Doctoral Thesis

Location Knowledge Discovery from User Activities

Chenyi ZHUANG

June 2017

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Doctoral Thesis
submitted to Department of Social Informatics,
Graduate School of Informatics,
Kyoto University
in partial fulfillment of the requirements for the degree of
DOCTOR of INFORMATICS

Thesis Committee: Masatoshi Yoshikawa, Professor
Qiang Ma, Associate Professor
Toru Ishida, Professor
Michihiko Minoh, Professor

Copyright © 2017 Chenyi ZHUANG All Rights Reserved.

Location Knowledge Discovery from User Activities*

Chenyi ZHUANG

Abstract

The explosion of Social Networking Services (SNSs), such as Twitter, Facebook, and Flickr, has lead to a wealth of research into social media content and various social graphs. Recently, with the development of mobile devices, the Global Positioning System (GPS), and Web 2.0 technologies, users can easily share their geo-spatial locations and location related contents in the physical world via SNSs. For example, major search engines, such as Google, Bing, and Yahoo, now incorporate these essential User Generated Content (UGC) and trend analysis in their results. How to mine the meaningful knowledge from these tens of thousands heterogeneous UGC has become a problem of significant interest.

The *location dimension* bridges the gap between the physical world and the digital online SNSs, giving rise to new opportunities and challenges in the social media content based research. Therefore, the location knowledge discovery becomes the key problem in this research area. Our research attacks this problem by analyzing the user activities. Corresponding to the two essential roles, i.e., "user" and "location", our work could be generally divided into three parts: (1) User Profiling; (2) Location Profiling; and (3) User-to-Location Profiling. Specifically, the major contributions in this dissertation are as follows.

In the User Profiling work, we proposed a new concept, *location authority*, for mining information asymmetries among different online user groups. By considering a user's different available social, temporal, and spatial information, we devised three different models to estimate his/her location authority, respectively. Specifically, a social-network-driven model, time-driven model, and location-driven model are devised to estimate the extent of user's familiarity with a target location. Furthermore, two ensembles of these models are proposed, which demonstrate robust performance across different user profiling tasks.

*Doctoral Thesis, Department of Social Informatics, Graduate School of Informatics, Kyoto University, KU-I-DT6960-26-4703, June 2017.

In the Location Profiling work, not only a single location's characteristics but also the relations among different locations are analyzed. To the best of our knowledge, we are the first who propose to use two dimensions, i.e., user group distribution based popularity and scenery quality, to describe a location. Accordingly, we devise several methods to quantify these two dimensions. To extract the meaningful relations among different locations, our work is among the first of its kind, which constructs a location relation knowledge graph. On this graph, locations' (i.e., graph nodes) temporal and spatial information are embedded in a way that the computers can understand, as well as the relations represented as graph edges. To construct this knowledge graph, two new unsupervised relation extraction methods are devised to learn the relations among nodes (locations) on this graph.

In the User-to-Location Profiling work, we predict the extent (high or low) of user attention to different locations in a city. By utilizing the location knowledge in our previous work, we devise two generative algorithms to simulate the user attention dynamics in a city. Meanwhile, by gaining insights into the interactions between users and locations, our work can detect some user movement patterns, which can be regarded as different lifestyles in a city. The essential patterns learned can be used in a wide range of problem domains, such as city planning, geo-social recommendation, traffic forecasting, and so on.

To draw a brief conclusion, the work presented in this dissertation have made the following progress. (1) We proposed novel viewpoints for jointly analyzing *user* and *location* from LBSNs; (2) We devised novel models and algorithms for gaining further insights into the knowledge mining of *user* and *location* in the real world; (3) Our work were verified by experiments that were conducted in real scenes. Encouraging results were obtained by comparison with state-of-the-art baselines.

Keywords: Location Knowledge Discovery, User Profiling, Urban Informatics, Social Media Analysis, Probabilistic Model, Matrix Factorization.

