-based social networking sites like Foursquare, Gowalla, and Facebook Places, provides a unique opportunity to study the social and temporal characteristics of how people use these services and to model patterns of human mobility. In [34], a simple socio-spatial property has been observed in Facebook dataset, i.e., the probability of friendship of two people is roughly inversely proportional to distance between home locations of those two. Based on this observation, a simple yet effective user’s home location prediction algorithm was proposed. In [35, 36], in addition of the similar observation regarding the social friendship and geographical distance as in [34], Scellato et. al. observe heterogeneous characteristics in the distance of interaction across users, with some of them exhibiting preference towards short-range rather than long-distance social friendship. Particularly, users with more friends tend to create triangles with individuals further apart far more than expected by chance. In [37], McGee et. al. propose to investigate the relationship between geographical distance and tie strength between two users by analyzing data collected from Twitter. Activities such as following, mentioning, and actively engaging in conversations with another user are strong signals which help to determine the geographical distance between a pair of users. In [38], Cheng et. al. analyze human mobility patterns by collecting data from Foursquare and Gowalla. They find that users follow the Levy Flight [31] mobility pattern and also exhibit regular behaviors [32]. Furthermore, they find that in addition of geographic and economic constraints, social network structure also has impact on the mobility patterns. In [39], Cho et. al. find that users movement in location-based social networks is a combination of periodic movement and random jump, where periodic movement is due to the constraint of geographical proximity and random jump is correlated with the social activities. More specifically, short-ranged travel exhibits both spatial and temporal regularity and is not effected by social networks, while long-distance travel is more affected by social ties.

As shown in the above studies, there is strong correlation between user’s spatial activities and their social relationships. So according to the observation of users’ spatial activities, it is possible to infer the social tie among users [40, 41]. More specifically, in [40, 41], they propose
to use data collected from mobile phones users to conduct the research. [40] show that there
is distinctive temporal and spatial patterns in physical proximity and calling patterns for those
people who are closely social connected. Thus the temporal and spatial patterns are strong
signals for social tie inference. In [41], Wang et. al. find that human mobility pattern could
indeed serve as a good predictor for the formation of new social ties, and the combination of
both mobility and network measures can significantly improve the social tie inference accuracy.
Finally, Crandall et. al. [42] conduct similar research by using a Flickr dataset. They investigate
the extent to which social ties between people can be inferred from co-occurrence in time and
space.

2.3.2 User’s Location Detection

In [84], Hecht et. al. propose to study the behavior of users with respect to the location-field in
their Twitter profiles. They find that 34% of users do not provide real location information, e.g.,
frequently incorporate fake locations. Nevertheless, they also find user tweets can help to infer
the user’s location with decent accuracy. In [85], Cheng et. al. use similar ideas to model the
spatial distribution of words in tweets to predict the user’s location. With the information about
user’s location, we could provide better personalized location-based services for individuals.

2.3.3 Location Type Classification

Meaningful names of locations/places are very important contexts for location-based services.
In [86], Liao et. al. propose a relational Markov network to label locations with the activities
that have occurred in these places. In [87], they hierarchically construct Conditional Random
Fields to extract and label places simultaneously, where only four types of names (work, home,
friend, parking) are included in the labeling sets. In [88], Lian et. al. treat the location naming
problem as the location-based search problem. They propose a local search framework to inte-
grate different kinds of popularity factors and personal preferences, and apply learning-to-rank
technique to realize location naming services. In [89], Lin et. al. model the naming prefer-
ence when sharing locations with others and study the factors that influence location naming
preference, including a place’s entropy, social group, recipient’s familiarity to sharing places,
and sharer’s comfortable level of sharing locations. In [90], Cheng et. al. study location-based
traffic patterns revealed through location sharing services and find that these traffic patterns
can identify semantically related locations and help predict the semantic category of uncate-
gorized locations. Our work in [91] aims to address the problem of semantic annotation for places in location-based social networks such as Foursquare and Whrrl, by treating it as a multi-label classification problem. We investigate the check-in patterns corresponding to each individual place and behavioral regularity of each individual user to derive features for classification. Besides, our study in [92], which investigates the temporal dimension of feature types in location-based social network, could also help regarding location name determination.

2.3.4 Location/Place Recommendation

Besides of research on location naming (or semantic type tagging), people also develop research such as recommendation system to help users explore new places/locations [93, 26, 94, 95, 96, 97]. In [94, 95], Zheng et. al., propose to provide two types of location recommendation services for individual users. The first is a generic one that recommends a user with the most popular locations and travel sequences in a given spatial region. The second is a personalized recommendation that provides an individual with locations matching her own travel preferences through the approach of collaborative filtering. In [93, 26], we propose to add social influence factors into recommendation algorithms as users may turn to friends for opinions sometimes. Particularly, in [26], we further investigate geographical influence, in addition of social influence, to improve performance of place recommendation. In [96], Chow et. al., introduce GeoSocialDB [98] – a system that support scalable location-based social networking services, such as location-based news feed and ranking [97] and location-based place recommendation [99].

2.3.5 Spatial Information Retrieval

In addition to tweets and place data, there is research that provides location-based service upon photo data collected from Flickr [100, 101, 102, 103]. In [101, 102], Rattenbury et. al. develop methods for extracting place semantics according to the tags that are assigned to photos taken around that place. In [100], Jaffe et. al. propose a framework for automatically selecting a summary set of photos from a large collection of geo-referenced photographs. The summary algorithm is based on spatial patterns in photo sets, as well as textual-topical patterns and user (photographer) identity cues. The algorithm can be expanded to support social, temporal, and other factors. In [104], Ahern et. al. investigate how to analyze the tags associated with the geo-referenced Flickr photos to generate aggregated knowledge in the form of "representative
tags” for arbitrary areas in the world. In [105], Crandall et. al. investigate how to organize a large collection of geo-tagged photos. Their approach combines content analysis based on text tags and image data with structural analysis based on geo-spatial data. In [106], Sizov et. al. propose a multi-modal Bayesian model which combines text features (e.g. tags as a prominent example of short, unstructured text labels) with spatial knowledge (e.g. geotags and coordinates of images and videos), aiming to construct better algorithms for content management, retrieval, and sharing. In [107], Yin et. al. propose to discover and compare geographical topics from GPS-associated document such as Flickr photos. They develop a latent geographical topic analysis approach (LGTA) that combines location and text, where this approach not only finds regions of interests but also provides effective comparisons of the topics across different locations.

2.3.6 Privacy Issues

In [108], Vicente et. al. propose to study the location-related privacy in location-based social networks, which provide services such as photo sharing, friend tracking and “check-ins”. They point out that new privacy issues related to social networks may be caused by exposing users’ locations. They discuss the new privacy issues including location privacy, absence privacy, co-location privacy and identity privacy. And they further describe possible means of protecting privacy in those circumstances. In [109], Freni et. al., propose two solutions based on spatial and temporal cloaking to preserve location and absence privacy in location-based social networks that support user tagging. Their solutions use reasoning on the distance and time among resources, user tags and geo-tags to detect resources that can cause privacy violations.

In [110, 111], Siksnys et. al. propose to provide a privacy-aware proximity detection service to determine if two mobile users are close to each other without requiring them to disclose their exact locations. Existing proposals for such services provide weak privacy, give low accuracy guarantees, incur high communication costs, or lack flexibility in user preferences. They address those shortcomings with a client-server solution for proximity detection, based on encrypted, multi-level partitions of the spatial domain.

In [112], Khoshgozaran et. al. identify and address the key challenges of enabling private spatial queries in social networks using an untrusted server model without compromising users’ privacy. They propose Private Buddy Search (PBS), a framework to enable private evaluation of spatial queries predominantly used in social networks, without compromising sensitive in-
formation about its users. Utilizing server side encrypted index structures and client side query processing, PBS achieves both scalability and privacy.

In [113], Motahari et. al. discuss privacy risk results from social inferences about user identity, location and other personal information. After analyzing the social inference problem theoretically, they assess the extent of the risk to users of computer-mediated communication and location based applications through 1) a laboratory experimentation, 2) a mobile phone field study, and 3) simulation.
On Theme Location Discovery for Travelogue Services

In this chapter, we aim to develop a travelogue service that discovers and conveys various travelogue digests, in form of theme locations, geographical scope, traveling trajectory and location snippet, to users. In this service, theme locations in a travelogue are the core information to discover. Thus we aim to address the problem of theme location discovery to enable the above travelogue services. Due to the inherent ambiguity of location relevance, we perform location relevance mining (LRM) in two complementary angles, relevance classification and relevance ranking, to provide comprehensive understanding of locations. Furthermore, we explore the textual (e.g., surrounding words) and geographical (e.g., geographical relationship among locations) features of locations to develop a co-training model for enhancement of classification performance. Built upon the mining result of LRM, we develop a series of techniques for provisioning of the aforementioned travelogue digests in our travelogue system. Finally, we conduct comprehensive experiments on collected travelogues to evaluate the performance of our location relevance mining techniques and demonstrate the effectiveness of the travelogue service.

3.1 Overview

Continued flourish of World Wide Web (WWW) and advance of transportation systems have made travel, an integral part of our life, very easy. Particularly, with the advantages of Web
2.0 technology, many people are willing to share their *travelogues*, which record travel experiences, on weblogs, forums and social communities for travels, (e.g., TravelPod\(^1\), IgoUgo\(^2\) and TravelBlog\(^3\)).

Travelogues, which usually contain rich travel information, such as the tours, lodging, meals, expenses, weather conditions, and so on, are highly valuable to a trip planner. Thus, many people turn to on-line travelogue archives or discussion forums to review travelogues well before taking their trips. While reading travelogues may be fun and enjoyable for some, digging into numerous travelogues to find useful trip information is a tedious and boring task for many. Thus, there is a need for developing automatic travelogue mining techniques to convey useful information in a travelogue to its readers in a more effective way.

Generally, there are two kinds of information for a travelogue to be mined; (1) “where-about” of the travelogue, and (2) “how-about” of the mentioned attractions.

As a travelogue intends to record activities and experiences of its author at locations on a trip (i.e., the whereabout of the trip), a key problem related to (1) concerns extraction of *theme locations*, i.e., locations appeared in a travelogue which are closely relevant to the main themes of the travelogue. Location extraction techniques have been developed in the past \cite{74, 76}. However, due to the nature of human language presentation, locations extracted are not necessarily theme locations of the travelogue. For example, in Figure 1, Siesta Key is very relevant to the theme of this paragraph but Boston is not.

| Siesta Key | - It is a paradise found in Sarasota, FL, located a little over an hour drive south to Tampa, FL, and is a favorite gateway spot for many Florida locals. I like to spend time there - mainly at beaches and bars ... We got back to Boston after one week vacation. |

*Figure 3.1.* A paragraph in a travelogue for Siesta Key. Relevant locations are in **bold**, partial relevant locations are in *teletype*, and irrelevant locations are in *italic*.

Another important problem, related to (2) is about extracting specific how-about information regarding theme locations. Hao et. al. \cite{82} pioneered a knowledge discovery technique that extracts general information (e.g., food, surfing, shopping) regarding travel destinations from a corpus of travelogues. However, this technique has mainly focussed on location in-

---

\(^1\)http://www.travelpod.com  
\(^2\)http://igougo.com  
\(^3\)http://www.travelblog.com
formation extraction without distinguishing whether locations in a travelogue is relevant to its themes or not. As a result, the result is inevitably biased by noisy locations.

To tackle the theme location discovery problem, we propose to extract *textual features* of locations (e.g., surrounding words and syntactic pattern) as well as *geographical features* of locations (e.g., individual location types, structural constraints derived from location ontology) for location relevance mining. With those features, we perform location relevance classification to differentiate relevant theme locations from others in a travelogue. Moreover, we propose a co-training model, by taking advantages of the independent textual and geographical views of the extracted features, to boost classification performance.

Unfortunately, human perception of locations relevant to the themes of a travelogue tends to be fuzzy! Different people may employ different personal criteria to assess relevant theme locations. What’s more, the content of a travelogue may sometimes be insufficient for one to decide whether a location is a theme location or not. However, based on the context of a travelogue, we often can still tell whether a location $l_A$ is more relevant than another location $l_B$ to the travelogue, without necessarily deciding whether $l_A$ and $l_B$ are considered as theme locations or not. Based on this observation, we also develop algorithms for location relevance ranking.

Relevance classification and ranking provide two alternative and complementary aspects in our study of location relevance mining. To our best knowledge, there is no existing work addressing the issues of location relevance mining for a given travelogue, not to mention its support for provisioning of various information digests in a travelogue service. Built upon the mining result of location relevance mining, we develop a series of techniques, including theme locations detection, geographical scope calculation, travel trajectory extraction and location snippet extraction, to embody a travelogue service which not only enable trip planners to find where to go but also what to do. Finally, we conduct a comprehensive set of experiments on collected travelogues to validate our ideas, evaluate the performance of our location relevance mining techniques, and demonstrate the effectiveness of the travelogue service.

In the following, we introduce the most related literatures, and then provide a overview about location relevance mining for theme location discovery, including feature extraction and detail algorithms. Furthermore, we introduce the techniques to realize travelogue services built upon LRM, and conduct performance evaluation to validate the proposed location relevance mining techniques and demonstrate the travelogue service. Finally, we summarize the chapter.
3.2 Related Work

In this section, we first introduce some existing studies on spatial information retrieval, such as news and travelogues. Then we review techniques related to location relevance mining, followed by a brief summary of semi-supervised learning techniques.

Location-based search has received a lot of interests from both the academia and industry in the past few years. With advances in location extraction techniques [74, 76], spatial interests expressed in text can be learned to support web services for users looking for spatial information [77, 74, 78, 79]. Wang et. al. propose a notion of query dominant location (QDL) to detect the locations truly searched for by queries [78]. On the contrary, Ding et. al., propose to study the geographical extent of spatial web resources that their creators intend to reach [77]. Amitay et. al. propose to assign to each page a (or a few) “geographic focus” in [74]. They identify the focus of a given page by assigning different confidence score for different locations and use hierarchical structure of location terms to summarize locations with high confidence. Silva et. al. [81] propose to use a graph-based approach akin to PageRank to determine a single scopes for each document, based on a geographic knowledge repository. In [79], Aencar et. al. describe strategies for document classification into geography-related categories, using evidence extracted from Wikipedia. Recently, Hao et. al., aiming to provide travelogue services based on summarization of a travelogue corpus, propose to mine location-specific knowledge from a large collection of travelogues using a probabilistic generative model [82]. Different from most existing work, our study enables provisioning of travelogue digests in our travelogue system, and provide new views for users to understand a travelogue.

Here, we explore location relevance classification and ranking techniques to address the issues of location relevance mining. While there exist many textual classification techniques such as Support Vector Machine (SVM) [114] and logistic regression [114] and document ranking methods [115, 116, 117, 118] in the literature, the problem of location relevance mining discussed in this paper is very much new, which requires a unique treatment of certain geographical characteristics. For example, Siesta Key is a sibling city of Tampa, while Florida is the state where Siesta Key and Tampa reside. These unique geographical characteristics motivate us to explore interesting location features in devising techniques for location relevance classification and ranking.

In this chapter, we follow the semi-supervised learning paradigm to address the issues of location relevance mining. Semi-supervised learning is desirable for our study because we
have obtained a small set of labeled data and a large volume of unlabeled data. The key idea of semi-supervised learning is to enrich labeled training data by labeling unlabeled data via certain learning techniques. Basically, there are three techniques to realize semi-supervised learning, including (1) using the EM (expectation-maximization) algorithm to estimate the parameters of a generative model and the labels of unlabel data [119], (2) defining a graph over the data instances on the basis of certain similarity metric and determining the labels of unlabeled data [120], and (3) using co-training algorithms to exploit the strengths of multiple learners to label the unlabeled data [121, 122]. In this work, we realize the semi-supervised learning by exploring two independent views, i.e., textual and geographical views, in a co-training approach.

3.3 Theme Location Discovery

Theme location discovery aims to discover and assess the theme locations in a given travelogue. The problem of theme location discovery raises two different yet complementary questions: (1) what locations in a travelogue are theme locations? (2) given two locations in a travelogue, which one is more likely to be a theme location?

These two questions are difficult to answer because we don’t know exactly what is the main theme of a travelogue perceived by author and readers. However, based on the intention of travelogue writing, we argue that there indeed exist one or more themes in a travelogue, which match well with the purposes/activities/destinations of this trip. To identify the theme locations, one naive approach is to mandate the travelogue authors to supply them or have the readers to label them, but this manual approach is hard to enforce. Moreover, there are already a huge volume of travelogues made available on-line. Therefore, a mining technique that extracts the theme locations for a given travelogue automatically is highly desirable. Based on observations obtained in our preliminary study of labeled travelogues, the theme locations perceived by different people are similar to some extent, even though they may not be the same. Thus, by leveraging the human knowledge, we apply a supervised learning approach to automatically discover theme locations from travelogues using a labeled training dataset. Moreover, as mentioned earlier, we adopt a semi-supervised learning approach to enrich the limited labeled dataset in order to boost the learning performance. To answer the two questions raised above, we study the issues of location relevance for finding the theme locations in two complementary aspects: location relevance classification and location relevance ranking.
**Location relevance classification.** The first question can be considered as a classical classification problem by extracting features for each candidate location in the travelogue. However, as we mentioned before, there is an inherent ambiguity among different people in terms of how relevant a candidate location is to the theme of the travelogue. What’s more, the content of travelogue is sometimes insufficient to assess whether a location is a theme location or not. For example, in Figure 3.1, one may assess that Siesta Key is the only theme location; while another person may consider both Sarasota and Florida are also theme locations. Thus, as shown in Figure 3.2, the main challenge for location relevance classification is where to set a boundary separating theme location from irrelevant locations.

![Figure 3.2. Dilemma in relevant location classification](image)

**Location relevance ranking.** Instead of determining whether a location is a theme location, the second question we aim to answer is whether one location is more relevant than another location to the theme of this trip. As shown in Figure 3.2, assuming that Siesta Key is the sole theme, we may assess that Tampa is less relevant to the theme location than Sarasota. In this paper, we adopt two alternative approaches, namely *location likelihood estimation* and *learning to rank*, for location relevance ranking. As discussed in Section 5.2, while document classification and ranking techniques have been explored for document retrieval, there is an inherent difference between travelogues and documents because locations in a travelogue usually have some interesting relationship, e.g., Siesta Key is a sibling city of Tampa, while Florida is the state which covers Siesta Key and Tampa. Thus, conventional document mining techniques cannot be applied directly to our problem. These unique spatial characteristics motivates us to explore interesting features for location relevance mining.
3.4 Feature Extraction

To perform location relevance mining, it is essential to identify useful features of a location that can help to assess whether the location is relevant to main themes of the travelogue or not. Since travelogues are textual documents recording the authors’ experiences in touring places of interests, it naturally contains textual and geographical features. Thus, we identify features associate with locations in these two aspects. Table 3.1 summarizes the features extracted in our study.

<table>
<thead>
<tr>
<th>Category</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual</td>
<td>(F_1): is in title?</td>
</tr>
<tr>
<td></td>
<td>(F_2): number of appearance</td>
</tr>
<tr>
<td></td>
<td>(F_3): bag-of-words</td>
</tr>
<tr>
<td></td>
<td>(F_4): syntactic pattern</td>
</tr>
<tr>
<td>Geographical</td>
<td>(F_5): location type</td>
</tr>
<tr>
<td></td>
<td>(F_6): partonomy distance</td>
</tr>
<tr>
<td></td>
<td>(F_7): geographical proximity</td>
</tr>
</tbody>
</table>

Table 3.1. Features and their categories

On the one hand, for a travelogue, we usually have a title to summarize the theme of the travel. The most relevant locations are usually included there. What’s more, interested locations might be mentioned several times in a travelogue, while irrelevant locations appear less frequently. According to this observation, we extract two kinds of features, namely, *is in title?* (denoted as \(F_1\)) and *number of appearance* (denoted as \(F_2\)), respectively. Besides, surrounding words of a location also provides useful hints. For example, in Figure 3.1, one may easily understand that “Siesta Key” is the most relevant location name in this paragraph, because important locations are usually heading the paragraphs. Thus for a location in a travelogue, we use its surrounding words to extract *bag-of-word* feature (denoted as \(F_3\)) and *syntactic pattern* feature (denoted as \(F_4\)) to describe the location. Those above four features are categorized as textual feature.

On the other hand, each location as a physical entity holds geographical properties in the real word. For example, location names referred in a travelogue usually have different location types in location partonomy. As shown in Figure 3.3, among the locations mentioned in the travelogue in Figure 3.1, some of them are state name (e.g., Florida), while some are county or township names (e.g., Sarasota and Siesta Key). Location names with smaller scope usually hold higher relevance, otherwise, travelers would not bother to mention them in the travelogue.
Thus, we consider the location type to be an important feature, named as location type (denoted as $F_5$).

Additionally, relevant locations are usually clustered in the partonomy hierarchy of a location ontology. Moreover, people would likely stay in the same state or the same city during the trip. In other words, partonomy distance among the locations provides important information to leverage relevant locations. More specifically, we extract the partonomy distance among locations as feature (denoted as $F_6$) as follows. Let the partonomy distance ($d_{\text{par}}$) between two locations be the total number of hops in the partonomy hierarchy to their immediate common ancestor. Let $l_1$ and $l_2$ be two different locations mentioned in a given travelogue, and $l_a$ be the immediate common ancestor of $l_1$ and $l_2$.

$$d_{\text{par}}(l_1, l_2) = \text{hops}(l_1, l_a) + \text{hops}(l_2, l_a).$$

(3.1)

where $\text{hops}(l_x, l_y)$ denotes the number of hops between $l_x$ and $l_y$ along the path of partonomy hierarchy (it equals zero if $l_x = l_y$). For example, in Figure 3.3, we have the partonomy distance between “Siesta Key” and “Boston” computed as 5, where the immediate common ancestor is “United State”.

Finally, if two locations are far away, e.g., Siesta Key and Boston in Figure 3.3, only one of them could likely be the relevant location for a travelogue. Therefore, we consider geographical proximity ($d_{\text{geo}}$) between two different locations (mentioned in the same travelogue) to be an important feature (denoted as $F_7$). More specifically, let $l_1$ and $l_2$ be two different locations mentioned in a given travelogue, and their latitude and longitude are denoted as $(x_1, y_1)$ and $(x_2, y_2)$ respectively. For simplicity, we use Euclidian distance to present the geographical

---

**Figure 3.3.** Partonomy hierarchy of locations mentioned in the travelogue shown in Figure 3.1
distance between $l_1$ and $l_2$ as follows.

$$d_{geo}(l_1, l_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$ (3.2)

### 3.5 Mining Algorithms

In this section, we introduce algorithms to realize the two mining tasks, namely, location relevance classification and location relevance ranking, for theme location discovery.

#### 3.5.1 Location Relevance Classification

Location relevance classification aims to predict whether a location is relevant to the theme of the travelogue or not. As aforementioned, both textual and geographical features are able to assess whether a location is relevant to the theme of a travelogue or not, so we exploit both feature sets of a location to realize location relevance classification. Because of its popularity and good performance in text mining, we adopt SVM to perform supervised classification. Nevertheless, based on our preliminary study, directly applying SVM to perform supervised classification does not produce a good performance. On the one hand, location relevance classification is very challenging due to the inherent ambiguity of location relevance and the limited

![Co-training framework for location relevance classification](image-url)

**Figure 3.4.** Co-training framework for location relevance classification
amount of labeled data available. On the other hand, we noticed that travelogues naturally contains two independent feature sets, i.e., textual and geographical features. Aiming to boost the performance of location relevance classification, we adopt the co-training framework (i.e., semi-supervised learning) as shown in Figure 3.4 to explore the two feature sets in order to enrich the training set with high-quality newly labeled data in order to obtain a better SVM classifier model. As one of the popular semi-supervised learning algorithms, co-training is a perfectly fit for the classification of location relevance here. It has been shown in [119] that co-training works well under two conditions: (1) each set of features is sufficient for classification, and (2) the two feature sets of each instance are conditionally independent. As both textual and geographical features provide strong hints about whether a location is relevant to the theme location of the travelogue. Based on our preliminary experimentation, we found that the SVM classifiers modeled based on either textual features or geographical features are capable of doing location relevance classification. Moreover, textual and geographical features are proposed to capture the characteristics of a location in totally independent angles, and thus complementary with each other. With both of the above conditions hold, we adopt co-training to boost the performance for location relevance classification.

Figure 3.4 illustrates the proposed framework for overall location relevance classification. As shown, the location extraction component in our system scans the travelogue corpus to extract the locations together with their features for each travelogue. The resulted datasets, including candidate set and training set corresponding to unlabeled and labeled travelogues, are fed to the co-training module to produce an enriched training set (including the original training set plus those high quality newly labeled data) for further classification in the final SVM module (i.e., the bottom box of the figure). Notice that, the final SVM module performs classification in the entire feature space including both textual and geographical features.

Co-training is an iterative process, which typically runs on a large corpus of text documents (i.e., the candidate set) together with a seed training data set (i.e., labeled travelogues here). The main idea of our co-training framework is to explore both textual features and geographical features associated with locations in travelogues to independently and iteratively mine high-quality labels for travelogues. Our co-training algorithm starts with a small number of seed data in the training set. By treating the data instance space in two different views, namely, textual view and geographical view, we have $X = \mathbf{X}_t \times \mathbf{X}_g$, where $X$ denotes the entire feature space, while $\mathbf{X}_t$ and $\mathbf{X}_g$ present textual feature space and geographical feature space, respectively. Iterating between these two different views, we use two separate SVMs (one for
each view) to decide the labels for those unlabeled travelogues in the candidate set. With newly labeled travelogues, we validate their labels by assigning confidence scores, which are derived based on the distance of a given travelogue to the classification hyperplane learned by SVM. More specifically, the confidence score $s(x)$ for a newly labeled travelogue $x \in X_t$ (or $X_g$) is calculated as follows.

$$s(x) = |f(x)| = |w^* x + b| \quad (3.3)$$

where $w^*$ is the optimized hyperplane, while $b$ is the bias term. In order to enrich the training set with high-quality labeled travelogues, we derive a good confidence score threshold $\theta$ based on a validation dataset and the current training model learned in each round. As such, only high-quality data (i.e., those newly labeled travelogues with high confidence scores ($s(x) \geq \theta$)) are included the training set. As the co-training runs, the training set is enriched continuously until it converges eventually. Finally, a new training dataset larger than the original one is derived from the co-training process. We then adopt SVM to learn upon this new training dataset in the complete feature space (including both textual and geographical features) to obtain the final classifier model.

### 3.5.2 Location Relevance Ranking

Location relevance ranking aims to sort locations in accordance with their relevances to the theme of a travelogue. It can be analogized to the problem of document relevance ranking. In location relevance ranking problem, we treat the themes of a travelogue as the query keywords, locations included in the travelogue as candidate documents. However, different from traditional document retrieval, the query keywords (i.e., themes) are implicitly contained in travelogues instead of explicitly specified by users. Here, we introduce two alternative approaches, namely, location likelihood estimation and learning to rank algorithms to address the ranking problem.

#### 3.5.2.1 Location Likelihood Estimation

A travelogue is written to record the trip experiences of the author over visited theme locations. Thus, an idea to estimate and rank the relevance of a location in a travelogue to the main themes of the travelogue, motivated by language modeling in information retrieval, is to consider the generation probability of a location in the travelogue. Let $l$ and $t$ denote a location and a travelogue, we use $p(l|t)$ to denote the probability of $l$ appearing in the observed travelogue
t. Intuitively, the larger the generation probability of l in t holds, the more relevant l is to the travelogue t. Thus, we define the relevance of location l to a travelogue t as below.

\[ r(l, t) = p(l|t) = \lambda p(l|M_t) + (1 - \lambda) p(l|M_c) = \lambda \frac{tf_l}{N_t} + (1 - \lambda) \frac{tf_l}{N} \] (3.4)

where \(\lambda\) is a smoothing factor, and \(M_t\) and \(M_c\) are the location vocabularies built upon travelogue t and the whole corpus respectively. \(tf_l\) are term frequency of location term l appearing in travelogue t and the whole travelogue corpus respectively, and \(N_t\) and \(N\) are the number of location terms in travelogue t and the whole travelogue corpus, respectively.

3.5.2.2 Learning to Rank

Recall our example in Figure 3.1. There are five locations included in the travelogue, but each of them appears only once in the whole document. Based on location likelihood estimate, we may assign them the same rankings, given that smoothing is not applied. Since Siesta Key, Florida, Tampa, and Boston are all popular places, the smoothing technique is not expected to help here. The question is why the location likelihood estimation approach would suggest all those locations to have the same ranks, which is actually against our intuition. The reason is that the language model mainly considers the frequencies of locations, without taking account of other valuable features of given documents, e.g., the surrounding words of locations and the correlation among locations in a travelogue. Thus, we propose an alternative approach, learning to rank model, to realize location relevance ranking by exploiting the textual and geographical feature of the travelogue.

Due to its popularity and simplicity, we adopt Ranking SVM [117], which takes pair-wise relationship of locations in a travelogue to learn a ranking model. Given an input feature space \(X\), where \(X\) includes both textual and geographical features, there exists an output space of ranks represented by labels \(Y = \{r_1, \cdots, r_q\}\), where \(q\) denotes the possible number of ranks. Further, assume that there exists a total order amongst the ranks \(r_q > r_{q-1} > \cdots > r_1\), where \(>\) denotes a preference relationship. Notice that, in this paper, we have three ranks of location relevance, namely “relevant”, “partial relevant” and “irrelevant”, and the preference relationship in location relevance ranking is “relevant > partial relevant > irrelevant”.

Given a set of ranked instances \(Z = \{(x_1, y_1), \cdots, (x_n, y_n)\}\), where \(n\) is the number of training instances, \((x_j, y_j)(1 \leq j \leq n)\) is from the space of \(X \times Y\). If \(y_i > y_j\), we have \(x_i\) ranked ahead of \(x_j\), denoted as \(x_i > x_j\). Assume that \(F\) is the set of ranking functions, such that each function
$f \in F$ can rank location instances as:

$$x_i > x_j \iff f(x_i) > f(x_j)$$ (3.5)

In Ranking SVM, $f$ is assumed to be a linear function [117],

$$f(x) = \langle \omega, x \rangle$$ (3.6)

where $\omega$ is weight vector and $\langle \cdot, \cdot \rangle$ denotes inner product. $f(x) = 0$ corresponds to a hyperplane in the feature space. Thus we have

$$x_i > x_j \iff \langle \omega, x_i - x_j \rangle > 0$$ (3.7)

Given an instance pair $x_i$ and $x_j$, we create a new instance $x_i - x_j$. If $x_i$ is ranked ahead of $x_j$, we assign a label $+1$, otherwise $-1$ to the newly generated instance. As such, we produce a new data set upon which we can build a binary classification model, i.e., the Ranking SVM model. With the Ranking SVM model, and the optimal setting of $\omega^*$, for each location instance $x$ we get its ranking score as $f(x) = \langle \omega^*, x \rangle$.

### 3.6 The Travelogue Service

In this section, we study how to realize the proposed travelogue service based on the mined location relevance. Particularly, we are interested in conveying theme locations, geographical scope, traveling trajectory and location snippets in a travelogue to the users. Figure 3.5 illustrates a mock up of our proposed travelogue service, which provides visualized information for a travelogue over a map. As shown, New York is the geographical scope of the trip. The traveler visited Central Park, Bronx Zoo and The Cloisters in sequence (i.e., three theme locations and a traveling trajectory). By moving the mouse cursor to one of the theme locations, associated location snippet pops out to provide some information. In the following, we present how we use the result of location relevance mining for provisioning of the digests in the travelogue service.

**Theme Locations.** Locations is an essential entity of travelogues. Anticipating that users would be interested in quickly identifying where the trip activities have happened, we display the theme locations of a travelogue. As we mentioned before, the theme locations can be
The Cloisters, a branch of the Metropolitan Museum of Art, is not just a museum filled with beautiful medieval art, it's a place to find serenity outside of New York...

Figure 3.5. Travelogue service

generated in two ways: i) automatically discovered from the travelogue; and ii) tagged by the travelogue author. Thus, our location relevance classification technique can be used to automatically identify the theme locations. For author-tagged theme locations, we develop a location tag recommendation service to assist the travelogue authors to tag theme locations when they upload their travelogues. Specifically, the location tag recommendation service, presented to the author as a list of ranked theme locations in accordance with their estimated relevances to the theme of the uploaded travelogue, is realized based on the location relevance ranking techniques presented earlier.

Geographical Scope. As a travelogue may cover several theme locations, it is desirable to present the geographical scope of trip activities to its readers. Previous works define geographical scope as a single (or a few) location names to cover all locations mentioned in a web page [74, 78]. For our travelogue service, we argue that not all locations are useful for deducing the geographical scope for a travelogue, especially when the noisy locations may produce misleading results. Consider the travelogue in Figure 3.1, we may argue that its geographical scope should be narrowed down to Florida or even Siesta Key, since both Tampa and Boston are de-
viating from the theme of this travelogue. Therefore, we follow the ideas in [74] to compute the geographical scope but put our focus on how to derive the input location set, since noisy locations hamper the accuracy of expected geographical scope.

To compute the geographic scope of a travelogue $t$, we consider only the theme locations $L'_t$ obtained from location relevance classification which effectively purges noisy locations from the entire location set $L_t$ in $t$.$^{4}$ Thus, the geographical scope computing algorithm takes $L'_t$ as input and operates upon the partonomy hierarchy of location ontology (see Figure 3.3 as an example). First, for each location $l$ in $L'_t$, we initiate an importance score $v_l$ of $l$ as the number of its appearance in the travelogue. Next, we propagate the influence of $l$ to its parent node $l'$ (and recursively to its ancestor nodes) by adding $v_l \times d$ to $v'_l$ where $0 < d < 1$ is a decay factor. Consider the example in Figure 3.3, Siesta Key would propagate its influence of importance score to its parent node Sarasota and grand-parent node Florida because the traveler visits Siesta Key and she also stays at Sarasota and Florida. As such, each location in $L'_t$ obtains an importance score that takes influence from its descendants into account. We derive the geographical scope as a location set by adding the theme locations in order of their importance but skipping those with one or more ancestors or decedents already included in the set.

**Traveling Trajectory.** Traveling trajectory (denoted as $T$) is extracted based on the result of location relevance classification. Given a travelogue, we obtain a set of theme locations, denoted as $L = \{l_1, l_2, \cdots\}$, from location relevance mining. Given two locations $l_x, l_y \in L$, it is possible that $l_x$ is part of $l_y$ in the partonomy hierarchy. In this case, we exclude $l_y$ from the trajectory. For example, in Figure 3.3, if Siesta Key and Sarasota are both classified as theme locations, we consider Siesta Key be the reason for the traveler to visit Sarasota. Thus, for trajectory extraction, we highlight the specific point-of-interests and investigate their travel sequence to form a trajectory, which is a subset of $L$. In other words, locations included in a trajectory are not part of each other. The algorithm for forming POI set is shown in Algorithm 1.

Upon the POI set, we construct a trajectory $T$ based on a simple assumption that the appearance sequence of those location in $T$ is supposed to match well with the travel sequence of the traveler’s real trajectory. Thus, we construct the trajectory by sorting locations in the POIs in accordance with the sequence of their appearances in the travelogue to form $T$. Of course,

$^{4}$An alternative of $L'_t$ is to use the highest ranked locations obtained from location relevance ranking with some predetermined rank threshold.
Algorithm 1 POI_SETFORMATION_ALGORITHM

// Input: theme location set $L$ for a given travelogue
$POI \leftarrow L$

for each $l_i \in T$ do
  if exists $l_j \in L$ such that $l_j$ is part of $l_i$ then
    $POI \leftarrow POI - \{l_i\}$
  end if
end for

return $POI$

there still is a room for improvement in forming the trajectory. We plan to further investigate this issue in the future work.

Location Snippet. Location snippet consists of a few (usually one or two) sentences extracted from a travelogue to describe the corresponding location. Given a location $l$ in a travelogue, a desired location snippet is one that includes relevant information about $l$, e.g., interesting description about some features of $l$. Based on topic generative model, we may find a word $w$ related to a location $l$. Specifically, we may derive a probability $p(w|l)$ that models how likely a word $w$ describes a location $l$ for all words in the vocabulary $V$, which contains all possible words in travelogues [82]. Based on $p(w|l)$, we construct a word set $W_l$ (i.e., a subset of $V$) which includes informative words regarding $l$, i.e., word $w$ with high $p(w|l)$ are included in $W_l$.

Next, for each theme location $l$, we determine the semantic relevance (SemRel) between $l$ and a candidate snippet $s$ by comparing the “word similarity” between $W_l$ and the set of words in $s$, denoted as $W_s$. Intuitively, we can simply use cosine similarity to calculate the distance between $W_l$ and $W_s$. However, cosine similarity does not capture the latent topic behind of words (e.g., “lunch” and “food” are treated as independent semantically). Thus, we propose to use latent topic distribution to measure the similarity between a location $l$ and a candidate snippet $s$ as follow.

$$\text{SemRel}(l, s) = \frac{\sum_{w_1 \in W_l} \sum_{w_2 \in W_s} \text{TermSim}(w_1, w_2)}{\log(1 + |W_s|)}$$ (3.8)

where

$$\text{TermSim}(w_1, w_2) = \exp\{-\tau D_{JS}(\delta_{w_1}||\delta_{w_2})\},$$ (3.9)

where $\delta_w$ denotes the topic distribution of the word [82], $\tau$ is a normalizing factor, and $D_{JS}(p||q)$ denotes the Jensen-Shannon(JS) divergence.

Here please note that, while we only consider the semantic relevance in our selection of
representative snippet for a theme location, the geographical relevance has been implicitly taken into consideration in the process of location relevance mining, which effectively purges noisy locations.

3.7 Performance Evaluation

In this section, we evaluate performance of the proposed location relevance mining techniques and demonstrate the effectiveness of the travelogue service.

3.7.1 Datasets

There are many sources of travelogues on the Web, ranging from Weblogs such as Windows Live Spaces to dedicated travel web-site such as TravelPod, IgoUgo and TravelBlog. We collected approximately 100,000 travelogues in English with location labels fallen in United States to form an English Corpus. Nevertheless, instead of relying on the location labels directly, we implemented a location extractor to extract locations mentioned in these travelogues, yielding 18,000 unique locations. Because some tasks require evaluation by human beings with travel-related background and knowledge, we also built a Chinese Corpus by collecting travelogues from Ctrip\(^5\), which consists of 94,000 Chinese travelogues related to around 32,000 locations in China.

3.7.2 Evaluation on LRM

Here, we first introduce the performance metrics and then present the results of our evaluation on location relevance mining.

3.7.2.1 Performance Metrics

In the experiments, Precision and Recall are used to evaluate the performance of location relevance classification; Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) are used to evaluate the performance of location relevance ranking. Since Precision and Recall are well known metrics, we only discuss NDCG and MAP.

\(^5\)http://www.ctrip.com
For location relevance ranking problem, given a travelogue \( t \), the NDCG score (NDCG\(_t\)) is computed according to the ranked location list \( L_t \) (i.e., the output of location relevance ranking), as 
\[
\text{NDCG}_t = n_t \sum_{j=1}^{||L_t||} \frac{2^{r_j} - 1}{\log(1+j)},
\]
where \( r_j \) is the rank of the location at the \( j\)th position of the location list \( L_t \), and \( n_t \) is a normalization constant, which is chosen so that a perfect ranking’s NDCG score is 1. The final NDCG score is averaged over the scores of all the travelogues. In this paper, the NDCG scores for ranked location lists which include top 1, 3, 5, 7, 9 and 11 locations are reported. MAP stands for the mean of average precisions over all the travelogues.

Given a travelogue \( t \) and a ranked location list \( L_t \), average precision is defined as 
\[
\text{AvePre}_t = \sum_{j=1}^{||L_t||} \frac{I(j)(R_j/j)}{R},
\]
where \( R \) and \( R_j \) denote the total number of relevant locations and the number of relevant locations before the position \((j+1)\), respectively, and \( I(j) \) is an indicator which takes value 1 if the location at position \( j \) is relevant and value 0 otherwise.

### 3.7.2.2 Evaluation Results

An important factor crucial to performance of location relevance classification and ranking is the location features extracted for mining. Thus, to facilitate our studies on impact of extracted features, we form the following four feature groups: (1) baseline group - we consider “is in title?” (\( F_1 \)) and “number of appearance” (\( F_2 \)) as two basic features, which are frequently used for text mining; (2) textual group - we consider all textual features; (3) geographical group - we consider all geographical features; (4) textual+geographical group - we consider both the textual and geographical features.

As there is no existing dataset for evaluation of location relevance mining, we have to built our own training dataset manually. Since we usually consider a destination in our trip as relevant locations, those pass-by/stop-by/nearby locations as partial relevant location, and others as irrelevant locations, we labeled 1,000 travelogues in the Chinese Corpus for our performance evaluation.

For location relevance classification, we consider relevant locations as positive cases and the other locations as negative cases. Among those 1,000 labeled travelogues, we randomly select 100 travelogues as the test set and the remaining travelogues for training. Figure 3.6(a) shows the precision-recall curves of location relevance classification with different feature groups. We found that the baseline group (i.e., whether the location appears in the title and the count of locations in a travelogue) already provides strong support for deciding relevant locations. However, classifier using baseline features is effective only to some extent. With more textual features, the location relevance classifier gain improvement. From the figure, we ob-
serve that geographical features perform classification very well, showing better performance than baseline features particularly when the precision is higher, as baseline features consider only very limited information. Also shown in Figure 3.6(a), combining textual and geographical features has the best performance because textual features and geographical features are independent. As a result, combining those two provide more information to assess whether a location is relevance or not.

In this work, co-training approach has been adopted to improve the classification performance (i.e., to combine approximately 1,000 labeled travelogues with approximately 93,000 unlabeled travelogues). In our experiment, we set the labeled data set as seed training set. As discussed in Section 3.5, a confidence threshold \( \theta \) has been used in our co-training algorithm to derive high-quality labeled training set. Here we estimate \( \theta \) based on the precision threshold value of \( \rho \), which is set as \( \rho = 0.95 \) in the test. In the co-training process, both textual-feature classifier and geographical-feature classifier are trained as high-precision classifiers (i.e., by setting \( \rho = 0.95 \)). In this experiment, after 4 round of iterations, co-training process terminates. The new training set is then used in classification. We compare the classification results by running the SVM classifier on the original training set (labeled as SVM) and new training set (labeled SVM+Co-Training). Figure 3.6(b) shows that co-training leads to a better precision-recall performance. As a matter of fact, this improvement can be expected, because both textual features and geographical features can be used to perform the classification task. Moreover, those two are independent and even complementary, so the combination of them enhance the classification performance as shown in Figure 3.6(a).
To show the inherent challenges in location relevance classification, we collect a training set with relevant and irrelevant locations, with all partial relevant location filtered out. Let $x$ denotes the feature vector of a location, we calculate $f(x) = w^*x + b$ (denoted as $f$-score) in the trained SVM model ($w^*$ is the optimized hyperplane, while $b$ is the bias term) for each location in the travelogues of testing dataset. Here, $f$-score indicates the distance to the classifier hyperplane. Notice that in the test dataset, locations are categorized into three groups, i.e., relevant locations, partial-relevant locations and irrelevant locations. For each group, we plot its $f$-score distribution in Figure 3.7. We find that, $f$-score distribution of partial-relevant location overlaps to some extent with one of relevant locations. This is due to the ambiguity in location relevance perceived by different people. In contrast to the location relevance classification problem, it seems to be easier to rank the relevance of different locations for a given travelogue. In Figure 3.7, we find the “centers” of the distributions with respect to each location relevance class differ. As shown, the irrelevant location class is positioned in the leftmost part while the relevant location class stands at the rightmost part.

Next, we evaluate the performance of location relevance ranking techniques, including location likelihood estimation (LLE) and learning to rank (L2R), using the same dataset. There are three labeled ranks in our dataset, i.e., relevant, partial relevant and irrelevant. As discussed earlier, NDCG@N and MAP are used as the performance metrics for the experiment.\footnote{Note that, we only consider relevant locations and other locations in measurement of MAP, since it considers binary cases in ranking performance evaluation.}

Since there is a smoothing factor $\lambda$ in the location likelihood estimation, we experimentally find the optimal setting of LLE by tuning $\lambda$ from 0.2 to 1.0 with a 0.2 step. Interestingly,
although LLE is based on the idea of language model, it shows a behavior different from a conventional language model. Specifically, the smoothing parameter in a language model is used to improve the document ranking performance but it has no effect on location relevance ranking, i.e., $\lambda = 1.0$, as shown in our experiments. The is possibly because location relevance ranking is constrained within the given travelogue. As a result, count of a location appearance is much more important than its global popularity.

Next, we show impact of extracted feature groups on LLE and L2R via NDCG@N test (see Figure 3.9(a)) and MAP test (see Figure 3.9(b)), respectively. For simplicity, only the best performance (when $\lambda = 1.0$) of LLE is plotted in these two figures. As shown L2R
consistently outperforms LLE. Particularly, since LLE is mainly based on word frequency, which is considered as feature $F_2$, the experiments demonstrate that both textual features and geographical features are indeed effective for improving the performance of location relevance ranking. Also note that the location relevance ranking problem is different from traditional document retrieval problem, where thousands and even millions of candidate documents are to be ranked for a given query. In location relevance ranking, the number of candidate locations in a given travelogue are usually no more than 15. Therefore, it is imperative to ensure all relevant locations ranked higher than the other locations. In the MAP test, we find that the learning to rank approach outperforms the location likelihood estimation approach significantly.

3.7.3 Evaluation on the Travelogue Service

Finally, we demonstrate the effectiveness of our travelogue service by evaluating the proposed travelogue digests, including theme locations, geographical scope, traveling trajectory and location snippet.