CONTENTS

1 Introduction **1**

1.1 Background 1

1.1.1 Social Background 1

1.1.2 Technological Background 4

1.2 Overview of the Research 6

1.2.1 User Activities & Location Knowledge 6

1.2.2 Major Contribution 7

1.3 Motivation and Major Tasks 10

1.3.1 User Profiling 10

1.3.2 Single Location Profiling 10

1.3.3 Location Relation Profiling 11

1.3.4 User-to-Location Attention Prediction 11

1.4 Dissertation Structure 12

2 Related Work **13**

2.1 Computational Social Science 13

2.2 Urban Computing 15

2.3 Matrix Factorization 17

3 User Profiling **21**

3.1 Motivation 21

3.2 Related Work 24

3.3 Users' Location-authority Analysis 25

3.3.1 Social Network: "Is the user living in the target city?" 25

3.3.2 Visiting Frequency: "Does the user visit the target city?" . . . 27

3.3.3 Mobility Behavior: "Where do users visit in a target city?" . . 28
3.3.4 Model Integration 33
3.4 Experimental Evaluation 36
3.4.1 Data Preparation 36
3.4.2 Parameter Selection 37
3.4.3 User-classification Evaluation 37
3.4.4 Visualizing the Classification Results 43

4 Single Location Profiling **47**
4.1 Motivation 47
4.2 Related Work 49
4.3 Preliminary Work 51
4.3.1 Data Construction 51
4.3.2 Overview of the Two Tasks 52
4.4 User Distribution based Popularity Estimation 53
4.4.1 User Classification 53
4.4.2 Location Popularity Score Calculation 54
4.5 Scenery Quality Evaluation 55
4.5.1 Criteria 1: Image Labeling 56
4.5.2 Criteria 2: Social Appreciation based Analysis 56
4.5.3 Criteria 3: Photographer Attention based Analysis 58
4.6 Experimental Evaluation 61
4.6.1 Data Preparation 62
4.6.2 Parameter Selection: 62
4.6.3 Experiment on Location Popularity Estimation 63
4.6.4 Experiment on Scenery Quality Evaluation 65
4.7 Application Scenario: Discovering Obscure POIs 67
4.7.1 System Architecture 68
4.7.2 Exploring POIs on Map 69

5 Location Relation Profiling **73**
5.1 Motivation 73
5.2 Related Work 75
5.3 Preliminary 76
5.4 Knowledge Graph Construction 76
5.4.1 Temporal View-based Construction 76

5.4.2 Spatial View-based Construction . . . 77
5.4.3 Spatio-temporal View-based Construction . . . 78

6 User-to-Location Attention Prediction 83
6.1 Motivation . . . 83
6.2 Related Work . . . 86
6.3 Transition Matrix based Probabilistic Matrix Factorization . . . 88
6.3.1 Problem Definition . . . 88
6.3.2 Modeling . . . 89
6.3.3 Parameter Learning . . . 92
6.3.4 Predicting . . . 93
6.4 Knowledge Graph based Probabilistic Matrix Factorization . . . 94
6.4.1 Problem Definition . . . 94
6.4.2 Modeling . . . 94
6.4.3 Parameter Learning . . . 95
6.4.4 Predicting . . . 95
6.5 Experimental Evaluation . . . 95
6.5.1 Data Preparation . . . 96
6.5.2 Evaluation Schema . . . 98
6.5.3 Our Methods and Baselines . . . 98
6.5.4 Evaluation on TMPMF . . . 100
6.5.5 Evaluation on KGPMF . . . 104
6.5.6 Visualization . . . 105

7 Conclusion and Future Work 109
7.1 Conclusion . . . 109
7.2 Future Work . . . 111

Acknowledgements 113

References 115

Selected List of Publications 127

Appendix 129
A Proof of Corollary 1 . . . 129
B Derivation of the Update Equations for Eq. 6.8 . . . 129

List of Figures

1.1 Illustrations of User Generated Content (UGC) and social networks. . 2
1.2 Illustrations of locations and location networks. 3
1.3 Illustration of the relation between "User" and "Location". 4
1.4 The "Location Knowledge". 6
1.5 Overview of our research. 7
1.6 The impact of our work. 9

3.1 Mobility behavior of different user groups. Red: tourists; blue: locals; yellow: either. 23
3.2 Social-network-driven method. Each node represents a user. Edges denote a friendship between two users. The scores measure the authority levels. 27
3.3 Indegree and outdegree distributions of 94 samples from the social network N (see Definition 1). Here, 47 blue samples are correctly identified, while the other 47 red samples are misidentified. 34
3.4 Labeling variances on the top 100 active users in each city, calculated using the Jaccard index. 39
3.5 Labeling variances on the bottom 100 inactive users in each city, calculated using the Jaccard index. 39
3.6 Overall results for the location-authority-based user-classification task. By drawing the ROC curves, the five models' performance on both active (upper bound) and inactive (lower bound) datasets are presented. 40

3.7 Detailed results for the SVM model: (a) the parameter-tuning process for obtaining the best classification model; (b) and (c) results from respectively using the $Active_{set}$ or $Inactive_{set}$ as the training set and the other for testing. Because the SVM model has only two outputs (high or low authority), there is only one threshold for the ROC curves. 41
3.8 Visualization of different user groups' mobility behavior in Beijing, Kyoto, and San Francisco. Temporal information is analyzed in (a), whereas location information is utilized in (b). 44

4.1 A sightseeing location can be evaluated from two aspects, i.e., (1) popularity and (2) scenery quality. 48
4.2 Data construction. Details are described in Definitions 1, 2 and 4. . . . 52
4.3 The framework of ranking spots. Corresponding to Sections 4.5.1, 4.5.2 and 4.5.3 respectively, our system uses three criteria (i.e., the three questions listed in this figure) for scenery quality evaluation. . . 55
4.4 Image classification. In (a), an image is vertically divided into 256 bins, and for each, the points of detected edges are counted to set up the histogram (i.e., EH). (b) presents the saturation (X-axis) and hue (Y-axis) distributions of close-up view images. 59
4.5 Distribution fitting. Red: 22,421 geo-tagged maple images taken at 32 spots with beautiful maples; Blue: 10,748 geo-tagged cherry blossom images taken at 35 spots with beautiful cherry blossom. 61
4.6 Precision-recall curves for discovering obscure cherry blossoms and maple trees in Kyoto. Each query's F-measure is also computed and shown in this figure. 64
4.7 NDCG values of the top-k ranking results evaluated by criteria "v" and "r". 67
4.8 The architecture of our system. It is comprised of three components, i.e., data construction, obscure sightseeing spots mining and recommendation. 69
4.9 Interfaces for the proposed application, in which Google Maps' APIs are utilized. From deep blue to deep red, the color of each location represents how popular (or obscure) it is for sightseeing. 70

5.1 Visual representation of S. On the x-axis, $[0 \sim 23]$ and $[24 \sim 47]$ are workday and holiday hours, respectively. 77

6.1 User attention dynamics. (a) shows the density estimation of our check-in dataset in Beijing. (b) shows [1]'s statistical results of the mobility behavior of different photographer groups in Beijing. Red: tourists; blue: locals; yellow: either. 84
6.2 The graphical illustration of our Transition Matrix based Probabilistic Matrix Factorization (TMPMF). The gray nodes, X and T, are observed data, while the others are latent variables. 88
6.3 Dataset preparation. (c) and (d) present the heat maps of the 50×50 grids' out-degrees in transition matrix T. 97
6.4 Performance comparison of TMPMF, GeoMF, CMF, GNMF and MF based on recall@M ($M = 10, 15, ..., 100$) (see Eq. 6.19). For the mutual dependency graph T, bus and taxi transition information is used, respectively. For the sparse-dense setting, the performance is calculated and presented when $x = 10\%, 30\%$. Please note that the value ranges of the y-axis are different. 101
6.5 Performance comparison of TMPMF and GeoMF based on recall@M ($M = 10, 15, ..., 100$) (see Eq. 6.19). In this figure, we present TMPMF's performance when the taxi transition T_{taxi} is used. For GeoMF, we use the exactly same POI-influence-areas matrix Y in [2] to obtain the best performance. 102
6.6 The plot of prediction recall@$M = 100$ for the proposed TMPMF algorithm by varying moving pattern parameter λ_P and dependency parameter λ_T. 103
6.7 Performance comparison of our KGPMF based methods and baselines based on recall@M ($M = 10, 20, ..., 100$). 104
6.8 Moving pattern visualization. Labeled by red and blue, in (a) and (b), two moving patterns are presented on Google map. By zooming in on the map, (c) and (d) show some representative grids involved in these two moving patterns. 106
6.9 Relation visualization. Red: heads; blue: tails. 108