3.7.3.1 Theme Locations

In our travelogue service, theme locations can be acquired via 1) location relevance classification; or 2) author tagging. The effectiveness of our classification technique has been studied in the last section. On the other hand, author tagging can be assisted by location tag recommendation backed by our location relevance ranking techniques. Here, we evaluate the precision@N and recall@N of tag recommendation (where $N = 1, 3, 5, 7, 9, 11$). Figure 3.10(a) and Figure 3.10(b) show that both L2R has a much better performance than LLE (especially when $N = 3$ and 5) in support of location tag recommendation. These results are consistent with the findings in Figure 3.9(b).

3.7.3.2 Geographical Scope

Here, we would like to evaluate whether location relevance mining can help improve the accuracy for geographical scope identification. Since there is no existing dataset with labeled geographical scope, we use the travelogues labeled with theme locations to derive the ground truth and use labeled theme locations as inputs for our geographical scope computation algorithm.
We introduce two performance metrics, namely, \textit{average number of missing scopes} ($N_m$) (i.e., the correct scopes not found) and \textit{average number of wrong scopes} ($N_w$) (i.e., the scopes found incorrectly) to evaluate the performance. Let $S^* = \{l_1, l_2, \cdots \}$ denote the ground truth scopes for a given travelogue and $S$ be the result we compute. Given two locations $l$ and $l'$, let $r(l, l')$ denote that $l$ is neither ancestor nor descendant of $l'$ in the ontology hierarchy, e.g., $r(\text{Boston, Florida})$ holds for the example in Figure 3.3 but $r(\text{Siesta key, Florida})$ does not. Thus, we have $N_m = ||\{l \in S^*, \forall l' \in S, r(l, l')\}||$ and $N_w = ||\{l \in S, \forall l' \in S^*, r(l, l')\}||$.

We compare our approach which takes theme locations as input for geographical scope computation with a naive approach that considers all locations as input. As shown in Figure 3.11, our approach achieves a much better performance than the naive approach in terms of average number of wrong scopes, while remaining to be competitive in terms of average
focused location | noisy location(s) | snippet
--- | --- | ---
Battery Park | 94th Street, Manhattan, the World Trade Center, the Statue of Liberty | Actually, I usually start at the Upper Westside around 94th Street and go all the way to Battery Park at the southern tip of Manhattan. Battery Park is a great place to end up. The scenery is great with the water, the marina, the World Trade Center and the Statue of Liberty...
The Cloisters | Metropolitan Museum of Art, New York, 190th Street | The Cloisters, a branch of the Metropolitan Museum of Art, is not just a museum filled with beautiful medieval art, it’s a place to find serenity around New York. Located up at 190th Street, the Cloisters is a bit of a trek but it’s worth it...
Golden Gate Park | NY, Central Park | I call Golden Gate Park a smaller version of NY’s Central Park. It offers so many things to do that I never tire of jogging through it...

Table 3.2. Location snippet extraction for each individual travelogue.

number of missing scopes. Hence, location relevance mining which helps filter out irrelevant locations can improve the accuracy of geographical scope identification.

3.7.3.3 Traveling Trajectory

In this experiment, we prepared 100 travelogues with labeled traveling trajectories as the ground truth dataset. In order to evaluate the accuracy of our proposed trajectory extraction technique, we introduce a performance metrics, namely trajectory similarity \(\text{TS}(t_1, t_2)\), which is defined as the graph edit distance [123] between two traveling trajectories \(t_1\) and \(t_2\). Here, we compare our approach with a naive approach that forms a trajectory by considering all locations in a given travelogue as the POI set. As shown in Figure 3.12, our approach achieves a much better performance since both location relevance classification and POI set formation improve the trajectory extraction accuracy significantly.

3.7.3.4 Location Snippets

Here we evaluate the effectiveness of the extracted location snippets. In [82], sentences/small paragraphs with multiple different locations are ignored due to the lack of location relevance mining techniques. However, those ignored snippets may contain interesting information to the users, because point of interests/locations are usually clustered in geographical proximity and
mentioned close by in the text of travelogues. For example, Table 3.2 shows some interesting location snippets that we extract but missed in [82]. Take the location snippet for “Battery Park” as an example. There are quite a few “noise” locations in the snippet but they provide very informative knowledge about what to and see in Battery Park. In the snippet of “the Cloisters”, we find that other locations also provide informative knowledge about where “the Cloisters” locates, i.e., “… around of New York. Located up at 190th Street …”. Finally, travelogue authors often describe locations by comparison, e.g., “Golden Gate Park” vs. “Central Park” in the snippet for “Golden Gate Park”.

We conduct a user study to evaluate whether our location snippets are informative in terms of conveying information of the corresponding location. Based on the Chinese corpus, we prepared 20 travelogues, where each travelogue consists of more than 5 locations mentioned. Intuitively, if a small paragraph contains a location name $l$, then we consider this small paragraph should be describing this location in some way. Thus, a naive approach is to consider those small paragraphs as snippets for the corresponding location involved.

Twenty graduate students were asked to assess the snippets of all 20 travelogues in 1 to 5 ratings in three aspects: namely, (1) geographical-relevance (i.e., to what extent the snippets are describing the corresponding location), (2) comprehensiveness (i.e., to what extent the snippets provide rich information about the themes of the given travelogue), and (3) overall satisfaction. Accordingly, we want to demonstrate whether the proposed method can suggest snippets not only relevant to the corresponding location, but also comprehensive to provide interesting information. For each travelogue, we average all the users’ evaluation in all three ratings of each snippet set in a travelogue. Three approaches (i.e., the naive approach, topic
modeling approach [82] and the proposed approach) are compared as shown in Figure 3.13. Since topic modeling approach purges the snippets by deleting those containing multiple locations, it performs the best in terms of geographical relevance. However, our proposed approach also shows very competitive performance in terms of geographic relevance, because our location relevance mining helps to highlight theme locations. For the comprehensiveness, both naive approach and topic modeling approach show worse performance than our proposed approach does. The reason is that our proposed approach includes the location snippet as shown in Table 3.2, thus providing very comprehensive information against topic modeling approach. Notice that, the naive approach can not differentiate which location is relevant, thus showing the worst performance in the aspect of geographical relevance, and affecting the comprehensiveness consequently. Overall, our proposed approach shows the best performance and user satisfactory.

3.8 Summaries

In this chapter, we develop location relevance mining techniques to discover theme locations of travelogues in support of a proposed travelogue service. Due to inherent ambiguity of location relevance, we perform location relevance mining in two complementary angles, relevance classification and relevance ranking, to provide comprehensive understanding of locations in a travelogue. Furthermore, we explore the textual and geographical features of locations and adopt a co-training model to improve the classification. Built upon the mining result of location relevance mining, we develop techniques for provisioning of various digests, including theme locations, geographical scope, traveling trajectory and location snippets, in our travelogue service. Finally, we conduct comprehensive experiments on collected travelogues to evaluate the performance of our location relevance mining techniques and demonstrate the effectiveness of the travelogue service.
On the Semantic Annotation of Places in Location-Based Social Networks

In this chapter, we develop a semantic annotation technique for location-based social networks to automatically annotate all places with category tags which are a crucial prerequisite for location search, recommendation services, or data cleaning. Our annotation algorithm learns a binary support vector machine (SVM) classifier for each tag in the tag space to support multi-label classification. Based on the check-in behavior of users, we extract features of places from i) explicit patterns (EP) of individual places and ii) implicit relatedness (IR) among similar places. The features extracted from EP are summarized from all check-ins at a specific place. The features from IR are derived by building a novel network of related places (NRP) where similar places are linked by virtual edges. Upon NRP, we determine the probability of a category tag for each place by exploring the relatedness of places. Finally, we conduct a comprehensive experimental study based on a real dataset collected from a location-based social network, Whrrl. The results demonstrate the suitability of our approach and show the strength of taking both EP and IR into account in feature extraction.

4.1 Overview

With the increasing availability of GPS-enabled smart phones, rapid development of location-based services, and growing interests in on-line social networking, a number of location-based
social networking (LBSN) services such as Whrrl\(^1\), Foursquare\(^2\) and Facebook Places\(^3\) have emerged. These services allow users to explore places, write reviews, and share their locations and experiences with others. The number of available places in LBSNs is growing continuously.\(^4\) Many places have been labeled with useful tags such as restaurant or cinema, which are crucial for assisting users in searching and exploring new places as well as for developing recommendation services [24, 25, 26]. However, based on our analysis of data collected from Whrrl and Foursquare, about 30% of all places are lacking any meaningful textual descriptions. To address this problem, we develop a novel technique, namely semantic annotation of places (SAP), to automatically and precisely annotate all places with semantic tags for LBSNs.

![Figure 4.1. Users and places in an LBSN](image)

The problem of place semantic annotation can be formulated as predicting appropriate tags for a given place. In LBSNs, a place may be associated with multiple tags. For instance, a place associated with a tag restaurant may also be tagged with bar. Hence, place semantic annotation in LBSNs may be addressed as a multi-label classification problem [124, 125]. While multi-label classification techniques have been developed for many applications, such

---

\(^1\)www.whrrl.com  
\(^2\)www.foursquare.com  
\(^3\)www.facebook.com/places  
\(^4\)Points of Interest (POIs) are usually referred to as places in LBSNs.
as protein function classification [126], music categorization [127] and semantic scene classification [124], the problem has not been explored previously under the context of LBSNs, where we can only operate over user check-in activities (i.e., \((u, p, h) \in C\)) for certain places and time stamps.

We propose to address the place semantic annotation problem by learning a binary SVM for each tag in the tag space in order to realize the multi-label classification. To do so, a fundamental issue is to identify and extract a number of descriptive features for each place in the system. Selecting the right features is important because those features have a direct impact on the effectiveness of the classification task. As mentioned earlier, the only data resource we have is the user check-in activities at various places and times. Therefore, we explore the user behaviors and seek unique features of places captured in the check-in activities. Fortunately, human behaviors are not completely random, e.g., people usually visit restaurants for lunch at around noon. Moreover, people exhibit patterns in their activities, e.g., different places visited by the same person at the same time may be similar (e.g., having the same tags).

By leveraging the observations hinted in the above-mentioned examples, we extract features of places in two different but complementary aspects: 1) explicit patterns (EP) at individual places; and 2) implicit relatedness (IR) among similar places. Features extracted from EP, corresponding to a given place, can be derived from all check-ins at the place based on statistical analysis. In this paper, we propose to extract population features (e.g., number of unique visitors) and temporal features (e.g., distribution of check-in time) as semantic descriptions of specific places. On the other hand, we extract features from IR to capture the relatedness among places by exploiting the regularity of user check-in activities to similar places. Since only some places are tagged, we could make good use of IR by deriving descriptive features of a given place from its “related” places.

To facilitate extraction of features from IR, we develop a novel algorithm to build a network of related places (NRP) that captures the relatedness amongst places by exploring regularities of user check-ins to similar places. We propose a family of graph representations that capture the user-place and time-place relationships from the user check-in activities. We employ the Random Walk and Restart technique [128] on these graphs to estimate the relatedness of places in order to build an NRP. In the obtained NRP, place pairs with high relatedness values imply high similarity in the tag space and thus are linked. Accordingly, we derive the probability for a specific tag being labeled to a place (called label probability) from its linked (similar) places. This label probability is thus treated as a feature of IR, along with population and temporal
features derived from EP, to feed the binary SVM in our SAP algorithm.

In the following, first, we review related works regarding to multi-label classification and classification in networked data; next, we give an overview of the proposed semantic annotation of places (SAP) algorithm, describe how we extract EP and IR features, and detail the realization of our SAP algorithm; then we further discuss the issue of extracting features from IR and detail our approach, and conduct an empirical study using the collected Whrrl dataset and analyze our results; finally, we summary this work.

4.2 Related Work

In this section, we review a number of existing works in the areas of data mining, multi-label classification, and classification of networked data.

Due to the increasing availability of location-based services and GPS-enabled devices, real traces of user locations and activities have been collected and used in several studies [46, 51, 89]. A variety of approaches for projecting user trajectories from GPS data have been proposed, including particle filtering [53], Markov models [46], Dynamic Baysian Networks [51], and Eigenbehaviors [57]. Data traces used in these studies typically do not contain explicit information regarding user activities. The LBSN data investigated in our research is unique in two aspects: i) semantic tags associated with places provide rich information about categories and activities (e.g., food, restaurant, hotel, shopping, etc.); and ii) user check-ins logged in LBSNs usually are not continual, thus revealing partial views of user activities. These differences bring several new challenges to our research. The most related work is [89], which studies people’s naming preferences. The authors argue that people have different naming preferences under different contexts, and thus use a wide range of terms such as home and near Liberty Bridge to disclose their locations to others. Notice that, even though our work also explores people’s naming preferences on places, we focus on enriching places with semantic tags such as restaurant and cinema for supporting location search and retrieval.

Place semantic annotation has been formulated as a multi-label classification problem in this paper. Previous studies on multi-label classification have primarily been conducted in the application domains of text classification [129, 130], protein function classification [126], music categorization [127], and semantic scene classification [124]. In [129], BoosTexer, extending AdaBoost [131], has been developed to handle multi-label text categorization. In [130], a mixture model derived by expectation maximization (EM) has been trained to select the most
probable set of labels from the power set of possible classes. In [132], a set of binary SVM classifiers have been developed to realize multi-label classification for text classification. In [126], the notion of entropy has been extended to include multi-label data for gene expression in order to generate accurate rules for gene expression comprehension. In [133], geographical knowledge has been employed to help assign semantic tags to geo-tagged Flickr photos. Note that place semantic annotation for LBSNs is a new research topic that has not been studied previously.

To derive correlations amongst places from patterns of individual check-ins, we depict users and places as nodes of a bipartite graph as shown in Figure 5.1. We then construct a network of related places to facilitate classification. There exists some work on classifying networked data, which are generally of the same type such as web-pages or text documents connected via various explicit relations (e.g., hyperlinks [134]). Studies on simultaneously inferring interrelated values over networked data have been reported in [135, 136]. In [137], a simple univariate classifier, called the weighted-vote relational neighbor (wvRN), has been developed by obtaining a weighted average of the estimated class membership scores of the nodes’ neighbors. Moreover, similar to [135], a relaxation labeling method has been proposed for collective inference [138]. In [139], a case study on learning attributes of network data has been presented. In [140], a cautious collective classification that adopts only top-k most confidently predicted labels has been proposed. Gallagher et al. propose ghost edges to create edges between nodes based on the intrinsic structure of the networks to improve the classification of sparse labels [141].

4.3 Semantic Annotation of Places

We design a two-phase algorithm to address the place semantic annotation problem. The first phase takes care of the feature extraction, while the second phase handles the semantic annotation. The task of feature extraction explores two lines of ideas as discussed earlier in the Introduction. On the one hand, we explore the explicit patterns (EP) corresponding to a specific place to abstract aggregated user behaviors as population features and temporal features. On the other hand, we explore the implicit relatedness (IR) amongst places in order to formulate descriptive features of a given place from its similar places. Features derived from EP and IR are used to learn a binary SVM for each tag in the tag space in the semantic annotation phase. Given a place, the prediction by a specific SVM classifier decides whether this place belongs to the category of the corresponding semantic tag or not. After checking all SVM classifiers,
we obtain all qualified semantic tags for the place under examination.

### 4.3.1 Features from EP

Our goal is to extract discriminative EP features from places with the same tag. Intuitively, users behave differently at different places due to the nature of functions and activities offered by these places. As a result, different patterns, naturally formed in aggregated behaviors of visitors to various kinds of places, are embedded in the user check-in activities which are logged in LBSNs. In a check-in record, the most important information is user and time, besides the place itself. In the following, we propose to extract several population features and temporal features to depict places as below.

- **$F_1$ (total number of check-ins)** - Based on the observation from the Whrrl dataset, shown in Figure 4.2, we find the number of check-ins to a restaurant is usually larger than the number of check-ins to a hospital. Hence the number of check-ins, which is discriminative for the classification of places such as restaurants and hospitals, is a good population feature for semantic annotation.

- **$F_2$ (total number of unique visitors)** - This feature focuses on the number of unique visitors. Based on our analysis on the Whrrl dataset, we find $F_2$ to be a similar phenomenon to $F_1$. Thus, we aggregate the number of unique visitors at a specific location as the second population feature extracted from EP.

- **$F_3$ (maximum number of check-ins by a single visitor)** - As shown in Figure 4.3, people may check in a place tagged as restaurant for multiple times, while they may check in a

![Figure 4.2. Distr. of # of visitors](image1)

![Figure 4.3. Distr. of Max # of check-ins by a single visitor](image2)
hotel for only 1-2 times. Thus, the maximum number of check-ins by a single user at a place is useful to decide whether a place is a restaurant or a hotel. We use it as the third population feature extracted from EP.

![Figure 4.4. Distr. of check-in time (day)](image1)

![Figure 4.5. Distr. of check-in time (hour)](image2)

- $F_4$ (*distribution of check-in time in a week*) - We analyze the distribution of check-ins at different categories of places over the days of a week. As shown in Figure 4.4, users check in college campuses more often on weekdays than on weekends. On the contrary, they check in bars on weekends more frequently than on weekdays. Since there are different distributions of check-in days for different kinds of places, we consider the distribution to be a very useful temporal feature.

- $F_5$ (*distribution of check-in time in 24-hour scale*) - By plotting the distribution of check-ins in the 24-hours time scale, we show in Figure 4.5 two very different distribution patterns corresponding to two kinds of places (i.e., restaurant and shop). There are clearly two peak times, corresponding to lunch and dinner periods, for places associated with the tag *restaurant*. On the other hand, shopping time looks like a normal distribution with most activities between 7:00am and 8:00pm, while there is no obvious peak shopping time observed. These observations regarding the patterns of massive visitors at different kinds of places provide strong support that check-in time distributions in 24-hours time scale is a good temporal feature for semantic annotation.

Note that, besides of the aforementioned patterns, check-in activities at different places may show seasonal patterns, e.g., most people go to ski areas during winter. However, due to the limited time span in the period of data collection, we only consider $F_4$ and $F_5$ as the temporal features in this study.
4.3.2 Features from IR

As discussed in [57], there is regularity in people’s activities. Take one of the Whrrl users as an example. We find that the user visits places in the *performing arts and entertainment* category (including museums and galleries) in the morning (at around 10:00am), visits places for food at lunch/dinner time, and usually goes shopping at around 4:00pm. Such regularity appears in certain users and thus can be used for correlating similar places. However, extracting features from implicated relatedness (IR) among similar places (e.g., checked in at the same time) is not as straightforward as extracting features from EP.

To capture the relatedness among places and extract discriminative features from IR, our approach is to build a *network of related places* (NRP). In an NRP, places are linked based on their relatedness, measured by the information provided in user check-ins through the Random Walk and Restart technique [128]. Upon the NRP, we determine the label probability for each place by exploring the relatedness of places. As such, the label probability derived from IR serves as a feature for classification. Details of feature extraction from IR will be introduced in Section 4.4.

4.3.3 Semantic Annotation

After the feature extraction phase, features derived from both EP and IR are used as inputs for the semantic annotation phase to learn a binary SVM for each tag. We choose SVM as the binary classifier because it has shown excellent performance in similar tasks. In our approach, all places are used for each binary SVM training, i.e., an instance labeled with the specific semantic tag under examination is considered as a positive example, while places without this label serve as negative examples. For instance, places tagged *shopping* are positive examples for a classifier for shopping, but negative examples for a classifier for *nightlife*. For a place to be annotated with such a semantic tag, a binary classifier for each tag is expected to classify the place as an instance of the tag class. As a result, the place will be automatically annotated with proper semantic tags.

4.4 IR Feature Extraction

To facilitate the extraction of features from implicit relatedness among similar places, we develop a new algorithm that builds a *network of related places* (NRP) to capture the relatedness
between places, and further derive the label probability as an IR feature for each tag and each place upon the obtained NRP as follows.

### 4.4.1 Network of Related Places

![Entropy distribution](image)

![Graph representations of LBSN data](image)

**Figure 4.6.** Entropy distribution  
**Figure 4.7.** Graph representations of LBSN data

As discussed earlier, we intend to exploit the behavior patterns of LBSN users for semantic annotation of places. By analyzing the Whrrl dataset, we find that the check-in activities of Whrrl users do exhibit a strong regularity that supports our idea. In the analysis, we study the diversity of places individual users visit by computing the entropy of semantic tags (in eight activity categories) in their check-ins. The result is shown in Figure 4.6. Smaller entropy indicates that places checked in by LBSN users usually have similar semantic tags. From the figure, we observe that about 22.07% of users have their check-in entropies smaller than 0.5 and about 75% of users have their entropies smaller than 1. In other words, a great number of Whrrl users visit similar places. Therefore, we build a *user-place (UP)* graph, which consists of users and places connected in accordance with the check-in records. Let $c(u_i, p_j, h_s) \in C$ denote a check-in record describing that user $u_i$ has visited place $p_j$ at time stamp $h_s$, where $C$ is the collection of all check-in records. Definition 4.1 gives the formal definition of the UP graph.

**Definition 4.1. User-Place (UP) Graph.** denoted by $G_u(V_u, E_u)$, is an undirected bipartite graph (as illustrated in Figure 4.7(a)). Here $V_u = U \cup P$, where $U$ and $P$ are the sets of all users and places, respectively, and $E_u = \{e_{i,j} | c(u_i, p_j, \cdot) \in C\}$, where $c(u_i, p_j, \cdot)$ denotes that user $u_i$ has visited place $p_j$ at some time. In this graph, each edge $e_{i,j} \in E_u$ is associated with a weight $w_{i,j}$, denoting how often user $u_i$ has visited place $p_j$. Formally, $w_{i,j} = \left|\{c(u_i, p_j, h_s)\}\right|$. 

On the other hand, the timing of check-ins at similar places may be similar. Therefore, we build a *temporal-place (TP)* graph, where the time space is discretized into twenty-four hours, to capture the similarity between places in the temporal dimension. Definition 4.2 gives the formal definition of the TP graph.

**Definition 4.2. Temporal-Place (TP) Graph.** denoted by $G_t(V_t, E_t)$, is an undirected bipartite graph (as illustrated in Figure 4.7(b)). Here, $V_t = H \cup P$, where $H$ and $P$ are the sets of all times (i.e. hours) and places, respectively, and $E_t = \{ e_{js} | c(\cdot, p_j, h_s) \in C \}$, where $c(\cdot, p_j, h_s)$ denotes that a user has visited place $p_j$ at time $h_s$. In this graph, each edge $e_{js} \in E_t$ is associated with a weight $w_{js}$, denoting how often $p_j$ has been checked in at time $h_s$. Formally, $w_{js} = \left| \{ c(u, p_j, h_s) \} \right|$.

In the aforementioned graphs, places are indirectly connected through users and times. In the following, we propose to use the Random Walk and Restart method [128] to estimate the relatedness between pair-wise places in both user and time aspects in order to build a network of related places (NRP), where edges are explicitly established among places according to their relatedness values.

To construct the NRP, we need to derive the relatedness of places from the UP graph and TP graph. In our approach, we first obtain two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ for every pair of places $p_x, p_y \in P$ through Random Walk and Restart (RWR) over the UP and TP graphs, respectively, and then combine them into one relatedness value between place nodes in the NRP. Here we only describe how RWR proceeds on the UP graph since operations on the TP graph are similar. Given a node $x$, an RWR is performed by randomly following one of its links to another node $y$ of the UP graph based on the transition probabilities of these links, in addition to a probability $a$ to restart at node $x$. For the UP graph, we prepare a random walk transition matrix that consists of two zero matrices, i.e., user-user matrix ($UU$) and place-place matrix ($PP$), and a user-place matrix (actually $UP$) and its transpose $UP^T$, where the probability of transiting between a place $p_j$ and a user $u_i$ is proportional to $w_{ij}$ (in Definition 4.1). The stationary, or steady-state, probabilities for each pair of nodes can be obtained by recursively processing Random Walk and Restart until convergence. The converged probabilities (i.e., relatedness values) give us the long-term visiting rates from any given node to any other node. In this way, we can calculate the relatedness of all pairs of location nodes, denoted by $r^p_{x,y}(\forall p_x, p_y \in P)$.

Note that the transition matrix for the TP graph can be derived in a similar way. Accordingly, we can obtain two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ for a pair of places $p_x, p_y$ from the
UP and TP graphs. Since both user and time information can help relate the semantic tags of places, we estimate the overall relatedness \( r_{x,y}^p \) between place pairs \( p_x, p_y \) by integrating them as follows.

\[
r_{x,y}^p = \eta r_{x,y}^u + (1 - \eta) r_{x,y}^t \quad \forall p_x, p_y \in P
\]

where \( \eta \) is a smoothing factor between 0 and 1. Based on the formula, places checked in by the same user at around the same time show strong relatedness because both relatedness from user and time aspects are considered.

Finally, we build a network of related places (NRP), where each place is connected to places with top-k relatedness values. More specifically, an NRP is defined as follows.

**Definition 4.3.** A network of related places NRP = \( \{P, E\} \) is a directed graph, consisting of only places. For each place \( p_i \in P \), let \( P_{ik}^i \) denote the set of top-k related places to \( p_i \). Thus, \( E = \{e(x, i) | \forall p_x \in P, p_x \in P_{ik}^i\} \). Here \( e(x, i) \) is a directed edge from \( p_x \) to \( p_i \).

### 4.4.2 Label Probability Derivation

As mentioned before, in a real LBSN, only some places have tags. The idea of extracting features from IR is to derive descriptive features of a given place from its “related” and tagged places. The network of related places (NRP) is constructed by connecting similar places together, so we aim to infer the tags of a given place by the tags of its neighbors. In order to derive an IR feature for use in the SVM, we derive the probability for a place to be labeled with a given semantic tag from its neighbors. More specifically, the label probability of a place can be estimated from the label probability of its neighbors recursively [134]. Let \( N_i \) be the set of immediate neighbors which have edges pointing to place \( p_i \), and \( y_i \) be a variable denoting a tag of place \( p_i \). For all possible tags \( t \in T \), we adopt the relaxation labeling method [138] to find the final \( Pr(y_i = t | N_i) \) \((t \in T)\) for each place \( p_i \). Relaxation labeling freezes the current estimations of each \( p_i \) so that, at round \( n+1 \), all places will be updated based on the estimations from round \( n \). As shown below, the label probability of \( p_i \) is calculated by considering both the weighted average of the label probabilities of places in \( N_i \), and the current label probability of \( p_i \) itself.

\[
Pr^{(n+1)}(y_i = t | N_i) = \beta_i^{(n+1)} \frac{1}{Z} \sum_{p_j \in N_i} r_{j,i}^p Pr^{(n)}(y_j = t | N_j) + (1 - \beta_i^{(n+1)}) Pr^{(n)}(y_i = t | N_i)
\]
where \( Z = \sum_{p_j \in N_i} r^p_{ji} \) is a normalization term and \( r^p_{ji} \) is the relatedness between places \( p_j \) and \( p_i \), and \( Pr^{(n)}(y_i = t|N_i) \) denotes the estimation of \( Pr(y_i = t|N_i) \) at round \( n \). Note that, we define the
\[
\beta^{(n+1)}_t = \beta^{(n)}_t \alpha,
\]
where \( \beta^{(0)}_t (t \in T) \) is a constant between 0 and 1, and \( \alpha \) is a decay factor, i.e., \( 0 < \alpha < 1 \). Note that in our daily activities, some of them exhibit more regularity than others (e.g., restaurants against shops). Therefore, we employ different \( \beta^{(0)}_t \) values for different semantic tags. Note that different tags have different \( \beta^{(0)}_t \) values, where label probability calculation with larger \( \beta^{(0)}_t \) settings converges slower than the one with smaller \( \beta^{(0)}_t \). More importantly, a larger \( \beta^{(0)}_t \) implies that the label probability of a given place should be estimated not only according to the immediate neighbors, but also influenced by places in multi-hops away as there are multiple rounds of calculation. A smaller \( \beta^{(0)}_t \) suggests that the label probability is only affected by close-by neighbors as there are very few rounds of calculation.

Here, we discuss how to initialize \( Pr^{(0)}(y_i = t|N_i) \) for each \( p_i \in P \). Let \( P_{test} \) denote the set of testing places, i.e., places that do not have any semantic tags. The label probability of a testing place is initialized as 0.5; while the label probability of a place already labeled with semantic tags is initialized as 1 or 0 according to the labels. Formally, the label probability is initialized as follows.

\[
Pr^{(0)}(y_i = t|N_i) = \begin{cases} 
0.5 & \text{if } p_i \in P_{test} \\
1 & \text{if } p_i \in P - P_{test} \text{ and } t \in T_i \\
0 & \text{if } p_i \in P - P_{test} \text{ and } t \notin T_i 
\end{cases}
\]

Once we get the label probability estimation for each possible tag on a place \( p_i \), they are treated as IR features for SVM training. Note that features extracted from IR do not consider the explicit patterns exhibited in each individual place. Thus, in our SAP algorithm, we propose to combine features extracted from both EP and IR to address the problem of semantic annotation of places.

### 4.5 Performance Evaluation

In this section, we conduct a comprehensive set of experiments to validate our proposed ideas and evaluate our SAP algorithm in terms of three different feature sets: i) features extracted
from EP, ii) features extracted from IR, and iii) combination of i) and ii). Here, we use one of the most popular classification toolkits, LIBSVM [142], as the binary SVM classifier. In the following, we first discuss the collected dataset and the preprocessing steps for experiments, then introduce the metrics employed to evaluate the performance, and finally analyze the experiment results.

### 4.5.1 Dataset Description

We crawled the Whrrl website, a representative LBSN, for a month to collect a dataset consisting of 5,892 users, 53,432 places and 199 types of tags.\(^5\) Among those places, 20% of them are not specified with any semantic tags. In the vocabulary of semantic tags, we find that a lot of tag words sharing the same topic could be grouped in the same category. For example, Pizzerias, Coffee, Bakeries, Snacks, Delis, Cafes, Ice Cream and etc, all belong to the same category, namely, *Restaurant & Food*. Without loss of generality, we build a tag hierarchy based on Yelp\(^6\) to merge those 199 semantic tags into 21 categories to simplify the task of place semantic annotation. We show the top eight major categories and their corresponding percentages in Table 4.1. As shown, *Restaurants & Food*, *Shopping*, *Nightlife* are the most popular check-in places in Whrrl, i.e., 74% of places are within these three categories. Furthermore, we find that about 33.5% of places belong to multiple categories in our dataset.

<table>
<thead>
<tr>
<th>Category</th>
<th>z(%)</th>
<th>Category</th>
<th>z(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restaurants&amp;Food</td>
<td>37(%)</td>
<td>Hotel &amp; Travel</td>
<td>4(%)</td>
</tr>
<tr>
<td>Shopping</td>
<td>18(%)</td>
<td>Arts &amp; Entertainment</td>
<td>3(%)</td>
</tr>
<tr>
<td>Nightlife</td>
<td>19(%)</td>
<td>Health and Medical</td>
<td>2(%)</td>
</tr>
<tr>
<td>Active life</td>
<td>5(%)</td>
<td>Beauty and Spas</td>
<td>2(%)</td>
</tr>
</tbody>
</table>

*Table 4.1. Categories and their percentages (z\(\%\))*

In order to conduct the experiments, we pre-process this raw dataset to obtain a ground-truth dataset for performance evaluation. First, places in the ground-truth dataset should have category tags, so we filter out those places without category tags. Next, since we are interested in exploring user behaviors, users who have less than 40 check-in records are not included in the ground-truth dataset. Third, we calculate the activity entropies of those users and select the users and their places with entropies less than 0.5 as the dataset to conduct performance

\(^5\)Unfortunately, we cannot obtain the check-in time in Foursquare. Thus, we conduct the performance evaluation only upon the whrrl dataset.

\(^6\)http://www.yelp.com
Figure 4.8. Performance comparison

evaluation. Moreover, we randomly remove the category tags of $x\%$ places (named testing places, and $x\% = 10\%, 20\%$ and $40\%$ with default value $20\%$) over the ground-truth dataset. The SAP algorithm is used to recover the category tags for those testing places.

4.5.2 Performance Metrics

Given a testing place set $P_{test}$, we conduct a performance evaluation by measuring the following four metrics: hamming loss, one-error, coverage and average precision, as they are widely employed in previous multi-label classification studies [129, 125]. Hamming loss aims to measure the accuracy of the predicted tag set against the ground-truth tag set associated with a testing place. The other three metrics concern the ranking of tags annotated by the SAP algorithm, i.e., we consider that SAP performs well when the ground-truth tags are ranked high
in the predicted ranked tag list. Note that although we define place semantic annotation as a classification problem, LIBSVM provides probability output \([143]\) (i.e., the probability of the corresponding label), which can be used to rank the semantic tag for each place. Let \(Pr(t_x|f_i)\) be the probability output for place \(p_i\) being with tag \(t_x (\in T)\), where \(f_i\) denotes the set of features of \(p_i\). According to \(Pr(t_x|f_i)\), we get a ranked list of semantic tags, denoted as \(Y_i\), where semantic tags with the highest \(Pr(t_x|f_i)\) are ranked at the top.

Hamming Loss \((hl_{P_{test}})\): evaluating how many times a place-tag pair is misclassified, i.e., a tag not belonging to the place is predicted or a tag belonging to the place is not predicted. Formally, \(hl_{P_{test}} = \frac{1}{|P_{test}|} \sum_{p_i \in P_{test}} HD(\vec{T}_i, \vec{Y}_i)\), where \(T\) is the whole tag space, \(\vec{T}_i\) and \(\vec{Y}_i\) are the ground-truth and predicted tag vectors for testing place \(p_i\), and \(HD(\vec{T}_i, \vec{Y}_i)\) is the hamming distance between \(\vec{T}_i\) and \(\vec{Y}_i\). In the ground-truth tag vector \(\vec{T}_i\) of a place \(P_{test}\), the vector element corresponding to a tag \(t\) is set to 1 if \(t\) is associated with \(P_{test}\); otherwise, it is set to 0. The predicted vector \(\vec{Y}_i\) is generated by the SAP algorithm accordingly.

One-error: evaluating how many times the first (or top) ranked predicted tag is not in the ground-truth tag set of the place. Formally, \(\text{one-error}_{P_{test}} = \frac{1}{|P_{test}|} \sum_{p_i \in P_{test}} f(\arg\max_{t_x \in T} Pr(t_x|f_i) \notin T_i)\), where for any predicate \(\pi\), \(f(\pi)\) equals 1 if \(\pi\) holds and 0 otherwise.

Coverage: evaluating how far we need, on average, to go down the list of predicted tags \((Y_i)\) in order to recover all the ground-truth tags associated with the place \(p_i\). Let \(R(x)\) denote the rank of \(t_x\) in the ranked list \(Y_i\) generated by SAP. Formally, \(\text{coverage}_{P_{test}} = \frac{1}{|P_{test}|} \sum_{p_i \in P_{test}} \max_{t_x \in T_i} R(x) - 1\).

Note that one-error and coverage measures are not sufficient for evaluating our SAP algorithm, which may achieve good coverage but suffer high one-error, or vice versa. Thus we introduce the average precision, which takes the ranking positions of all ground-truth tags into consideration, to evaluate the predicted ranked tag list.

Average Precision (AP): Given a place \(p_i \in P_{test}\) and a ranked tag list \(Y_i\) generated by our SAP algorithm, the average precision for a test place \(p_i\) is defined as \(\text{AvePrec}_i = \sum_{j=1}^{|[T_i]|} \frac{I(j)(n_j/j)}{|[T_i]|}\), where \(|T_i|\) and \(n_j\) denote the total number of ground-truth tags and the number of ground-truth tags before the position \((j + 1)\) in the tag list \(Y_i\), respectively, and \(I(j)\) is an indicator which
takes value 1 if the tag at position \( j \) is a ground-truth tag and value 0 otherwise. Therefore, the overall average precision is measured as

\[
AP_{P_{\text{test}}} = \frac{1}{|P_{\text{test}}|} \sum_{p_i \in P_{\text{test}}} AvePrec_i.
\]

### 4.5.3 Experimental Results

As mentioned earlier, we conduct a series of experiments to evaluate the proposed SAP algorithm by comparing three different feature sets. We label the results obtained using features derived from EP and IR by \( \text{EP} \) and \( \text{IR} \), respectively, and label the results obtained using all features by \( \text{SAP} \). We also perform sensitivity tests on a number of tuning parameters and different mark-off rates, as well as discretized and continuous representations of temporal information. Note that we use \( \eta = 0.2 \), \( k = 5 \) and \( \beta \) as listed in Table 4.2 as the default parameter settings throughout the experiment.\(^7\)

<table>
<thead>
<tr>
<th>Category</th>
<th>( \beta )</th>
<th>Category</th>
<th>( \beta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restaurants&amp;Food</td>
<td>0.9</td>
<td>Hotel &amp; Travel</td>
<td>0.3</td>
</tr>
<tr>
<td>Shopping</td>
<td>0.1</td>
<td>Arts &amp; Entertainment</td>
<td>0.1</td>
</tr>
<tr>
<td>Nightlife</td>
<td>0.9</td>
<td>Health and Medical</td>
<td>0.1</td>
</tr>
<tr>
<td>Active life</td>
<td>0.1</td>
<td>Beauty and Spas</td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Table 4.2.** Optimal \( \beta \) settings under \( \eta = 0.2 \) and \( k = 5 \)

### 4.5.3.1 Overall Performance

In order to evaluate the performance of our SAP algorithm in detail, we not only show the performance over the entire dataset (labeled with \textit{overall} as show in Figure 5.4), but also the performance for subsets of testing places in the same category (according to the ground-truth). More specifically, we show three categories of places: \textit{Restaurants & Food} (labeled with \textit{Res}), \textit{Nightlife} (labeled with \textit{NL}) and \textit{Shopping} (labeled with \textit{Sh}). These categories were chosen since they constitute the majority (i.e., about 74\% ) of all places. As shown in Figure 5.4, under the default setting, \( \text{SAP} \) shows the best performance consistently, while both \( \text{EP} \) and \( \text{IR} \) also demonstrate good strength for the task of semantic annotation in LBSNs. Note that \( \text{EP} \) performs better than \( \text{IR} \) for the places in the groups of \textit{Res} and \textit{NL}, particularly for the

\(^7\beta \) refers to \( \beta_1^{(0)} \) in this experiment.
performance metrics one-error, coverage and average precision. The reason is that most people have the same routine for activities in those categories. As a result, those activities have very distinctive characteristics, such as the distribution of check-in times extracted from EP. Thus, EP is able to tag these kinds of places very well. On the other hand, IR shows great strength in labeling places with shopping tags. Regularity of shopping activities of individuals helps to discover the shopping places from other related shopping places, although different people may go shopping at different times.

4.5.3.2 Tuning Parameters

Next, we test the impact of tuning parameters, including $\beta$, $\eta$ and $k$, on classification performance of SAP for Res, NL and Sh. Notice that the impact of each tuning parameters is tested by fixing all other parameters in default settings.

![Figure 4.9. Impact of $\beta$](image)
As mentioned earlier, the optimal settings for the classifiers for different categories are different as shown in Table 4.2. Note that $\beta$ tunes the influence from immediate neighboring places and places in multiple hops away. Here, we check the impact of $\beta$ on the performance of classification, with particular interests in Res, NL and Sh. As shown in Figure 4.9, $\beta$ has significant impact on the classification performance for all categories. Both Res and NL show very similar behavior with the variation of $\beta$. The best performance setting of $\beta$ for Res and NL is 0.9, implying that the label probability of a given place should be estimated not only according to its immediate neighbors but also the places in multiple hops away. The reason is that in an NRP, Res (NL) places are clustered together, thus a larger $\beta$ can provide more robust and accurate estimation. On the other hand, the best $\beta$ setting of Sh is 0.1, indicating that the label probability estimation of shopping places is very sensitive to the influence from their neighbors. A smaller $\beta$ suggests that the label probability of a given place is only affected by
immediate neighbors. We find that Sh places are usually not clustered as well as Res because the regularity of Sh activities are not as regular as Res activities. Thus, it is better to only use information from immediate neighbors to estimate label probability for Sh places.

![Hamming Loss](image1)

![One-error](image2)

![Coverage](image3)

![Average Precision](image4)

**Figure 4.11. Impact of k**

In Figure 4.10, we show our test on the parameter η, which is used to tune the weight of place relatedness values computed from user and time aspects. As shown, we find the impact of η on classification of Res and NL places is very limited. A possible reason is that places in both Res and NL categories are well clustered according to either common users or common time. Another reason is that the default β setting for Res and NL is 0.9, which means influence from places even in multiple hops away is contributing to accurate estimation of label probability. The selection of immediate neighbors is not that sensitive to η as long as places in those categories are clustered together. Nevertheless, η does affect the performance for Sh places, as label probabilities of Sh places are mostly affected by immediate neighbors.
(i.e., $\beta = 0.1$ for Sh activities). As shown in Figure 4.10, when $\eta = 0.2$, SAP shows the best performance on classification of Sh places in terms of the metrics of hamming loss, one error and average precision; when $\eta = 0.9$, the coverage performance turns out to be the best for Sh places. Accordingly, we consider $\eta = 0.2$ as a proper parameter setting and use it as the default setting throughout the experiment. It implies that both user and time are important to discover similar places through user behavior, particularly for the classification of Sh places. Besides, as the majority of check-ins for a visitor are usually Res places, time information plays an important role to link similar Sh places through the regular behaviors of people.

We further test the tuning parameter $k$ in Figure 4.11, where $k$ determines the number of neighboring places for a given place in an NRP. Similarly, we find that the variation of $k$ has almost no impact on the classification performance of Res and NL places since places in both categories are clustered together. However, the selection of $k$ affects the performance of classification for Sh places. In Figure 4.11, the best setting of $k$ for the classification of Sh places is different for various performance metrics. Nevertheless, we find that $k$ should be set to a proper value, in order to avoid the noise introduced by a large number of neighboring places.

### 4.5.3.3 Test on Mark-off Rate

Here, we investigate the impact of different mark-off rates to the performance of EP, IR and SAP. As shown in Figure 4.12, the performance of algorithms with different feature sets all degrade to some extent as the mark-off rate increases. Nevertheless, SAP shows the best performance consistently over all mark-off rates as it includes all the features.

### 4.5.3.4 Test on Continuity of Time

Check-in time is continuous in the temporal dimension, even though we simplify it as discrete twenty-four hours in the initial design of TP graph. Here, we investigate the impact of the continuity of time on the performance of SAP. In order to capture the continuity of time, we propose a method to smooth hours following the intuition that a user who checks in a place at time $h_s$, would probably check in similar places around the times $h_{s-1}$ and $h_{s+1}$, where $h_{s-1}$ and $h_{s+1}$ are adjacent times to $h_s$. More specifically, for each check-in at place $p_j$ and time $h_s$, we establish additional $m$ edges to the $m$ most adjacent time nodes beside the time node $h_s$ during the TP graph construction. For example, if $h_s$ presents 20:00 and $m = 1$, we establish edges
from the place to the time nodes 19:00 and 21:00, in addition to 20:00 in the construction of the TP graph.

Finally, we test the SAP algorithm, with \( m = 1 \) and \( \beta, \eta \) and \( k \) following the aforementioned default settings. The SAP algorithm following the initial design is denoted as Discrete-SAP, while the SAP algorithm with smoothed-hour TP graph design denoted by Smoothed-SAP. As shown in Figure 4.13, the impact on classification of Res and NL places are marginal, since places in those categories have been clustered together with Discrete-SAP. However, considering continuity of time does help improve the classification performance for Sh places, as shopping places checked in around the same time period (although in different hours) are possibly discovered as neighboring places in an NRP.
### 4.6 Summary

In this chapter, we investigate the place semantic annotation problem, which aims to automatically annotate all places with semantic tags in location-based social networks. Such tags are a crucial pre-requisite for location search, recommendation services, or data cleaning. In order to tackle this problem, we propose a novel semantic annotation algorithm which learns a binary SVM for each tag. Based on the check-in behavior of users, we extract features of places from two aspects: explicit pattern (EP) at individual places and implicit relatedness (IR) among similar places. Specifically, we extract EP features by aggregating user check-in behaviors to the corresponding places and extract IR features by exploiting the place relatedness exhibited by regularity of user behavior. Finally, we conduct a comprehensive experimental study based on a real dataset collected from Whrrl. The results demonstrate the suitability of our approach and
also support the assumption that both EP and IR need to be taken into account. Particularly, most people follow the same and distinctive pattern to visit restaurants and nightlife places. Thus, features extracted from EP hold very powerful discriminative capability. On the other hand, against EP, features from IR are excellent for tagging places related to shopping because some individuals exhibit strong patterns in certain shopping activities.
In this chapter, we aim to provide a point-of-interests (POI) recommendation service for the rapid growing location-based social networks (LBSNs), e.g., Foursquare, Whrrl, etc. Our idea is to explore user preference, social influence and geographical influence for POI recommendations. In addition to deriving user preference based on user-based collaborative filtering and exploring social influence from friends, we put a special emphasis on geographical influence due to the spatial clustering phenomenon exhibited in user check-in activities of LBSNs. We argue that the geographical influence among POIs plays an important role in user check-in behaviors and model it by power law distribution. Accordingly, we develop a collaborative recommendation algorithm based on geographical influence based on naive Bayesian. Furthermore, we propose a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence. Finally, we conduct a comprehensive performance evaluation over two large-scale datasets collected from Foursquare and Whrrl. Experimental results with these real datasets show that the unified collaborative recommendation approach significantly outperforms a wide spectrum of alternative recommendation approaches.
5.1 Overview

With the rapid development of mobile devices, wireless networks and Web 2.0 technology, a number of location-based social networking services, e.g., Loopt\(^1\), Brightkite\(^2\), Foursquare\(^3\) and Whrrl\(^4\), have emerged in recent years.\(^5\) These LBSNs allow users to establish cyber links to their friends or other users, and share tips and experiences of their visits to plentiful point-of-interests (POIs), e.g., restaurants, stores, cinema theaters, etc. In LBSNs, a POI recommendation service, aiming at recommending new POIs to users in order to help them explore new places and know their cities better, is an essential function that has received a lot of research momentum recently [94, 95].