LIST OF TABLES

3.1 Notations used for the location-driven model. 30
3.2 Datasets collected from the three cities: Beijing, Kyoto, and San Francisco. 36
3.3 Contextual data used in Algorithm 2 and Eq. 3.2. 36
3.4 Calculating the true-positive rate and the false-positive rate. 38
3.5 Labeling results of the top 100 active users in each city, denoted by $Active_{set}$. 38
3.6 Labeling results of the bottom 100 inactive users in each city, denoted by $Inactive_{set}$. 39
3.7 Classification accuracy of the general SVM models when using different numbers of features. $\mathfrak{D}_1$: $Active_{set}(Beijing)$; $\mathfrak{D}_2$: $Active_{set}(Kyoto)$; $\mathfrak{D}_3$: $Active_{set}(SanFrancisco)$; $\mathfrak{D}_4$: $Inactive_{set}(Beijing)$; $\mathfrak{D}_5$: $Inactive_{set}(Kyoto)$; $\mathfrak{D}_6$: $Inactive_{set}(SanFrancisco)$; 41
3.8 Representative users for the two latent groups. 45

4.1 Details of the target two image datasets that are labeled by the queries $q_1 =$ "$Kyoto, Maples$" and $q_2 =$ "$Kyoto, CherryBlossom$". 62
4.2 Details of the cherry blossom dataset and the maple tree dataset. . . . 63
4.3 Two criteria of calculating scenery ranking scores. 66

6.1 Notations used in the algorithm. 90

CHAPTER 1

INTRODUCTION

In this chapter, we first introduce (1) what is "User"? and (2) what is "Location"? on Location-Based Social Networks (LBSNs). The relations between "User" and "Location" will also be described. Then, the technical background will be introduced. After that, the overview of our research will be introduced. On the basis of the observations from the real world, we introduce the research motivations followed by the contributions we have made in each of our work, as well as the corresponding problem domains to which our work can be apply. At last, we show the structure of this dissertation.

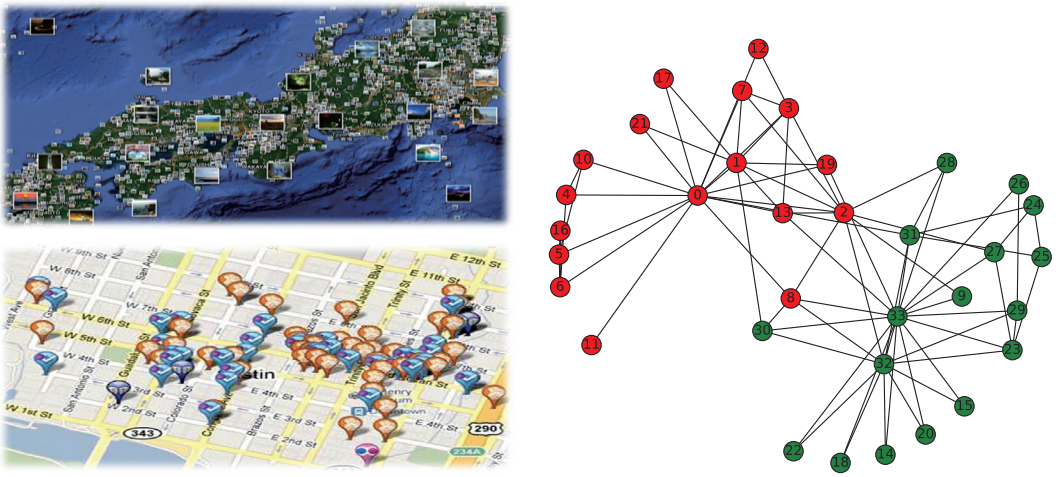
1.1 Background

1.1.1 Social Background

User

In our research area, a user is a person who uses social network services (SNSs). Generally, a user often has a user account and is identified to the SNSs by a username. However, in our work, we mainly focus on the User Generated Content (UGC) and the social networks on which users are connected with each other. As shown in Figure 1.1, we illustrate some simple examples about UGC and social networks.

User Generated Content. UGC is any form of content created by users and always made available publicly on the Internet. UGC can be used for a wide range of



(a) UGC: geo-tagged images and check-ins

(b) A small example of social network

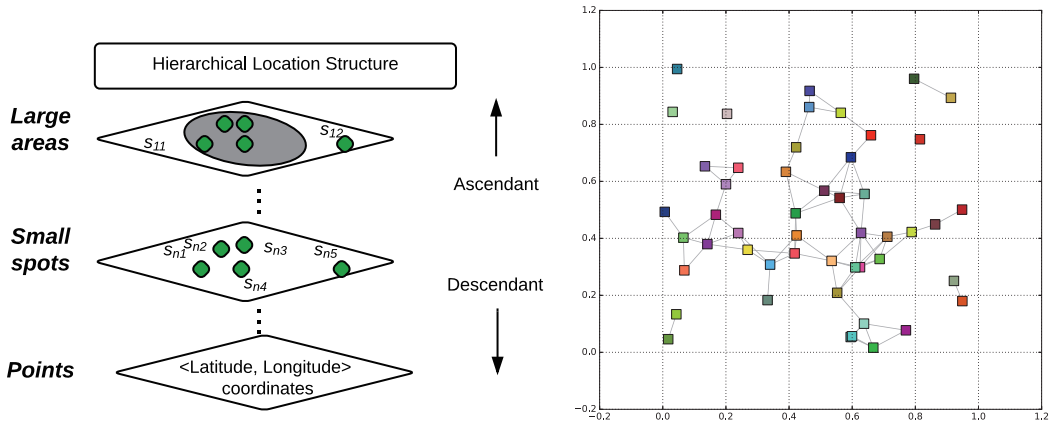
Figure 1.1. Illustrations of User Generated Content (UGC) and social networks.

applications in different research areas, including web searches, ad targeting, urban computing, and so on. Since we are aiming to analyze the characteristics of locations, the UGC collected in our work are mainly geo-tagged.

In Figure 1.1a, it shows two kinds of geo-tagged UGC that are publicly available on SNSs. The vital information geo-tagged UGC contains provides researchers with excellent opportunities for Points of Interests (POIs) discovery [3], user mobility simulation [4], landmark 3D reconstruction [5], and so on. By taking our work for example, by analyzing the geo-tagged social images, we propose novel methods to evaluate the scenery quality of a location. In addition, we use geo-tagged check-ins to predict the user attention dynamics to locations in a city.

Social Network. Figure 1.1b illustrates a social network. On this network, a user is denoted as a node and an edge indicates the relationship between two users (nodes). We can regard a social network as an online sociology structure, which is useful to study relationships between individuals, groups, and organizations. For example, the network shown in Figure 1.1b is called Karate Club Graph, which has two communities labeled by red and green colors, respectively. By analyzing the netowrk's structure, a lof of related work are conducted to correctly classify these two communities [6].

In our work, by analyzing the topology of a social network, we aims to mine the user authority to locations. A friendship based social network (i.e., edges on the network indicate friendships among users) is collected and then analyzed to estimate user familiarity with different locations.



(a) A hierarchical structure of locations

(b) A small example of location network

Figure 1.2. Illustrations of locations and location networks.