![Graph representation of user-user friendship and user-location check-in activity in a LBSN](image)

**Figure 5.1.** Graph representation of user-user friendship and user-location check-in activity in a LBSN

Indeed, facilitating POI recommendations in LBSNs is a promising and interesting research problem because valuable information such as the “cyber” connections among users as well as the “physical” interactions between users and locations have been captured in the systems. Nevertheless, these information have not been fully explored in prior research studies relevant to POI recommendations. For example, Zheng et. al. have extracted visited locations from GPS trajectory logs of mobile users for location recommendations [94, 95]. However, their studies consider neither the social links between users nor the interactions between users and locations in the recommendation process. In this paper, we aim to exploit the unique geographical implications embedded in users’ interactions with locations, in addition to applying

\(^{1}\)www.loopt.com

\(^{2}\)www.brightkite.com

\(^{3}\)www.foursquare.com

\(^{4}\)www.whrrl.com

\(^{5}\)These services are often referred as location based social networks and thus abbreviated as LBSNs in the paper.
the social influence from users’ friends, for POI recommendations in LBSNs.

Users and POIs are two essential types of entities in LBSNs. As illustrated in Figure 5.1, users in an LBSN, denoted as \( u_1, u_2, u_3, u_4 \), are interconnected via social links to form a user social network. Moreover, POIs, denoted as \( l_1, l_2, \ldots, l_6 \), are connected with users via their “check-in” activities, which generally reflects the users’ tastes on various POIs. Finally, as also logically illustrated in the figure, the POIs, geocoded by \( \langle \text{longitude, latitude} \rangle \), are constrained geographically. To make recommendations of POIs to users, obviously the records of previous user check-in activities are very useful. With the availability of such information in LBSNs, an intuitive idea for supporting POI recommendations is to employ the conventional collaborative filtering (CF) techniques by treating POIs as the “items” in many successful CF-based recommender systems. The basic argument for this idea is that users’ tastes can be deduced by other users who exhibit similar visiting behaviors to POIs in previous check-in activities. Thus, user-based or item-based collaborative filtering techniques may be applicable to POI recommendations. Additionally, the social network of users, which is handily available in the LBSN, can be explored to enhance performance of POI recommendations. Recent studies have argued that social friends tend to share common interests and thus can be used in the process of collaborative filtering for making recommendations [144, 145, 146, 147, 148].

While the ideas above aim to explore the essential information available in LBSNs, i.e., the user-location interactivities and user-user social links, we argue that the geographical influence naturally existing in the activities of users and their geographical proximities cannot be ignored. According to Tobler’s First Law of Geography “Everything is related to everything else, but near things are more related than distant things” [149]. Thus, a user intuitively tends to visit nearby POIs. There are two major implications that can be derived from this intuition for POI recommendations: (1) people tend to visit POIs close to their homes or offices; and (2) people may be interested in exploring nearby POIs of a POI that they are in favor of, even if it is far away from home, e.g., a user may explore some restaurants and shops around Time Square when she goes there for a broadway show. Due to the geographical nature of the LBSNs, we believe the geographical influence between users and POIs as well as that amongst POIs are as important as the social influence amongst users, which as indicated earlier may play a positive role for supporting POI recommendations in LBSNs. In short, we are interested in studying the impact of geographical influence and social influence on POI recommendations in LBSNs.

Our approach for supporting POI recommendation service in LBSNs is to develop effective collaborative recommendation techniques that discover POIs of users’ interests by incorporat-
ing the three complementary factors: i) user preference of POIs; ii) social influence; and iii) geographical influence. Notice that users’ implicit preferences of POIs can be derived from their check-in activities on POIs. By considering two users who have checked into a lot of common POIs as similar users, we may discover the implicit preference of a user through the previous check-in activities of her similar users. Recall the example in Figure 5.1. Since $u_1$ and $u_2$ shares many commonly visited POIs, they may be considered as similar users who are assumed to share similar check-in behaviors, i.e., preference of POIs. As a result, $l_1$ is a good candidate for recommendation to user $u_2$ since $u_1$ has visited this POI before. On the other hand, social influence of friends can be incorporated in the recommendation process. For example, when considering $l_4$ as a recommendations candidate for $u_1$, the social influence of $u_4$ on $u_1$ may contribute to the decision making. Finally, the geographical influence of POIs on nearby POIs can be considered. As shown in the example, since $u_2$ has visited $l_2$ and $l_3$ before, their nearby POIs $l_1$ and $l_5$ may be considered positively due to the geographical influence.

As discussed earlier, the conventional item recommendation techniques based on user preference [150, 151, 152, 153, 147] and social influence [145, 147] seem to be applicable for POI recommendation. Nevertheless, their effectiveness on POI recommendations in LBSNs have not been studied. Most importantly, the idea of incorporating the geographical influence between POIs, which is refreshing and promising for POI recommendation, has not been investigated previously. In this paper, we examine the “geographical clustering phenomenon” of user check-in activities in LBSNs and propose a power-law probabilistic model to capture geographical influence among POIs. Accordingly, we realize the targeted collaborative POI recommendation service for LBSNs by incorporating the geographical influence of POIs via Bayesian theory. Finally, we propose a unified location recommendation framework to fuse user preference to POIs along with the social influence among users and the geographical influence among POIs.

In the following, we provide some background on conventional recommendation techniques according to user’s own preference and social influence and review related works in the literature; then we describe the location recommendation process according to geographical influence. Furthermore, we propose a location recommendation framework, which unifies all three factors together. We perform an empirical study on the different location recommendation algorithms upon two large scale datasets crawled from Foursquare and Whrrl, respectively. Finally, we summarize this work.
5.2 Preliminaries

In this section, we first provide background on user-based collaborative filtering and friend-based recommendation, which serve as the building blocks in our fusion approach to exploit user preference and social influence. Next we review some relevant studies in recommender systems.

5.2.1 User-based Collaborative Filtering

Based on collaborative filtering, users’ implicit preference can be discovered by aggregating the behaviors of similar users. Let $U$ and $L$ denote the user set and the POI set in an LBSN, which keeps track of check-in activities in the system. The check-in activity a user $u_i \in U$ has at a POI $l_j \in L$ is denoted as $c_{i,j}$ where $c_{i,j} = 1$ represents $u_i$ has a check-in at $l_j$ before and $c_{i,j} = 0$ means there is no record of $u_i$ visiting $l_j$. These recorded user check-in activities at POIs are thus used to discover a user’s implicit preference of a POI, which can be represented as a probability to predict how likely the user would like to have a check-in at an unvisited POI. We denote this prediction by $\hat{c}_{i,j}$ and obtain this predicted check-in probability of $u_i$ to $l_j$ as follows.

$$\hat{c}_{i,j} = \frac{\sum_{u_k} w_{i,k} \cdot c_{k,j}}{\sum_{u_k} w_{i,k}} \quad (5.1)$$

where $w_{i,k}$ is the similarity weight between users $u_i$ and $u_k$.

To compute the similarity weights $w_{i,k}$ between users $u_i$ and $u_k$, several similarity measures can be adopted, e.g., cosine similarity and Pearson correlation. In our study, we choose cosine similarity due to its simplicity. The cosine similarity weight between users $u_i$ and $u_k$, denoted as $w_{i,k}^{UI}$, is defined as follows.

$$w_{i,k} = \frac{\sum_{l \in L} c_{i,l}c_{k,l}}{\sqrt{\sum_{l \in L} c_{i,l}^2} \sqrt{\sum_{l \in L} c_{k,l}^2}} \quad (5.2)$$

5.2.2 Friend-based Collaborative Filtering

Friends tend to have similar behavior because they are friends and might share a lot of common interests, thus leading to correlated check-in behaviors [153, 147]. For example, two friends may hang out to see a movie together sometimes, or a user may go to a restaurant highly recommended by her friends. All those possible reasons suggest that friends might provide
good recommendation for a given user due to their potential correlated check-in behavior. In other words, we can turn to user’s friends for recommendation, and we call it recommendation based social influence from friends.

POI recommendations based on social influence can be realized by the friend-based collaborative filtering approach as described in [147].

$$\widehat{c}_{i,j} = \frac{\sum_{u_k \in F_i} S_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} S_{k,i}}$$

(5.3)

where $\widehat{c}_{i,j}$ is the predicted check-in probability of $u_i$ at $l_j$, $F_i$ is the friends set of $u_i$, and $S_{k,i}$ is directional social influence weight $u_k$ has on $u_i$ [152, 153, 147].

On the one hand, we think friends who have closer social tie may have better trust in terms of their recommendation; on the other hand, friends who show more similar check-in behavior should have more similar tastes with the active user, thus suggestions from those friends are more worthy. Thus, in the following, we introduce how to derive the social influence weight by combining the above two aspects.

One way to derive the social influence weight between two friends is based on both of their social connections and similarity of their check-in activities [145].

$$S_{k,i} = \eta \cdot \frac{|F_k \cap F_i|}{|F_k \cup F_i|} + (1 - \eta) \cdot \frac{|L_k \cap L_i|}{|L_k \cup L_i|}$$

(5.4)

where $\eta$ is a tuning parameter ranging within $[0, 1]$, and $F_k$ and $L_k$ denote the friend set and POI set of user $u_k$, respectively.\(^6\)

Another way of measurement is via the Random Walk with Restart (RWR) technique [128] over the graph that captures both the social connections among users as well as the check-in activities between users and POIs [148]. Starting from a node $k$, an RWR is performed by randomly following a link to another node at each step. Notice that there is a probability $a$ in every step to restart at node $k$. By iterating RWR repeatedly until the whole process converges, a stationary (or steady-state) probability for each node can be obtained. The stationary probabilities of nodes give us a long-term visit rate for each user node (e.g., user $u_i$) given a bias towards a particular starting node (e.g. user $u_k$). This can be interpreted as the social influence weight user $u_k$ have on $u_i$, i.e., $S_{k,i}$.

\(^6\)The friend set of a user refers to the socially connected friends of the user in the LBSN, while her POI set refers to the set of POIs she has check-in activities.
5.2.3 Related Work

Content-based and collaborative filtering techniques are two widely adopted approaches for recommender systems [154]. A content-based system selects items for recommendation based on the similarity between item content (e.g., keywords/tags describing the items) and user profile [155, 156, 157]. Since it mainly relies on dictionary-bound relations between the terms used in user profiles and item content, implicit associations between users are not considered.

The collaborative filtering systems are divided into two categories, i.e., memory-based and model-based. Memory-based systems can be further classified into user-based and item-based systems. For user-based systems [150], the similarity between all pairs of users is computed based on their ratings on associated items using some selected similarity measurement such as cosine similarity or Pearson correlation. Based on the user similarity, missing rating corresponding to a given user-item pair can be derived by computing a weighted combination of the ratings upon the same item from similar users. For item-based systems [151], instead of using similarity between users to predict missing rating, predictions are made by finding similarly rated items first in order to compute a weighted combination of user ratings upon similar items. On the other hand, the model-based collaborative filtering systems assume that users may form clusters based on their similar behavior in rating items. A model can be learned based on patterns recognized in the rating behaviors of users using machine learning techniques such as clustering algorithms or Bayesian networks [158, 159].

Under the context of social networking systems, social friendship is shown to be beneficial for collaborative filtering based recommendation systems, e.g., memory-based [144, 145] and random walk based [146, 144, 145]. These works argue that social friends tend to share common interests and thus their relationships should be considered in the process of collaborative filtering. Random walk captures a social network as a graph with probabilistic weighted links to represent social relations and thus is able to accurately predict user preferences to items [145] and social influence to other users [148]. On the other hand, social friendship has also been explored in the model-based systems [152, 147]. These work mostly focus on conventional recommendation systems for recommending items such as movies.

Recently, location recommendation and mining has attracted a lot of attentions from the research community [94, 95, 49, 93]. Among them, [94, 95, 49] are mainly focused on GPS datasets which do not consider social relationships among users. In these works, unfortunately, the geographical influence among POIs are not explored [94, 95]. Recently, the correlation of locations in GPS trajectories are explored [49]. In this work, however, locations are still treated
as conventional items. As such, the correlations between locations are established through users’ activities instead of their geographical influence. [93] is the first research to provide location recommendations services in LBSNs, but with the goal of improving efficiency of location recommendation.

Our study differentiates itself from all these prior works in four aspects: i) the application domain of location-based social networking systems, embracing both social and geographical features in the captured data, is new and unique; ii) the study of social influence and geographical influence in recommender systems for LBSNs is unexplored previously; iii) the proposal of unified collaborative recommendation approach, which incorporates geographical influence along with user preference and social influence, is new and innovative; iv) two large-scale real dataset collected from well known LBSNs, namely, Foursquare and Whrrl datasets, are adopted for performance evaluation.

5.3 Recommendation via Geographical Influence

As mentioned earlier, the check-in activities of users in LBSNs record their physical interactions (i.e., visits) at POIs, Thus, we argue that the geographical proximities of POIs have a significant influence on users’ check-in behavior. To better understand this geographical influence on users, we perform a spatial analysis on real datasets of user check-in activities collected from two well known LBSNs, i.e., Foursquare and Whrrl. Specifically, we aim to study the implication of distance on user check-in behavior by measuring how likely two of a user’s check-in POIs are within a given distance. To obtain this measurement, we calculate the distances between all pairs of POIs that a user has checked in and plot a histogram (actually probability density function) over the distance of POIs checked in by the same user. As shown in Figure 5.2, a significant percentage of POIs pairs checked in by the same user appears to be within short distance, indicating a geographical clustering phenomenon in user check-in activities.\footnote{Note that the figure has been shown in log-log scale.}

This phenomenon may be attributed to the geographical influence which may be intuitively explained by the following tendencies: (1) people tend to visit POIs close to their homes or offices; and (2) people may be interested in exploring nearby POIs of a POI that they are in favor of, even if it is far away from home. As a result, the POIs visited by the same user tend to be clustered geographically. We believe that this geographical clustering phenomenon in user check-in activities can be exploited for POI recommendations in LBSNs. Thus, in the
following, we study and model this geographical influence on user check-in behavior at POIs, aiming to utilize it in POI recommendations.

To achieve our goal, we would like to compute the likelihood that a user $u_i$ would check in both POI $l_j$ and $l_k$. Based on Figure 5.2, we intuitively think the check-in probability may follow the power-law distribution. Nevertheless, we observe that the check-in probability of POI pairs visited by the same person over distance is not a standard power-law distribution. Even though the left part of the figure decreases linearly (i.e., decreases exponentially in regular scale) and thus fits power-law distribution very well, the right part may sometimes deviate irregularly (i.e., the probability is high at some points). A reasonable explanation is that users may travel to different places and thus create multiple check-in spatial clusters. Generally speaking, the fact that a user’s check-in POIs tend to be in a short distance is confirmed in our data analysis. As mentioned earlier, nearby POIs are more related to each other, which exhibits strong geographical influence. Moreover, the linear portion of the plot in Figure 5.2 covers the majority (90%) of the POI pairs. Thus, we propose to use power law distribution to model the check-in probability to the distance between two POIs visited by the same user as follows.

$$y = a \times x^b$$

where $a$ and $b$ are parameters of a power-law distribution, and $x$ and $y$ refer to the distance between two POIs visited by the same user and its check-in probability, respectively.
Equation (5.5) can be transformed into Equation (5.6) in “log-log” scale to fit a linear model.

\[
\log y = w_0 + w_1 \log x
\]  

(5.6)

Thus, the original power-law distribution can be recovered via the following equation.

\[
a = 2^{w_0} \quad b = w_1
\]  

(5.7)

Hence, we can simply apply a linear curve fitting method to realize regression as follows. More specifically, let \( y' = \log y \) and \( x' = \log x \). We shall fit data as follows

\[
y'(x', w) = w_0 + w_1 \cdot x'
\]  

(5.8)

where \( w_0 \) and \( w_1 \) are the linear coefficients, collectively denoted by \( w \). In other words, the model can be learned in form of \( w \). In order to avoid over-fitting, we approach the weight coefficients by least square error method and add a penalty term (i.e., regularization term) to discourage the coefficients from reaching large values as below [114].

\[
E(w) = \frac{1}{2} \sum_{n=1}^{N} \left( y'(x'_n, w) - t_n \right)^2 + \frac{\lambda}{2} ||w||^2
\]  

(5.9)

where \( E(w) \) denotes the loss function, \( N \) presents the cardinality of input dataset, \( t_n \) is the ground truth corresponding to \( x'_n \), and \( \lambda \) is the regularization term.

Accordingly, the optimal values of \( a \) and \( b \) form the setting which minimizes the loss function \( E(w) \) as follows.

\[
\text{opt}\{a, b\} = \arg \min_{a,b} E(w)
\]  

(5.10)

In the following, we introduce a collaborative recommendation method based on the naive Bayesian method to realize POI recommendation in LBSNs. For a given user \( u_i \) and its visited POI set \( L_i \), we define the probability that \( u_i \) has check-in activities at all locations in \( L_i \) by considering the pair-wise distances of POIs in \( L_i \) as follows.

\[
\Pr[L_i] = \prod_{l_m, l_n \in L_i \land m \neq n} \Pr[d(l_m, l_n)]
\]  

(5.11)

where \( d(l_m, l_n) \) denotes the distance between POIs \( l_m \) and \( l_n \), and \( \Pr[d(l_m, l_n)] = a \times d(l_m, l_n)^b \).
which follows the pow-law distribution model we obtained above. Note that here we assume the distances of POI pairs are independent.

Thus, for a given POI \(l_j\) (i.e., the recommendation candidate), user \(u_i\), and her visited POI set \(L_i\), we have the likelihood probability for \(u_i\) to check in \(l_j\) as follows.

\[
Pr[l_j | L_i] = \frac{Pr[l_j \cup L_i]}{Pr[L_i]} = \prod_{l_y \in L_i} Pr[d(l_j, l_y)] = \prod_{l_y \in L_i} Pr[d(l_j, l_y)]
\] (5.12)

To make a POI recommendation, we sort all the POIs in \(L - L_i\) in accordance with their \(Pr[l_j | L_i]\) \((l_j \in L - L_i)\) to return the POI with the highest \(Pr[l_j | L_i]\) to the user.

### 5.4 Unified Collaborative POI Recommendation

In this section, we propose a unified framework to perform collaborative recommendation, which fuses ideas factors of user preference, social influence and geographical influence in POI recommendation. Notice that, different from predicting a POI’s rating, we aim to return a ranked list of candidate POIs, which is very similar to conventional information retrieval [160].

#### 5.4.1 Fusion Framework

As discussed, each factor, i.e., user preference, social influence or geographical influence, can be utilized to realize POI recommendation. Thus, we intuitively can implement three different recommender systems. We propose to use a linear fusion framework to integrate ranked lists provided by the three above-mentioned recommenders into the final ranked list [160, 161]. By integrating multiple recommenders, top-ranked POIs from each of the recommendation algorithms could increase both recall (due to the different highly ranked POIs) and precision (giving that the recommender systems have a high density of user-preferred POIs on top of the results lists.

Let \(S_{i,j}\) denote the check-in probability score of user \(u_i\) at POI \(l_j\), i.e., the more likely \(u_i\) has a check-in activity at \(l_j\), the larger \(S_{i,j}\) is. Let \(S_{i,j}^u\), \(S_{i,j}^s\) and \(S_{i,j}^g\) denote the check-in probability scores of user \(u_i\) at POI \(l_j\), corresponding to recommender systems based on user preference, social influence and geographical influence, respectively. We have \(S_{i,j}\) as follows.

\[
S_{i,j} = (1 - \alpha - \beta)S_{i,j}^u + \alpha S_{i,j}^s + \beta S_{i,j}^g
\] (5.13)
where the two weighting parameters $\alpha$ and $\beta$ ($0 \leq \alpha + \beta \leq 1$) denote the relative importance of social influence and geographical influence comparing to user preference. Here $\alpha = 1$ states that $S_{i,j}$ depends completely on the prediction based on social influence; $\beta = 1$ states that $S_{i,j}$ depends completely on the prediction based on geographical influence; while $\alpha = \beta = 0$ states that $S_{i,j}$ counts only on user preference.

### 5.4.2 Check-in Probability Score Estimation

According to the above fusion framework, in order to estimate the check-in probability score $S_{i,j}$, we need to predict the check-in probability score of $S_{u,i,j}$, $S_{s,i,j}$ and $S_{g,i,j}$ corresponding to user preference, social influence and geographical influence, respectively. Accordingly, we estimate the check-in probability $p_{u,i,j}$, $p_{s,i,j}$ and $p_{g,i,j}$ for a user $u_i$ to visit a POI $l_j$ in order to obtain $S_{u,i,j}$, $S_{s,i,j}$ and $S_{g,i,j}$, respectively.

First, the prediction of $p_{u,i,j}$ can be estimated based on the idea of user-based collaborative filtering as discussed before. More specifically, we utilize the behavior of similar users to realize the prediction as Equation (5.1). Thus we have

$$p_{u,i,j} = \frac{\sum_{u_k} w_{i,k} \cdot c_{k,j}}{\sum_{u_k} w_{i,k}} \quad (5.14)$$

where $w_{i,k}$ is the similarity weight between users $u_i$ and $u_k$.

Similarly, the prediction of $p_{s,i,j}$ can be estimated based on the idea of friend-based collaborative filtering. Thus, according to Equation (5.3), we have

$$p_{s,i,j} = \frac{\sum_{u_k \in F_i} SI_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} SI_{k,i}} \quad (5.15)$$

where $F_i$ is the friends set of $u_i$, $SI_{k,i}$ is the weight measuring social influence from $u_k$ to $u_i$.

Finally, $p_{g,i,j}$ can be directly obtained from Equation (5.12)

$$p_{g,i,j} = Pr[l_j|L_i] = \prod_{l_y \in L_i} Pr[d(l_j, l_y)] \quad (5.16)$$

where $L_i$ is the visited POI set of $u_i$, and $d(l_j, l_y)$ denotes the distance between POIs $l_j$ and $l_y$.

After we get the check-in probability estimation, we obtain the corresponding scores as follows.
\[ S_{ij}^u = \frac{p_{ij}^u}{Z_i^u}, \text{ where } Z_i^u = \max_{j \in L_i} \{ p_{ij}^u \} \]
\[ S_{ij}^s = \frac{p_{ij}^s}{Z_i^s}, \text{ where } Z_i^s = \max_{j \in L_i} \{ p_{ij}^s \} \]
\[ S_{ij}^g = \frac{p_{ij}^g}{Z_i^g}, \text{ where } Z_i^g = \max_{j \in L_i} \{ p_{ij}^g \} \]

where \( Z_i^u \), \( Z_i^s \) and \( Z_i^g \) are normalization terms.

### 5.5 Empirical Evaluation

In this section, we design and conduct several experiments to compare the recommendation qualities of the proposed collaborative recommendation algorithms with some state-of-the-art recommendation techniques, including collaborative filtering and random walk with restart, and to investigate several interesting questions. Specifically, the design of the experiments aims to achieve the following goals. (1) As our proposed method factors in user preference, social influence from friends and geographical influence from nearby location, we intent to study parameters \( \alpha \) and \( \beta \) to understand the roles/weights of the above-mentioned factors in obtaining optimal recommendations. (2) We intend to validate our ideas by comparing the effectiveness of the proposed approach with other state-of-the-art techniques. (3) Due to the growing research interests in social influence from friends, we intend to further study the similarity of check-in behaviors in terms of the strength of “social ties” between two friends. (4) In our proposal, user-based collaborative filtering approach has been employed to discover user preference. We intend to explore the feasibility and necessity of integrating item-based collaborative filtering approach to further enhance the recommendation quality. (5) We would like to understand how data sparsity may affect POI recommendations in LBSNs. (6) How well our techniques deal with cold start users, who do not have many check-in records for discovery of their interests [162].

#### 5.5.1 Dataset Description

We crawled the websites of Foursquare and Whrrl, two of the most representative LBSNs, for a month to collect two datasets consisting of 153,577 users and 96,229 POIs in Foursquare, and 5,892 users and 53,432 POIs in Whrrl, respectively. Our performance evaluation is conducted
on these two large-scale real datasets. After summarizing the check-in records, we get the user-POI check-in matrix densities as $4.24 \times 10^{-5}$ for Foursquare dataset and $2.72 \times 10^{-4}$ for Whrrl datasets, respectively. Note that, the effectiveness of recommendation service for sparse dataset (i.e., low density user-POI check-in matrix) is usually not high due to the limited information provided by the dataset. For example, the reported precision in [145] is 0.17 over a pre-prossed dataset with $7.8 \times 10^{-4}$ density. Thus, in our experiments, we focus on observing the relative performance of algorithms instead of their absolute effectiveness measures, which we expect to improve as the number of LBSN users continues to grow and more check-in activities are logged. To facilitate our evaluation, for each individual user in the datasets, we randomly mark off $x\%$ ($x = 10, 30, 50$ (with 30 as the default value) of all POIs visited by the user. In the experiments, the evaluated POI recommendation algorithms are used to recover the missing user-POI pairs that have been marked off.
5.5.2 Performance Metrics

A POI recommendation algorithm under evaluation computes a ranking score for each candidate POI (i.e., POI that user has not visited) and returns the top- \( N \) highest ranked POIs as recommendations to a targeted user. To evaluate the prediction accuracy, we are interested in finding out how many POIs previously marked off in the preprocessing step recovered in the returned POI recommendations. More specifically, we examine two metrics: (1) the ratio of recovered POIs to the \( N \) recommended POIs, and (2) the ratio of recovered POIs to the set of POIs deleted in preprocessing. The former is \( \text{precision}@N \) while the latter is \( \text{recall}@N \), and collectively referred as \( \text{performance}@N \). In our experiment, we test the performance when \( N = 5, 10, 20 \) with 5 as the default value.
5.5.3 Evaluated Recommendation Approaches

Three factors, namely user preference (U), social influence from friends (S) and geographical influence from POIs (G), are incorporated in our unified collaborative recommendation algorithm, denoted by USG in our evaluation. A number of state-of-the-art and new collaborative filtering approaches, some of which can be configured by controlling the weight parameters, $0 < \alpha, \beta < 1$, in USG, are also evaluated for comparison. In addition of USG, the recommendation approaches under evaluation are listed below.

- **user-based CF (denoted by U)** - this is a special case of USG by setting both $\alpha$ and $\beta$ as zeros. In other words, only user preference is considered for recommendation.

- **friend-based CF (denoted by S)** - this is also a special case of USG, where $\alpha = 1$. Here, only friends of the targeted user are used in making a specific recommendation. As introduced before, there are two alternative methods to derive the social influence weight between friends. One is to compute the social influence weight based on friends based on Equation( 5.4) [145] and the other is to derive social influence weight between friends using Random Walk and Restart technique [148]. To differentiate these two approaches, we denote them as $S$ and $S_{rwr}$, respectively.

- **GI-based recommendation (denoted by G)** - this approach, considering only the factor of geographical influence, is a special case of USG where $\beta = 1$.

- **Random Walk with Restart (denoted by RWR)** - this is a state-of-the-art algorithm recently developed for collaborative item recommendation based on social networks [145]. Users’ preferences to items are predicted by Random Walk and Restart over a graph capturing social graph and user-item matrix.

- **User preference/social influence based recommendation (denoted by US)** - this method, considering both user preference and social influence from friends, is a special case of USG, where $0 < \alpha < 1$ and $\beta = 0$.

- **User preference/geographical influence based recommendation (denoted by UG)** - this approach, considering both user preference and geographical influence, is a special case of USG, where $0 < \beta < 1$ and $\alpha = 0$. 
5.5.4 Tuning Parameters

As mentioned, two parameters $\alpha$ (for social influence factor) and $\beta$ (for geographical influence factor) can be controlled to tune the performance of USG and to configure it into other recommendation approaches for evaluation. Here we vary them in USG to understand the roles of user preference, social influence from friends and geographical influence from POIs played in achieving the optimal USG performance. Similarly, we tune $\alpha$ in US and $\beta$ in UG to find-out their optimal settings as well. Figure 5.3 shows the performance@5 results of USG under different $\alpha$ and $\beta$ settings, where the best parameter settings are indicated in the figures. The optimal settings for US and UG can also be observed in the figures, i.e., dashed line for US and solid line for UG. Those optimal parameter settings are also summarized in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>Precision@5</th>
<th>Recall@5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>Foursquare</td>
<td>US</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>UG</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>USG</td>
<td>0.1</td>
</tr>
<tr>
<td>Whrrl</td>
<td>US</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>UG</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>USG</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 5.1. Optimal parameter settings

Through this study, we can easily observe that user preference plays a dominate role in contributing to the optimal recommendation, while both social influence and geographical influence are innegligible. More specifically, as shown in Table 5.1, the factor of user preference contributes at least 70% in making the best recommendation, while both social influence and geographical influence contribute at least 10%.

5.5.5 Performance Comparison

Next, we compare the effectiveness of the recommendation approaches under evaluation. Figure 5.4 shows the performance@$N$ ($N = 5, 10, 20$) of all approaches in terms of their best performance (i.e., the performance under the optimal parameter settings). The experiments used both Foursquare and Whrrl datasets. The precision and recall for them are plotted in Figure 5.4(a) and Figure 5.4(b), and Figure 5.4(c) and Figure 5.4(d), respectively. In these figures, USG always exhibits the best performance in terms of precision and recall under all values of
Ns, showing the strength of combines all three factors of user preference, social influence and geographical influence. Notice that both of our real datasets (i.e., Foursquare and Whrrl) have low density. According to the empirical study in [145], the reported precision is about 0.17 over a pre-processed dataset with \(7.8 \times 10^{-4}\) density of user-item matrix. Thus, the measured low precision over our datasets (which are not preprocessed) is reasonable. Most importantly, USG outperforms the baseline approach \(U\) (i.e., user-based CF) by about 50% percentage of performance improvement in both datasets.

Between the two alternative social influence measurement methods (i.e., \(S\) and \(S_{rwr}\)) for friend-based CF, we find \(S\) to have much better performance than \(S_{rwr}\). Moreover, RWR shows poor performance for POI recommendation in these experiments. This raises a very interesting question of whether Random Walk and Restart technique is suitable for POI recommendations.
In a later section, we shall answer this question by analyzing the correlation between (i) the similarity of check-in behavior among friends and (ii) social ties among friends. For the rest of the experiments, we use $S$ as the component of social influence from friends in $\text{US}$ and $\text{USG}$.

Figure 5.4 also indicates that both social influence and geographical influence can be utilized to perform POI recommendation. As shown, both $S$ and $G$ provide comparable results against $U$. Notice that, in LBSNs, since the check-in activities involve physical interaction between users and POIs, geographical influence matters a lot, which is confirmed in the study. As shown, $G$ usually outperforms $S$ and sometimes even performs better than $U$, e.g., when $N = 20$. Also, $\text{UG}$ always show better performance than $\text{US}$. This is due to the spatial clustering phenomenon appearing in user check-in activities. Thus, when $N$ is relatively large, there is very good chance to discover most of user’s check-in activities based on social influence.

In both Foursquare and Whrrl datasets, we find that when more factors are considered the performance turns out to be better. For example, $\text{US}$ is better than $U$ and $S$, $\text{UG}$ is better than $U$ and $G$, and $\text{USG}$ shows the best performance.
5.5.6 Study on Item-based CF

In addition to user-based CF, item-based CF can also estimate a user’s preference to an item, by exploring the similarity between items instead of users [151]. In [161], a CF technique has been proposed to fuse both user-based and item-based similarity to overcome the data sparsity problem [162]. Thus, a potential idea for POI recommendations is to employ the item-based CF (denoted by $L$). Additionally, geographical influence, which models the influence among POIs, may be seemingly similar to “item similarity” in item-based CF. However, we would like to point out that they are conceptually different and thus should not be mistaken. In this section, we explore the idea of further incorporating $L$ into our framework by examining whether fusing $L$ with $U$ and $G$ respectively into new approaches denoted by $UL$ and $GL$ would outperform $U$ and $G$ alone.

Similar to [161], we introduce a weighting parameter $\lambda$ in $UL$. When $\lambda = 1$, $UL$ is reduced to $U$; and when $\lambda = 0$, $L$ is obtained. Similarly, we introduce a weighting parameter $\gamma$ in $GL$. Figure 5.5 and Figure 5.6 show the performance of $UL$ and $GL$ on Foursquare and Whrrl datasets under various settings of $\lambda$ and $\gamma$. Surprisingly, these figures show that $L$ brings no advantage at all in enhancing $U$ or $L$ in POI recommendations, indicating item-based CF is not an effective approach in our application. Our explanation is that, at the current stage, POIs in LBSNs may not have been visited by sufficient many users to make item-based CF work well. In other words, the computed similarity between two POIs may not provide a good clue to decide whether a user likes a POI or not. Since $U$ or $G$ alone show much better performance than $L$, we don’t integrate $L$ in our recommendation framework.

5.5.7 Study on Social Influence

As shown earlier, Random Walk with Restart [145] does not perform well for POI recommendations. To obtain a comprehensive understanding of the reasons behind, we analyze the correlation between the similarity of user check-in behaviors and the user similarity calculated based on Random Walk and Restart. Note that, based on [145], user similarity can be derived from the social graph matrix and user-POI check-in matrix. Figure 5.7(a) and Figure 5.7(d) show the plots on Foursquare and Whrrl datasets under the best RWR settings. Both figures show that similar users do not necessarily have high similarity in their check-in behaviors. For example, user pairs with similarity larger than 0.1 usually share nothing in their check-in behavior in both Foursquare and Whrrl datasets. The results indicate that the tastes of a user’s
friends may actually vary significantly, which has also be discussed in [163] recently. To further verify this finding, we also examine the correlation between the similarity of check-in behaviors between two friends and the strength of their social ties. In our tests, the social tie is defined in two forms: 1) number of common friends (see Figure 5.7(b) and Figure 5.7(e) for experimental results) and common friend ratio (see Figure 5.7(c) and Figure 5.7(f) for experimental results), where common friend ratio is measured by Jaccard coefficient. For friends who have very strong social tie (i.e., larger number of common friends or larger common friend ratio), we again find their check-in behaviors are not necessarily similar as shown in the figures.

From the above observations, we conclude that friends have different tastes. The similarity in friends’ check-in behaviors may not necessarily be reflected in terms of the strength of their social ties. As a matter of fact, in measuring the social influence between friends, we find the factor of check-in behavior to be more important than the factor of social tie. Through
5.5.8 Impact of Data Sparsity

Here, we study how USG deals with the data sparsity problem. In order to produce user-POI check-in matrix with different sparsity, we mark off $x\% = 10\%, 30\%$ and $50\%$ of user’s check-in activity records from the original check-in datasets for three groups of tests as shown in Figure 5.8. The larger the mark-off ratio $x$ is, the sparser the user-POI check-in matrix is. As shown, USG always exhibits the best performance@5 under all mark-off ratios. Particularly, when the data is very sparse, e.g., $x\% = 50\%$, geographical influence plays an extremely im-
portant role in recommending interesting POIs to users. The reason is that both users and their social friends have relatively small check-in logs. Thus, the similarity weight or social influence score derived from such sparse data may be misleading. On the other hand, geographical influence, reflecting a global behavior affected by geography, fits the behaviors of most users in LBSNs. Thus, the approaches incorporating geographical influence factor, i.e., $G$, $UG$ and $USG$, show great strengths under various data sparsity scenarios.

5.5.9 Test for Cold Start Users

Finally, we test the performance of POI recommendations for cold start users. Here, we consider those users who have less than 5 check-in activities in the user-POI check-in matrix after removing 30% check-ins as cold start users. As shown in Figure 5.9, in all cases we tested, $USG$ always shows the best performance. Note that in POI recommendations for cold start users, user preference is hard to capture as POIs visited by this user are few. Consequently, $U$ shows the worst performance as it only considers user preference. $G$, which explores the spatial clusters of user check-in activities, is also affected. On the other hand, $S$ overcomes the lack of user’s check-ins as social friends may supply many useful check-ins, potentially useful for POI recommendations. Thus, in this experiment, we find that the recommendation performance of $S$ usually works better than $U$ and $G$ do. Notice that, we find the performance of $G$ to be better than $S$ in extremely sparse scenario in Figure 5.8 because in that scenario, social friends’ check-in records are very limited as well. Thus, geographical influence prevails due to its applicability to most of the people. However, it is noteworthy that all three factors are very important for the POI recommendations to cold start users, as $USG$ is always the best.

5.6 Summaries

This research attempts to facilitate a POI recommendation service in location-based social networks. Our idea is to incorporate user preference, social influence and geographical influence in the recommendation. In addition to deriving user preference by user-based collaborative filtering and capturing social influence from friends, we model the geographical influence among POIs by employing power law distribution to uncover the spatial clustering phenomenon in user check-in activities. Furthermore, we propose a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence. We
conduct a comprehensive performance evaluation over two large-scale real datasets collected from Foursquare and Whrrl. Experimental results show that the unified collaborative recommendation technique is superior to all other recommendation approaches evaluated. Additional findings have been uncovered through analysis of the experimental results, including 1) geographical influence shows a more significant impact on the effectiveness of POI recommendations than social influence; 2) Random Walk and Restart may not be suitable for POI recommendation in LBSNs, because friends exhibit significantly different preferences (i.e., the strength of social ties do not reflect the similarity of check-in behavior among users in LBSNs; 3) Item-base CF is not an effective approach in our application due to insufficient number of visitors to many locations at the current state of LBSNs.
Exploring Social Influence for Recommendation - A Generative Model Approach

Social friendship has been shown beneficial for item recommendation for years. However, existing approaches mostly incorporate social friendship into recommender systems by heuristics. In this chapter, we argue that social influence between friends can be captured quantitatively and propose a probabilistic generative model, called social influenced selection (SIS), to model the decision making of item selection (e.g., what book to buy or where to dine). Based on SIS, we mine the social influence between linked friends and the personal preferences of users through statistical inference. To address the challenges arising from multiple layers of hidden factors in SIS, we develop a new parameter learning algorithm based on expectation maximization (EM). Moreover, we show that the mined social influence and user preferences are valuable for group recommendation and viral marketing. Finally, we conduct a comprehensive performance evaluation using real datasets crawled from last.fm and whrrl.com to validate our proposal. Experimental results show that social influence captured based on our SIS model is effective for enhancing both item recommendation and group recommendation, essential for viral marketing, and useful for various user analysis.
6.1 Overview

In this chapter, we take the view of “social influence” from friends to explain the decision making of item selection by a user. Social influence here refers to the phenomenon that a user adopts a suggestion from friends, which may or may not deviate from her own preferences.\(^1\) In our daily life, besides of our own preference, we usually turn to our friends for opinions of books, movies or restaurants. Obviously, friends play a role in our decision making of many daily activities and events and thus impose some influence on us. Nevertheless, the influences from different friends are not equal. We do not take friends’ opinions/suggestions purely based on our “trust” or “similarity” in preferences. Moreover, some friends with different interests and expertise may be very influential to us, while some other friends with very similar interests may not contribute that much in our decision making. Therefore, the notion of social influence discussed here is fundamentally different from the social correlation, trust intensity, or the various measures of friend similarity based on different heuristics.

To systematically exploit social influences from friends for item recommendation, we aim to quantitatively capture the social influence to a user from each of her friends by leveraging information embedded in the user social network, user behavior and item content. To meet our goal, we propose to adopt the \textit{probabilistic generative model} as a methodology to model the decision making of item selection, e.g., deciding which restaurant to dine. We attribute an item selection decision by a user \textit{probabilistically} to her own preference or the preferences of her friends (due to social influence). Assuming a set of latent topics existing in the targeted application domain\(^2\), we propose the \textit{social influenced selection (SIS)} model, which captures (1) the distribution of social influence from friends for each user, (2) the distribution of personal preference over the latent topics for each user, (3) the distribution of generated items for each topic, (4) the distribution of generated content for each topic; and thus enables a seamless integration of social influence, user behavior and item content for item recommendation.

Based on SIS, we aim to mine the model parameters listed above from the social network of users and the access transactions of users on items (as well as their content) through statistical inference based on \textit{expectation maximization} (EM). In a typical EM algorithm, the E-step is

---

\(^1\)Note that social influence is a very general term related to many different phenomena. For example, Herbert Kelman identified three broad varieties of social influence including compliance, identification and internalization [164]. In this paper, we consider the social influence quantitatively as the probability that a user adopts a suggestion from friends.

\(^2\)The term \textit{topic}, from topic models, represents a genre of items in this paper. Take movies as an example of the items, a topic could be action, thriller, romantic or even a latent genre that cannot be expressed literally.
used for computing the posterior of a latent variable. The multiple layers of variables we have in SIS bring new challenges. For example, the latent variables and other associated factors need to be processed together in a very large joint-variable space, which mandates development of a new and efficient learning algorithm. Assuming that the social influence exists only between friends instead of arbitrary users, we address the scalability issue by devising a new model learning algorithm which handles two layers of latent variables at the same time. In addition, to explore the flexibility and applicability of SIS, we explore the social influence and user preferences mined based on SIS for group recommendation, viral marketing and various user analysis.

The remainder of this chapter is organized as follows. Section 6.2 summarizes the related work. Section 6.3 introduces our SIS model which seamlessly captures social influence along with user behavior and item content. Section 6.4.1 proposes a new group recommendation method using the social influence obtained and discusses its potential application for viral marketing. Section 6.5 reports the experimental results and findings in our performance evaluation. Finally, Section 6.6 summarize the chapter.

### 6.2 Related Work

In this section, we review some related works, including recommender systems, recommendation techniques enriched by social networks, and group recommendation.

Recommendation systems have attracted a lot of attention in the past decade and have been successfully deployed in many e-commerce websites, such as Amazon and Netflix. Two widely adopted approaches for recommendation systems are collaborative filtering (CF) and content-based filtering [154]. Collaborative filtering [165, 166] recommends items for a given user by referencing item ratings from other similar users, while content-based filtering techniques [167] make recommendations by matching a user’s personal interests (or profiles) with item content (e.g., item description or tags).

Recent development of social media and social networking systems brings new opportunities for recommender systems. Several research works find that there is a correlation between items selected by a user and those selected by her friends (i.e., friends share some common interests) and propose to exploit the correlation for item recommendations [144, 145, 147, 152, 163, 168]. The basic idea behind these techniques is that, since a user’s friends may share common interests, recommendations can be made to the user by considering interests of her friends.
For example, [26] makes recommendations by assigning weights to friends based on the similarity of item selection behavior (in terms of the number of commonly selected items) and similarity of social friendship (in terms of the number of their common friends). In addition, [144, 145] employ the random walk approach [128] to incorporate user’s social relationship in item recommendation. This approach fundamentally assumes that friends exhibit similar behavior and thus aims to incorporate the friendship into the random walking process. However, due to the lack of quantified strength in friendship, equal weights have been assigned to social links from a user to her friends. This uniform weight assignment obviously does not truthfully capture the role of friendship played in the item selection process and thus is not expected to be effective for recommendations.

Some prior works avoid the aforementioned problem by integrating users’ social trust network into their recommendation techniques. Assuming that trust intensities among users and their friends are handily available and that users and friends exhibit similar behaviors, the trust intensities among friends are used as the basis to assign more similarity weight to trusted friends when making recommendations [147, 152]. In [163], the authors learn a user’s preference from the aggregated preferences of her friends. More specifically, they add additional regularization term in the matrix factorization approach to impose the constraints that user's latent topic vector should be similar to the average or weighted average of her friends’ latent topic vectors. Recently, [168] explores a notion of social correlation to capture the correlation/similarity in the item selection behavior of social friends, i.e., two friends with very similar item selection behavior have very high social correlation. The authors adopt latent factor model to learn the social correlation among friends from their social network and a user-item accessing matrix. Experimental result shows that social correlation is able to improve recommendation performance.

All of the above mentioned research works obtained varied performance improvement over conventional item recommendation techniques, demonstrating the advantages of take into account the social friendship in item recommendation. It’s not surprising that recommendation techniques that exploit the similar item selection behavior of friends have achieved a certain degree of success, as several research studies on social network analysis have indicated that a user’s behavior indeed often correlates to the behavior of her friends [?, ?]. The question, nevertheless, is whether the success can be truly attributed to the ideas behind those aforementioned techniques? We argue that, while the general idea of incorporating the item selection behavior of friends is in the right direction, the heuristics adopted in these existing works do
not exactly capture the core factors and thus cannot be fully accounted for the claimed success.

We also explore the potential use of social influence for group recommendation. Here we review a number of group recommendation techniques. Group recommendations have been designed for various domains such as web/news pages [169], tourism [170], music [171, 172], and TV programs and movies [173, 174]. In summary, two main approaches have been proposed for group recommendation [175]. The first one creates an aggregated profile for a group based on individual profiles of its group members and makes recommendations based on the aggregated group profile [171, 174]. The second approach aggregates the recommendation results for individual members into a single group recommendation list. In other words, recommendations (i.e., ranked item lists) for individual members are created independently and then aggregated into a joint group recommendation list [176], where the aggregation functions could be based on average or least misery strategies [177]. Different from these prior works, our approach exploits the preferences of group members and their influence on each other to reach the final decision. Evaluation from real datasets demonstrates a significant improvement from the our proposal of using social influence alone over the state-of-the-art methods.

### 6.3 Social Influenced Selection

In this section, we propose a probabilistic generative model to capture the various factors contributing to the decision making of item selection. Motivated by reported empirical evidence of influence propagation via on-line social links [144, 145], one important factor we argued for, along with user preference, is the social influence. Instead of trying to capture exactly how social influence takes effect in the decision making process, we assume the existence of social influence and aim to capture quantified social influence among friends for use in recommender systems. In the following, we first introduce our social influenced selection (SIS) model and then present our approach to statistically inference a number of model parameters valuable for recommendation services and other applications.

---

3How other people affect one’s beliefs, feelings and behaviors (i.e., the study of social influence) is the focus of social psychology research in the past 50 years [164]
6.3.1 The SIS Model

In addition to personal preference of certain items and their specific content features, we believe that social influence from friends also contribute to one’s item selection decision. For example, Alice went for a movie “Kung Fu Panda” because she likes funny cartoon animations and Bob, a fan of Chinese martial art stories, had recommended it to her. In this example, we attribute the decision of watching Kung Fu Panda over other choices to Alice’s own preference and Bob’s (as well as other friends’) suggestions. Notice that several factors are involved here, including (i) Alice’s tastes of movies as well as her friend’s possible preferences; (ii) Alice’s independence in decision making; and (iii) influences from Alice’s friends. We argue that the decision is made implicitly in accordance with the collective preferences of Alice’s and her friends’. Thus, the decision making is similar to drawing a preference from the above-mentioned collection based on Alice’s independence in decision making and her friend’s influence. Consequently, a preference common to Alice and her friends has a higher probability to be drawn than an uncommon preference from some friend (which may have a lower probability but still possible, as evident by abnormal selection behavior sometimes made by people).

Figure 6.1. The SIS model

In the proposed SIS model, as shown in Figure 6.1, we follow the ideas of [178, 179] to represent user preferences in terms of a number of latent topics and to correlate the items
(and their content) with users through these latent topics. Let \( U = \{u_1, u_2, \cdots, u_N\} \) and \( I = \{i_1, i_2, \cdots, i_M\} \) be the user set and item set, respectively. Each item \( i \in I \) is associated with a set of tags/words \( W_i = \{w_1, w_2, \cdots\} \). A latent topic set \( Z = \{z_1, z_2, \cdots, z_K\} \) is employed to capture the latent user interests (i.e., preferences) and to characterize the items and their content. The SIS model assumes that a decision made by a user \( u \in U \) to select an item \( i \in I \) (associated with a tag \( w \in W_i \)) is attributed to a preference (i.e., one of the latent topics \( z \in Z \)) due to the user’s independent choice and the social influence from friends. For instance, in the example discussed earlier, Alice’s decision to see “Kung Fu Panda” may be attributed to funny cartoon animation (her own preference) or Chinese martial art (Bob’s preference), both of which may be considered as the latent topics in our model. Obviously, our goal (and the novelty of SIS) lies in qualitatively capturing the social influence, which along with other model parameters of SIS such as user preference, is shown later to be valuable for recommendations and various applications. Thus, we introduce another variable \( f \) to represent the friends in SIS.

Let \( F(u) \subseteq U \) denote the friend list of a user \( u \). The social influence to \( u \) from one of her friends \((f \in F(u))\) is depicted as how likely \( f \)’s personal preference has contributed to the item selection decision. As discussed earlier, the probability for a friend of \( u \)’s to affect the item selection is in proportional to the social influence of this friend. Notice that \( u \)’s independence in decision making and her personal preference play a similar but even more important role as the social influence and friends’ preferences. In SIS, for simplicity, we assume \( u \) is a special friend of herself (i.e., \( u \in F(u) \)). Therefore, the SIS model, as illustrated in Figure 6.1, are depicted as follows. A decision of item selection made by user \( u \) is probabilistically determined based on the preference of a friend (including herself) \( f \in F(u) \). If \( f = u \), the selection is attributed to \( u \)’s own preference; otherwise \((f \neq u)\), the decision is influenced by \( f \) at this time (i.e., the selection follows \( f \)’s interests rather than \( u \)’s own tastes). As shown in the figure, we define a parameter social influence distribution \( \Pr(f|u) \) as the probability for \( u \) to be influenced by a friend \( f \). Once \( f \) is picked based on \( \Pr(f|u) \), we randomly draw a topic \( z \) from \( f \)’s preference based on probability \( \Pr(z|f) \). Then, the topic \( z \) in turn generates an item \( i \) and a tag \( w \) according to the topic’s item distribution \((\Pr(i|z))\) and topic’s content distribution \((\Pr(w|z))\), respectively. Note that the selection behavior of user \( u \) on both item \( i \) and its content \( w \in W_i \) is probabilistically captured in a coherent fashion by the generative process of SIS. Thus, SIS is able to comprehensively incorporates various information, including social influence, user behavior and item content, in the model.
6.3.2 Parameter Learning Algorithm

The problem of recommending new (previously unaccessed) items to a user \( u \) can be addressed by estimating the probability for \( u \) to select an item \( i \) (i.e., \( \Pr(i|u) \)). Candidate items with the highest aforementioned probability are recommended to \( u \). Note that \( \Pr(i|u) \) can be computed as

\[
\Pr(i|u) = \frac{\Pr(u, i)}{\Pr(u)} \propto \Pr(u, i)
\]  

(6.1)

According to the SIS model, we have

\[
\Pr(u, i) = \sum_{f \in F(u)} \sum_{z \in Z} \sum_{w \in W_i} \Pr(u, f, z, i, w)
\]  

(6.2)

Similar to [179], we assume items and contents are independently conditioned on the topics. As a result, the joint probability distribution over all factors is:

\[
\Pr(u, f, z, i, w) = \Pr(u) \Pr(f|u) \Pr(z|f) \Pr(i|z) \Pr(w|z)
\]  

(6.3)

where \( w \in W \) and \( W \) is the space of all item content.

The key observations from this model as shown in Figure 6.1 are 1) \( u \) and \( z \) are independently conditioned on \( f \), and 2) \( f, w \) and \( i \) are independently conditioned on \( z \). To model the item selection probability in terms of \( f \) and \( z \), we transform Equation (6.3) into the following form:

\[
\Pr(u, f, z, i, w) = \Pr(z) \Pr(u|f) \Pr(f|z) \Pr(i|z) \Pr(w|z)
\]

Notice that we will need to obtain a number of model parameters, including \( \Pr(z) \), \( \Pr(u|f) \), \( \Pr(f|z) \), \( \Pr(i|z) \), \( \Pr(w|z) \) in order to compute \( \Pr(u, f, z, i, w) \) and \( \Pr(u, i) \). Among them, \( \Pr(u|f) \) captures the social influence from friend \( f \) to user \( u \) and \( \Pr(f|z) \) depicts a user \( f \)'s preference, i.e., \( \Pr(z|f) = \frac{\Pr(z, f)}{\Pr(f)} \propto \Pr(z, f) \), where \( \Pr(z, f) = \Pr(z) \Pr(f|z) \). We will show how these parameters can be used to facilitate various applications in Section 6.4.

In this study, we employ expected maximization (EM) to learn those model parameters from the user-item-content history \( H \), i.e., \( \langle u, i, w \rangle \in H \), where \( u \in U \), \( i \in I \), and \( w \in W_i \) (i.e., \( W_i \) denotes the tag/word set associated with item \( i \)). Note that an item may contain multiple tags/words. For a history record of a user \( u \) selecting an item \( i \) where \( W_i = \{w_1, w_2, \ldots\} \), we
have $\langle u, i, w_k \rangle \in H, k = 1, 2, \cdots$. Note that existing expected maximization (EM) algorithms mostly handle single latent variable. Thus, it brings new challenges when there are multiple variables, i.e., $z$ and $f$, in the model and thus mandates a new and efficient learning algorithm. To address this issue, we have performed a detailed mathematical derivation to develop a new EM algorithm in order to learn the model parameters by statistical inference. Our model parameter learning algorithm is based on the idea of maximizing the log-likelihood of $L(\theta)$. 

\[
L(\theta) = \sum_{\langle u, i \rangle \in H} \log(\Pr(u, i|\theta))
\]

(6.4)

where $\theta$ denotes the model parameters, i.e., $\Pr(z), \Pr(u|f), \Pr(f|z), \Pr(i|z), \Pr(w|z)$, in our model.

The EM algorithm is a way to find model parameters to achieve local maximum of log-likelihood function (i.e., Equation (6.4)). Since direct maximizing $L(\theta)$ is difficult, EM algorithm applies an iterative method to improve model parameters step by step. Starting from the log-likelihood $L(\theta)$, we have:

\[
\begin{align*}
L(\theta) &= \log \prod_{\langle u, i \rangle \in H} \Pr(u, i|\theta) = \sum_{\langle u, i \rangle \in H} \log \Pr(u, i|\theta) \\
&= \sum_{\langle u, i \rangle \in H} \log \sum_{z, f, w} \Pr(u, i, w, z, f|\theta) \\
&= \sum_{\langle u, i \rangle \in H} \log \left( \sum_{z, f, w} \Pr(z, f|u, i, w, \theta_\lambda) \frac{\Pr(u, i, w, z, f|\theta)}{\Pr(z, f|u, i, w, \theta_\lambda)} \right) \\
&\geq \sum_{\langle u, i \rangle \in H} \sum_{z, f, w} \Pr(z, f|u, i, w, \theta_\lambda) \log \left( \frac{\Pr(u, i, w, z, f|\theta)}{\Pr(z, f|u, i, w, \theta_\lambda)} \right) \\
&\triangleq Q(\theta|\theta_\lambda)
\end{align*}
\]

(6.5)

Therefore, instead of maximizing $L(\theta)$, the EM algorithm tries to find model parameters $\theta_{x+1}$ to maximize $Q(\theta|\theta_\lambda)$. Therefore, we can drop constant terms w.r.t. $\theta$ as

\[
\begin{align*}
\theta_{x+1} &= \arg \max_\theta \{Q(\theta|\theta_\lambda)\} \\
&= \arg \max_\theta \left\{ \sum_{\langle u, i \rangle \in H} \sum_{z, f, w} \Pr(z, f|u, i, w, \theta_\lambda) \log \Pr(u, i, w, z, f|\theta) \right\} \\
&\geq \arg \max_\theta \left\{ \sum_{\langle u, i \rangle \in H} E_{z, f|u, i, w, \theta_\lambda} \{\log \Pr(u, i, w, z, f|\theta)\} \right\}
\end{align*}
\]

(6.6)
Therefore, the EM algorithm consists iterating:

1. E-step: Determine the conditional expectation in Equation (6.6).

2. M-step: Maximize this expectation with respect to $\theta$.

In E-step, instead of computing the expectation of the log-likelihood for individual latent variables (i.e., $z$ and $f$ separately), we propose to compute the expectation of the log-likelihood for $\langle z, f \rangle$ jointly. More specifically, we calculate

$$
\Pr(z, f|u, i, w) = \frac{\Pr(z) \Pr(f|z) \Pr(u|f) \Pr(i|z) \Pr(w|z)}{\sum_{z \in Z} \sum_{f \in F(u)} \Pr(z) \Pr(f|z) \Pr(u|f) \Pr(i|z) \Pr(w|z)}
$$

(6.7)

where the right hand side of Equation (6.7) only consists of the parameters in $\theta_x$. Note that we only need to compute the posteriors of those triples $\langle u, i, w \rangle \in H$, because the expectation to be maximized only weights on the observed user-item-word triples. While the friend space could potentially be the entire user space, the averaged number of friends per user is limited. The total size of parameters $\Pr(z, f|u, i, w)$ for each record $\langle u, i, w \rangle$ is estimated by $|z| \cdot |F(u)|$.

In EM algorithms, the E-step is to calculate the expected value of the log likelihood function, with respect to the conditional distribution of latent variables, under the current estimate of the model parameters. In our algorithm, $z$ and $f$ are considered jointly in E-step. This is very important because, if we consider only a one latent variable, say $z$, the M-step cannot estimate the parameter $\Pr(f|z)$, which depends on the joint probability of $z$ and $f$.

In the M-step, we find model parameters to maximize Equation (6.6), which needs to incorporate the following constraints

$$
\sum_z \Pr(z) = 1
$$

$$
\sum_u \Pr(u|f) = 1
$$

$$
\sum_f \Pr(f|z) = 1
$$

$$
\sum_i \Pr(i|z) = 1
$$

$$
\sum_w \Pr(w|z) = 1
$$

(6.8)

4In our collected real data sets, the averaged number of friends per user is less than 10.
Thus, we have

\[
H = \sum_{(u,i) \in H} \mathbb{E}_{z,f|u,i,w,\theta} \{ \log \Pr(u, i, w, z, f|\theta) \} + \lambda_1 (1 - \sum_z \Pr(z)) + \lambda_2 (1 - \sum_u \Pr(u|f)) \\
\quad + \lambda_3 (1 - \sum_f \Pr(f|u)) + \lambda_4 (1 - \sum_i \Pr(i|z)) + \lambda_5 (1 - \sum_w \Pr(w|z))
\]  

(6.9)

In M-step, model parameters are computed to maximize \( H \) found on the E-step as below.

\[
\Pr(i|z) \propto \sum_{(u',i',w') \in H} \sum_{f' \in F(u')} \Pr(z, f'|u', i, w')
\]

\[
\Pr(w|z) \propto \sum_{(u',i',w') \in H} \sum_{f' \in F(u')} \Pr(z, f'|u', i', w)
\]

\[
\Pr(u|f) \propto \sum_{(u',i',w') \in H} \sum_{f' \in F(u')} \Pr(z', f|u', i', w')
\]

(6.10)

\[
\Pr(f|z) \propto \sum_{(u',i',w') \in H} \sum_{f' \in F(u')} \Pr(z, f'|u', i', w')
\]

\[
\Pr(z) \propto \sum_{(u',i',w') \in H} \sum_{f' \in F(u')} \Pr(z, f'|u', i', w')
\]

By repeating the E-step and M-step, the EM Algorithm improves the model parameters iteratively until they converge to a local log-likelihood maximum.

### 6.4 Applications of Social Influence

In this section, we discuss how social influence learned from SIS benefits other applications such as group recommendation and viral marketing.

#### 6.4.1 Group Recommendation

Given a group of people \( G \), group recommendation aims to identify items that are welcomed by the whole group instead of individual group members, e.g., find romantic restaurants for a couple or movies for a family with kids. Although the SIS model targets on item recommendation for an individual user, the social influence learned based on SIS is very useful for group recommendation. In this section, we first briefly introduce the *aggregation-based group recommendation* strategies and then discuss our new strategy, called *social influence based group*
For group recommendation, a widely adopted approach is to apply some aggregation strategies to obtain a “consensus” group ranking/score for a candidate item. Two popular aggregation strategies, namely average and least misery, are proposed in [177]. Using the item access probability estimation $\text{Pr}(i|u)$ as the score of item $i$ to $u$, the average strategy calculates the group score of an item $i$ to a group $G$ as

$$S_{\text{average}}(G, i) = \frac{\sum_{u \in G} \text{Pr}(i|u)}{|G|} \quad (6.11)$$

On the other hand, the least misery strategy calculates the group score for an item $i$ to a group $G$ as the smallest predicted rating for item $i$ in the group, specifically

$$S_{\text{misery}}(G, i) = \min_{u \in G} \{\text{Pr}(i|u)\} \quad (6.12)$$

Basically, the item least disliked by each individual member shall have the highest group score for recommendation.

Note that $\text{Pr}(i|u)$ can be calculated based on our generative model by incorporating user behavior (UB) and item content (IC). These two aggregation-based group recommendation approaches capture a group consensus of item ranking by assuming all the decisions made by users are independent and equally important. However, in a group activity, people interact with and influence each other. Thus, we take social influence into account to reach a group consensus.

Intuitively, a user in a group activity selects an item due to two possible reasons: 1) her own preference of the item, 2) influence from other group members. One direct extension of the existing aggregation-based group recommendation strategies is to incorporate social influence in the calculation of probability value $\text{Pr}(i|u)$.

$$\text{Pr}(i|u) \propto \text{Pr}(u, i) \approx \sum_{f \in (F(u) \cap G)} \text{Pr}(u, f, i) \quad (6.13)$$

where

$$\text{Pr}(u, f, i) = \sum_{z \in Z} \text{Pr}(z) \text{Pr}(u|f) \text{Pr}(f|z) \text{Pr}(i|z) \quad (6.14)$$

can be easily obtained from the model parameters of SIS. Note that $\text{Pr}(u, f, i)$ denotes the
probability for $u$ to select $i$ under influence of $f$. If $f = u$, it estimates the probability that the user $u$’s own preference in this item. This newly derived $\Pr(i|u)$ is applicable to Average (Equation (6.11)) and Least Misery (Equation (6.12)) for group recommendation.

An alternative idea is that users in a group will listen to others’ opinions, i.e., social influence (instead of personal preferences) is given the highest priority to get a consensus score for the group. SIS can naturally support the group recommendation as a group selection activity, i.e., $u$ is influenced by other group members to jointly select item $i$. Consider a “two-member” group $G_2 = \{u_1, u_2\}$. To select an item for the group, user $u_1$ may influence user $u_2$ and vice versa. Therefore, we define the score for recommending an item $i$ to the group $G_2$ as

$$S_{\text{influence}}(G_2, i) = \Pr(u_1, u_2, i) + \Pr(u_2, u_1, i) \quad (6.15)$$

![Figure 6.2. Decompose an arbitrary group into a set of two-member groups.](image)

The ideas described above can be generalized for groups with more than two members by decomposing such a group into a set of two-member groups based on the friendship of members (see the example in Figure 6.2 for illustration). To make a group recommendation, we assume the social influence only happens between friends. Intuitively, if most pairs of friends in the group prefer a particular item, it would be a good candidate for recommendation to the group. Let $G$ denote a group with arbitrary cardinality. The score for recommending an item $i$ to $G$ is defined as the sum of $S_{\text{influence}}(G_2, i)$ score over all possible friend pairs in the group. Formally,

$$S_{\text{influence}}(G, i) = \sum_{\forall \{u, f\} \in G \times G, u \neq f, f \in F(u)} S_{\text{influence}}(\{u, f\}, i) \quad (6.16)$$

The ranking of items for group recommendation is based on the sorted group scores of items as defined above. We find superior performance of our social-influence strategy over the two aggregation strategies (to be shown in Section 6.5.6).
6.4.2 Viral Marketing

Recently, the phenomenon of influence propagation in social networks and its application in viral marketing have attracted tremendous interest. The idea is that when a user sees her social contacts performing an action such as purchasing a product, that user may decide to adopt the purchase as well [180, 181, 182, 183]. Obviously, social influence is important for viral marketing. In this section, we demonstrate how the social influence and user preference derived from our SIS model can be applied in the viral marketing study. Although the user-item selection process in SIS does not capture the causality in influence propagation, we assume that the social influence learned from our model may approximate the probability for a user to adopt the product given that her friend adopts the product.

Conventional research of viral marketing are mainly based on the social influence probability from $f$ to $u$, i.e., $Pr(u|f)$. We argue that, in addition to social influence, the personal preferences of users are also very important for viral marketing. Note that, as discussed earlier, a friend $f$ showing high social influence to a user $u$ doesn’t necessarily implies that $f$ and $u$ are highly similar in terms of their item access behaviors. Intuitively, whether a product will be adopted by a user is dependent on user preference of the product itself, even though we fully agree that social influence is important for viral marketing. In short, we argue that both user’s preference and social influence are critical to viral marketing. If solely based on social influence, a product marketing plan may likely to target on a user with high social influence and high interest in the product (even if his friends are not interested in the product based on their own preferences). As a result, the aforementioned conventional viral marketing plan based purely on social influence could fail.

Here, we adopt the popular Threshold Model [181] as a case study to demonstrate how to utilize the social influence and user preferences obtained based on SIS for viral marketing strategy design. Note that, in viral marketing study, each individual user is assumed to be either active (an adopter of the product) or inactive. We assume a setting where each user’s tendency to become active increases monotonically as more of her friends become active and focus on the case where users only switch from being inactive to being active, instead of switching in the other direction. In the model, the probability for a user $u$ to an item $i$ is denoted as $Pr_{VM}(i|u)$, which is calculated as follows. Let $F(u)$ denote the friend set of user $u$ and $F'(u) \subset F(u)$ denote
the subset including all active friends.

\[
Pr_{VM}(i|u) = \begin{cases} 
0 & |F'(u)| = 0 \\
\alpha Pr_{VM-self}(i|u) + (1 - \alpha) Pr_{VM-friend}(i|u) & \text{Otherwise}
\end{cases}
\]  

(6.17)

where \( \alpha \) is a weighting factor between 0 and 1. \( \alpha = 1 \) means users’ decision is solely based on their personal preferences; while \( \alpha = 0 \) means users’ decision is only based on the social influence, i.e., the case of conventional viral marketing study.

\[
Pr_{VM-self}(i|u) = Pr(i, u) = \sum_z \sum_w Pr(z|u) Pr(i|z) Pr(w|z)
\]  

(6.18)

where \( Pr(z|u) \), \( Pr(i|z) \) and \( Pr(w|z) \) are model parameters learned from our proposed model. And

\[
Pr_{VM-friend}(i, u) = \sum_{u' \in F'(u)} Pr(u|u')
\]  

(6.19)

where \( Pr(u|u') \) is the social influence parameter learned from our generative model.

Each user \( u \) has a threshold \( \theta_u \). Once \( Pr_{VM}(i|u) \) exceeds the threshold \( \theta_u \), user \( u \) will turn to active. The process of viral marketing starts with an initial set of users \( S \). Users in \( S \) get to know the item \( i \) by advertisement. Thus, a seed user \( s \in S \) has a initial probability \( Pr_{VM}(i|s) = \alpha Pr_{VM-self}(i|s) \). Of course, for viral marketing, we always want to select users interested in the items as the seed users. The diffusion process unfolds deterministically in discrete steps: in step \( t \), all users who were active in step \( t - 1 \) remain active; and we update the value of \( Pr_{VM}(i|u) \) for each inactive user \( u \) and activate any user \( u' \) for which its \( Pr_{VM}(i|u') \geq \theta_{u'} \).

### 6.5 Performance Evaluation

In this section, we conduct a comprehensive performance evaluation using two real datasets, one from last.fm and the other from whrrl.com. We develop web crawlers to collect theses two datasets, which include user-item accessing history, users’ friendship network and tags associated with each item. Besides, we collect group check-in history data from whrrl.com to validate our group recommendation approach. In our evaluation, we adopt the memory-based collaborative filtering approach (denoted as CF) as a baseline and propose to study the effec-
tiveness of various factors (i.e., social influence (SI), user behavior (UB) and item content (IC)) considered in our SIS model. The different configurations of factors included in our evaluation are: 1) user behavior (UB) [178], which is taken as the second baseline, 2) both user behavior and social influence (UB+SI), 3) user behavior and item content factors (UB+IC) [179], which is the third baseline, and 4) all of the user behavior, social influence and item content factors (UB+SI+IC), i.e., our SIS model. In this evaluation, we conduct an extensive set of experiments for item recommendation, group recommendation, viral marketing, and various user analysis studies.

6.5.1 Dataset Description

Here we first discuss the datasets, i.e., last.fm and whrrl.com, used in our experiments. Last.fm is an on-line music radio web service and whrrl.com is a location-based social network web service. The last.fm dataset contains music access history of 3,143 users over 23,467 unique songs; while whrrl.com dataset includes the check-in history of 7,145 users to 74,217 unique places. It is worth noting that the whrrl.com dataset includes 17,587 group check-in records which are very valuable for evaluating the group recommendation approaches. Additionally, both datasets have their user social networks available. The basic statistics of these two datasets are summarized in Table 6.1.

<table>
<thead>
<tr>
<th></th>
<th>last.fm</th>
<th>whrrl.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Users</td>
<td>3,143</td>
<td>7,145</td>
</tr>
<tr>
<td>Number of Items</td>
<td>23,467</td>
<td>74,217</td>
</tr>
<tr>
<td>User-Item Matrix Density</td>
<td>$8.02 \times 10^{-3}$</td>
<td>$2.3 \times 10^{-4}$</td>
</tr>
<tr>
<td>Average Friends per User</td>
<td>1.91</td>
<td>9.08</td>
</tr>
<tr>
<td>Average Tags per Item</td>
<td>4.92</td>
<td>2.73</td>
</tr>
<tr>
<td>Average Group Size</td>
<td>N/A</td>
<td>2.93</td>
</tr>
</tbody>
</table>

Table 6.1. Datasets statistics

Note that the effectiveness of recommendation service provisioned with sparse dataset (i.e., low-density user-item matrix) is usually not high. For example, the reported precision in [26] is 0.05 over a preprocessed dataset with $2.72 \times 10^{-4}$ density. Thus, in our experiments, we focus on comparing the relative performance of algorithms instead of their absolute effectiveness measures, which are expected to improve over time as the number of social network users continues to grow. Specifically, we use UB as the reference to compare the relative effectiveness measures, which are calculated as $\frac{X_{UB}}{UB}$, where $X$ are CF, UB+SI, UB+IC or SIS.
6.5.2 Item Recommendation

Item recommendation serves as the primary test case for our evaluation. For the various configurations degenerated from SIS, we use log-likelihood as model converge indicators and terminate the EM algorithms when an additional EM iteration cannot improve the training data’s log-likelihood by 0.0001 or when the maximum iteration threshold (empirically set with 50) is reached. We adopt the cross-validation method to obtain the precisions and recalls of item recommendation. For both datasets, we mark off 30% of item assess history corresponding to each user for testing. In other words, the rest 70% of user-item pairs are used as the training data to infer model parameters. After each model is learned, we use the model parameters to find $\forall i, \Pr(i|u)$ for all users. The precision and recall for top $n$ recommendations are used as
the evaluation metrics, where \( n = 5, 10, 20, 50 \) (5 is the default value).

Figure 6.3 and Figure 6.4 show the precision and recall of top 5 item recommendations for last.fm and whrrl.com by varying \(|Z|\), the size of latent topics. We find that social influence indeed improves the recommendation performance, for both UB+SI against UB and SIS against UB+IC. The result shows that the best recommendation performance is reached when the chosen topic size is around 60. Therefore, we set the default value of the latent topic size to 60 for the remaining experiments.

Figure 6.5 and Figure 6.6 compare the performance of various item recommendation approaches, using last.fm and whrrl.com, respectively. Note that UB’s performance is used as
the reference and these two figures demonstrate the performance ratio against the UB. As shown, all the model-based approaches clearly outperform the conventional memory-based collaborative filtering (CF), i.e., values of CF are smaller than 1 while all other ratio values are greater than 1. Again, we find that leveraging the social influence indeed improves the performance (i.e., values of UB+SI are greater than 1 and values of SIS are greater than those of UB+IC). Most importantly, SIS (which integrates user behavior, social influence and item content) achieves the best performance.

6.5.3 Topic Analysis

![Figure 6.7. Example topics learned from whrrl.com dataset, based on SIS](image)

<table>
<thead>
<tr>
<th>Topic No.</th>
<th>Representative Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 1</td>
<td>coffee, shops, theater, live performance, movie</td>
</tr>
<tr>
<td>Topic 7</td>
<td>sports, retails, recreation, food, store</td>
</tr>
<tr>
<td>Topic 15</td>
<td>bars, clubs, subs, pub, wine</td>
</tr>
<tr>
<td>Topic 28</td>
<td>restaurant, subs, sandwich, cafe, steak</td>
</tr>
<tr>
<td>Topic 35</td>
<td>store, service, grocery, retail, accessory</td>
</tr>
<tr>
<td>Topic 39</td>
<td>movie, museum, galleries, lodging, theater</td>
</tr>
<tr>
<td>Topic 49</td>
<td>restaurant, bars, Americans, Mexican, wine</td>
</tr>
</tbody>
</table>

Here, we study the latent topics learned from our SIS model. Notice that, in SIS, a latent topic is represented by a joint distribution of item and content. Figure 6.7 shows some latent topics learned from the whrrl.com dataset. As shown, places located in distance tend to belong
to different latent topics (e.g., places in topic 15 and topic 49), while categorical tags with very different semantics also tend to belong to different latent topics (e.g., representative tags in topic 39 and topic 49). Obviously, in the whrl.com dataset, the content and locations of places both have an important role in determining the latent topics. We find that several latent topics have similar representative tags, e.g., topic 15 and topic 49, but located in distance. On the other hand, some latent topics that are clustered at the same areas, e.g., topic 1 and topic 15, have quite different tags. Moreover, items (even with similar contents) in different spatial clusters tend to be in different latent topics, which confirms that the learned latent topics from our SIS model are distributions over both items (which exhibit strong locality in whrl.com dataset) and their content.

<table>
<thead>
<tr>
<th>Topic No.</th>
<th>Representative Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 1</td>
<td>80s, Bon Jovi, Robbie Williams, Guns N Roses, American</td>
</tr>
<tr>
<td>Topic 10</td>
<td>Rock, Guitar Virtuoso, Roxette, English, New Wave</td>
</tr>
<tr>
<td>Topic 16</td>
<td>Punk, Rock, 70s, Ramones, American</td>
</tr>
<tr>
<td>Topic 20</td>
<td>Alizée, French, alternative rock, Italian, Lorie</td>
</tr>
<tr>
<td>Topic 27</td>
<td>Usher, RnB, hip-hop, Craig David, urban</td>
</tr>
<tr>
<td>Topic 35</td>
<td>JAY-Z, hip hop, Kanye West, rap, east coast hip hop</td>
</tr>
<tr>
<td>Topic 49</td>
<td>jazz, Bill Evans, jazz piano, Bebop, saxophone</td>
</tr>
</tbody>
</table>

**Figure 6.8.** Example topics learned from last.fm dataset, based on SIS

Figure 6.8 shows some latent topics learned from the last.fm dataset. Topic 1 discovers popular American rock stars in 80’s, except for Robbie Williams who is an English singer. However, since the topic is discovered from users’ listening history, this may indicate that users like Bon Jovi may also like Robbie Williams. Topic 10 represents another genre in rock music with extensive guitar practice, e.g., Roxette is one of the representative bands. Topic 20 has two French female singer Alizée and Lorie, and indicates the topic may be related to Italian singers. Topic 27 contains genre related to R&B and two well known singers Usher (the king of R&B) and Craig David. Topic 35 discovers topic related to hip hop music, with representative singers Jay-Z and Kanye West. And topic 49 demonstrates music related to jazz and music instruments related to jazz (piano and saxophone).

### 6.5.4 Social Influence Study

In this section, we study the social influence between friends. Instead of investigating how social influence improves the recommendation performance, here we analyze the role of user
influence in different application domains (i.e., music listening vs. location check-in). We plot the distributions of social influence probabilities, denoted as $Pr(u|f)$, among all friend-pairs in Figure 6.9(a). Additionally, we also consider the probability for a user $u$ to make independent selection (denoted as $Pr(u|u)$) and plot the distributions of independent selection probabilities in Figure 6.9(b). Intuitively, we believe users making decisions mainly based on their own preferences, although friends’ opinions may affect the decision making to certain degree, i.e., the probability for a user to follow an uncommon preference of his friends is relatively small. Here, our experiments based on the dataset last.fm and whrrl.com confirm this intuition.

![Figure 6.9](image)

**Figure 6.9.** Social influence analysis

By comparing results between last.fm and whrrl.com datasets, we find that the friend influence in whrrl.com is smaller than that in last.fm. As shown in Figure 6.9(a), 95% (for last.fm) and 99% (for whrrl.com) of friend-pairs have friend influence values less than 0.05, respectively. This is probably because users in whrrl.com have more friends than users in last.fm (i.e., average 9.08 friends in whrrl.com vs. 1.91 friends in last.fm). Usually not all friends are influential and, very often, a small number of friends take the most part of social influence. From Figure 6.9(b), we find that although both datasets suggest significant independent selection probabilities, users in whrrl.com have smaller independent selection probabilities than those in last.fm. As shown in Figure 6.9(b), 5% (for whrrl.com) and 1% (for last.fm) of users have independent selection probability smaller than 0.8, respectively. Generally speaking, the social influence of users in location check-in activities is more significant than those in music listening activities. One explanation is that location check-ins are inherently social activities while music listening is usually for self-entertainment.
6.5.5 User Profile Analysis

Both user preference of a user and social influence from her friends can be learned through our SIS model to build user profiles. In this section, we analyze two user profiles to facilitate a better understanding of user behaviors. In Figure 6.10, we show the profile of user 338 (consisting of her preference of latent topics and influence from friends) from last.fm dataset. As shown, user 338 is influenced by user 1196 and user 668, with the influence probability values 0.005 and 0.043, respectively (other friends with lower influences to user 338 are omitted). Also, the top 5 topics of users’ preferences are also shown, where the probabilities representing users’ interests in the topics are labeled in the corresponding edges. Notice that there are 4 overlapped topics for user 1196 and user 338; while there are 3 overlapped topics between user 668 and user 338. In terms of common interests, user 1196 is more similar/correlated to user 338 than user 668. However, we find that user 668 exhibits more social influence upon user 338 than user 1196 does. Notice that the notion of social influence in our SIS model is able to capture the “abnormal” behavior of a user due to influence from her friends. Figure 6.10 confirms that friends with high correlation/similarity or share a lot common items do not necessarily exhibit strong social influence, while friends with less overlapped interests may still be quite influential, as some friends may have some special knowledge useful to others.\(^5\)

Similarly, we study the user profile discovered from whrrl.com dataset. Figure 6.11 shows a user profile for user 8, consisting of his top ranked topics and social influence from two most influential friends, i.e., user 65 and user 2175. As shown, user 65 and user 2175 have the same

\(^5\)As discussed in the Introduction, the ideas and assumptions made in many existing work contradict this observation.
influence on user 8 (who is relatively weak in independent selection) although they have different degrees of overlapped topics with user 8. Again, social correlation and similarity between friends do not reflect the social influence between them.

Figure 6.11. Example user profiles and social influence learned from whrrl.com dataset, by using SIS

6.5.6 Group Recommendation

Next, we report our findings in evaluating group recommendation algorithms, including the SIG algorithm we proposed, along with two aggregation-based strategies. We use the 17,587 group check-in records in whrrl.com in our experiment, where we consider a group check-in record (i.e., the ground truth) at a time and take the average of tested records. Notice that a record indicates a group of people visiting a place. An effective group recommendation algorithm should have this place ranked high among all the places recommended. Therefore, we propose a metric called relative ranking to evaluate the performance of the group recommendation algorithms. Suppose that a given algorithm returns a ranked list of $m$ items (i.e., all places in this experiment). If the actual visited place is ranked in the $l$-th position of the returned recommendation list, the relative ranking is calculated as $\frac{l}{m}$. For example, if an actual visited place is ranked 10th among a total of 100 items returned by a group recommendation algorithm, the relative ranking is $10/100 = 0.1$.

Figure 6.12 compares the performance of SIG, Average and Least Misery. The values in Y-axis represent the relative rankings of actual visited places (the lower the better). The conventional Average and Least Misery strategies are devised based on user behavior (UB)
and item content (IC). Here we also implement both algorithms using the model parameters discovered based on our SIS model. The result shown in Figure 6.12(a) indicates that SIG always outperforms the other four and reach its optimal point when the topic size is around 60. Besides, we also observe that our SIS model enhances the performance of both Average and Least Misery algorithms. Recall that the SIG algorithm only takes into account the social influence (without considering the user preference) in making group recommendation. The result is surprising as we may think that both preferences and social influence of group members are important for a group of whrrl users to decide where to go. However, this experimental result may hint that people tend to listen to other people’s opinions, by compromising their own preference, in group activities. Thus, social influence based group recommendation shows the best performance.

In Figure 6.12(b), we find that SIG outperforms the Average-SIS and Least Misery-SIS strategies for most group sizes (the result for conventional implementations are not shown for clarity of presentation). However, as the group size growing larger, the improvement obtained from SIG decreases. This finding implies that for smaller groups, the social influence among group members plays a major role in item selection for the group. However, for larger groups, the group consensus aggregated from individual preferences may dominate the group decision, as it probably becomes too complicated to communicate with all members when the group is large. This finding is consistent with our experience in activity planning, i.e., for a smaller group, one or two influencing members may significantly determine the activity venue. On the other hand, for a large group, the social influence from individuals may become difficult to take effect on the entire group. As a result, the group’s common interest dominates.
6.5.7 Viral Marketing

![Figure 6.13. A subgraph of social network labeled with social influence and user preference.](image)

To validate our idea about using social influence together with user preference for viral marketing study, we present a case to demonstrate that our approach can identify good targets for marketing. Here, we extract a small social graph from the last.fm dataset as shown in Figure 6.13. The social influence learned from SIS are labeled as the weight of social links in the figure. Here we select an item 2940 from last.fm as the marketing item for our case study. We also list each user’s preference to item 2940 on the left-lower corner of Figure 6.13. In this study based on the Threshold Model, we compare the effect of viral marketing with two different approaches: 1) a conventional method (baseline), solely based on the social influence; and 2) our proposal, which combines both user preference and social influence. In the experiments, we set the threshold $\theta_u$ as 0.01 and the weight between user’s preference to item 2940 and the influence from friends as 0.5. As such, these two methods execute for 10 iterations. For each seed user $s$, the corresponding effect of viral marketing $V_s$ (i.e., final number of activated users) are shown in Table 6.2, indicating our proposal is expected to be more effective than (or at least complementary to) the baseline.

![Table 6.2. Comparison of viral marketing strategies](image)

We also use the different seed selection made in both methods to illustrate the importance
of taking into account both the user preference and social influence for viral marketing. Notice that our proposal suggests to start with seed user 5 but the baseline method suggests user 4. Obviously, user 4 is chosen by the baseline because i) she has high social influence to her neighbors, and ii) she is at the center of the social network. Under this reasoning, user 5 or 6 are inferior to user 4 because they have lower influence, comparing to the hub user 4. Actually, user 5 is the best choice in our proposal because, even though user 5 may not influence user 4 directly, user 4 can still get activated easily since she likes the item 2940 (i.e., she has a great interest in 0.0172). Therefore, our proposal targets on user 5.

6.6 Summaries

In recent years, the so-called “social influence” has been incorporated into recommender systems based on various heuristics, e.g., assign extra similarity weight to friends in collaborative filtering. In this paper, we discuss the differences between social influence and social correlation/similarity and argue that the phenomenon of social influence can be captured quantitatively for use in various recommendations and applications. We propose a probabilistic generative model, called social influenced selection (SIS) model, to capture the the social influence between linked friends and user preferences and develop an efficient algorithm to discover those valuable information from real datasets through statistical inference. A novel contribution of this work is a coherent approach which facilitates seamless integration of social network, user behavior and item content for recommendation. Additionally, the social influence and user preference extracted based on SIS can be used for a variety of applications, including group recommendation, viral marketing, user analysis, and topic analysis. Moreover, through a comprehensive set of empirical experiments using datasets collected from last.fm and whrrl.com, a number of interesting findings are obtained, including: 1) the social influence is beneficial for recommender systems and item recommendation made based on SIS (which integrates user behavior, social influence and item content) achieves the best performance; 2) user behavior (in form of location check-ins) in whrrl.com shows the spatial clustering phenomenon, and items (even with similar contents) in different spatial clusters tend to belong in different latent topics; 3) users making decisions mainly based on their own preferences, although friends’ opinions may affect the decision making to certain degree, i.e., the probability for a user to follow an uncommon preference of his friends is relatively small; 4) friends with high correlation/similarity or share a lot common items do not necessarily exhibit strong social influence, while friends...
with less overlapped interests may still be quite influential; 5) in group activities, people tend to listen to other people’s opinions, by compromising their own preference; and 6) combining both user preference and social influence (our strategy for viral marketing) is expected to be more effective than (or at least complementary to) the conventional approach solely based on social influence.
Conclusions and Future Works

In this thesis, we proposed several techniques proven to be effective in provisioning of search and recommendation services over user generated geo-social data. In this chapter, we summarize our contributions made in this research, followed by a discussion of our future research plans.

- **Travelogue Digests Mining.** we develop a travelogue service that discovers and conveys various travelogue (or trip blog) digests, in form of theme locations, geographical scope, traveling trajectory and location snippet, to users. In this service, theme locations in a travelogue are the core information to discover. Thus we aim to address the problem of *theme location discovery* to enable the above travelogue services. Due to the inherent ambiguity of location relevance, we perform location relevance mining (LRM) in two complementary angles, relevance classification and relevance ranking, to provide comprehensive understanding of locations. Furthermore, we explore the *textual* (e.g., surrounding words) and *geographical* (e.g., geographical relationship among locations) features of locations to develop a co-training model for enhancement of classification performance. Built upon the mining result of LRM, we develop a series of techniques for provisioning of the aforementioned travelogue digests in our travelogue system.

In the future, we plan to improve the traveling trajectory extraction by considering context provided in travelogue, and utilize location relevance mining techniques to support other travelogue services, such as location-based search. Besides, we plan to investigate the impact of our study on the other travelogue studies, such as location topic word extraction and destination summarization [82].
• **Semantic Annotation of Places.** we develop a semantic annotation technique for location-based social networks (e.g., Foursquare and Whrrl) to automatically annotate all places with category tags which are a crucial prerequisite for location search. Our annotation algorithm learns a binary support vector machine (SVM) classifier for each tag in the tag space to support multi-label classification. Based on the check-in behavior of users, we extract features of places from i) *explicit patterns* (EP) of individual places and ii) *implicit relatedness* (IR) among similar places. The features extracted from EP are summarized from all check-ins at a specific place. The features from IR are derived by building a novel *network of related places* (NRP) where similar places are linked by virtual edges. Upon NRP, we determine the probability of a category tag for each place by exploring the relatedness of places. Both EP and IR features are complementary with each other and beneficial for the proposed classification task.

Through our analysis on the Whrrl dataset, we find some semantic tags usually co-occur, e.g., restaurant and bars. In the future, we plan to explore the correlation among semantic tags for the semantic annotation of places. In addition, we plan to include some alternative approaches (e.g., [184]) for comparison and to use multiple large-scale datasets to validate our proposed SAP algorithm.

• **POI Recommendation in LBSNs.** we provide a point-of-interests (POIs)\(^1\) recommendation service for the rapid growing location-based social networks (LBSNs), e.g., Foursquare, Whrrl, etc. The idea is to explore user preference, social influence and geographical influence for POI recommendations. In addition to deriving user preference based on user-based collaborative filtering and exploring social influence from friends, we put a special emphasis on geographical influence due to the spatial clustering phenomenon exhibited in user check-in activities of LBSNs. We propose a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence.

The semantic tags of POIs contain very rich information brought in by LBSN users. As for the next step, we plan to incorporate the semantic tags of POIs to further improve the unified POI recommendation framework we proposed in this work. Besides, we also plan to explore other model based approaches such as matrix factorization to realize POI recommendation in LBSNs.

\(^1\)point-of-interests (POIs) are also referred as places in our context of location-based social networks.
• **Social Influence Selection (SIS) Model.** Social friendship has been shown beneficial for item recommendation for years. However, existing approaches mostly incorporate social friendship into recommender systems by heuristics. Here, we argue that social influence between friends can be captured quantitatively and propose a probabilistic generative model, called **social influenced selection (SIS)**, to model the decision making of item selection (e.g., what book to buy or where to dine). Based on SIS, we mine the social influence between linked friends and the personal preferences of users through statistical inference. To address the challenges arising from multiple layers of hidden factors in SIS, we develop a new parameter learning algorithm based on expectation maximization (EM). Moreover, we show that the mined social influence and user preferences are valuable for group recommendation and viral marketing.

As for the future work, we would like to further extend the SIS model to capture social influences with respect to latent topics. In addition, we also plan to conduct field study to demonstrate the advantages of our proposed viral marketing strategy. Finally, instead of relying on the learned knowledge from SIS (which for individual item selection) to make recommendations for a group, we plan to develop a new generative model to capture the item selection process for group activities.

To validate the performance of all the proposed techniques, we conducted an extensive set of experiments. All of them are shown to outperform existing state-of-the-art correspondences. As we can anticipate, with the flourish of location-based social networking services, the volume of user generated geo-social data grows exponentially. In the future, we aim to propose mining algorithms which are capable handle large scale data set, by using hadoop and map-reduce algorithm paradigm.
Bibliography


[22] “OpenTable, http://www.opentable.com/.”


[34] 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.


[145] I. Konstas, V. Stathopoulos, and J. M. Jose, “On social networks and collaborative recommenda-

and M. Tennenholtz, “Trust-based recommendation systems: an axiomatic approach,”

[147] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in

tive recommender by fusing explicit social relationships,” in ACM RecSys-Workshop,
2009, pp. 49–56.


[152] H. Ma, M. R. Lyu, and I. King, “Learning to recommend with trust and distrust relationships,” in


[154] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Sys-
tems: A Survey of the State-of-the-Art and Possible Extensions,” IEEE TKDE, vol. 17,
no. 6, pp. 734–749, 2005.


[156] A. M. Ferman, J. H. Errico, P. van Beek, and M. I. Sezan, “Content-based filtering and


[158] D. Billsus and M. J. Pazzani, “Learning collaborative information filters,” in ICML,

[159] H. Yildirim and M. S. Krishnamoorthy, “A random walk method for alleviating the


Vita
Mao Ye

Mr. Mao Ye obtained his Bachelor’s degree in Computer Science from Nanjing University, China, and started his PhD study in the Pennsylvania State University in Fall 2007. In his 5 year PhD program, Mao Ye completed all required courseworks within his first two academic years and pass the PhD candidacy examination in his first year and both the English proficiency test and the comprehensive examination by the end of his second year. He coauthored more than twenty research papers, some of which were published in prestigious conference proceedings (such as SIGKDD, SIGIR, WWW, ICDE, etc.) and journals (such as TKDE, etc.); and he gave talks, presentations and demonstrations about his latest research results to the research community and the general public, and provided professional services as external reviewers for conferences (such as ICDCS, ICDE, MDM, etc.) and journals (such as TIST, TKDE, JPDC, etc.). In recent years, he assisted his PhD advisor in mentoring one student in MSc program for his master thesis, one master student in MEng program for his master dissertation, and one junior PhD students in his research group. He is currently the student member of the ACM and the IEEE.