Location

A location can be represented in different forms. Generally, there are two forms to describe a location: (1) From the geography standpoint, a location refers to a specific physical point on Earth, which is a closed shape defined by a connected sequence of {latitude, longitude} coordinate pairs. As shown in Figure 1.2a, a location usually has a hierarchical geospatial representations from *points* to *areas*. (2) From the semantic standpoint, a location could be represented in a symbolic form. Namely, someone could use different categories (e.g., home, office, shopping mall, residential area, and so on.) to describe a location. Since our work rarely involve semantic information based analysis, we refer to the geography based definition.

Location Network. Similar to the social network, locations can also be connected with each other to construct a location graph of which nodes and edges are their locations and relations. Figure 1.2b shows some locations on a map. Among them, some edges are added. Obviously, the relations among locations can be explained in different ways. For instance, an edge could be explained to indicate the similarity between its head and tail nodes (locations). In another case, an edge is constructed to characterize the physical distance between two locations. However, in our work, we are the first who construct a location graph to represent the location functionality dependencies. By mining the functionality dependencies (e.g., a tight dependency between *work locations* and *residential locations* [4]), meaningful movement patterns (also can be regarded as lifestyles) are discovered.

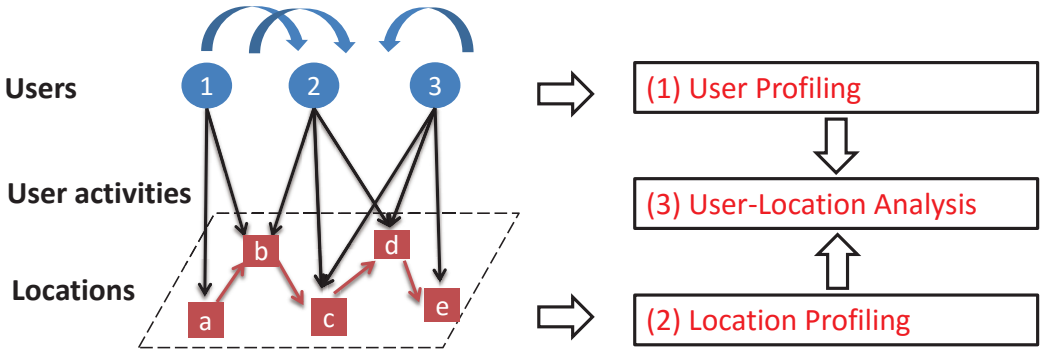


Figure 1.3. Illustration of the relation between "User" and "Location".

User-Location

Benefiting from the advances in LBSNs, they bring online social networks (i.e., Users) back to reality (i.e., Locations), bridging the gap between physical world and online SNSs. It does provide us a big opportunity to jointly study on *Users* and *Locations*.

Figure 1.3 shows an ecosystem of *Users* and *Locations*. We can find that *User Activities* plays a middleware role who overcame the gap between *Users* and *Locations*. Furthermore, as the output of *User Activities*, UGC is the key factor in this research area. More precisely, the geo-tagged UGC (e.g., geo-tagged images and check-ins shown in Figure 1.1a) contains both user and location information. To this end, the research problems related to LBSNs have been boiled down to how to efficiently analyze the publicly available UGC.

Our work introduced in this dissertation is committed to solving this problem. We have not only proposed several novel research perspectives for inspiring more interest in this wide problem domain, but also made some technical breakthroughs that can be applied in different machine learning or information retrieval models.

1.1.2 Technological Background

Recently, more and more attention has been paid to solve social problems using computing technologies. Accordingly, new concepts, like "Computational Social Science", "Urban Computing", "Big Data" and so on, are proposed in the technical field. Similar to these concepts, our work is aiming to bridge the gap between physical world and online SNSs. Since there are tens of thousands social problems to be addressed, in

this subsection, we show some related work that have been done in this wide research domain followed by an introduction of challenges. The detailed contributions of our work will be described in Section 1.3.

Computational Social Science

In [7], the authors proposed a new paradigm: *computational social science with big data*. They find that technology advances (in IT, Internet, Social Media, Data Analytics) prompted the rise of big data. And, the big data prompted scientific convergence across academic disciplines (e.g., computer science, sociology, economics, etc.) to discover new knowledge. The social science research has emphasized development of theoretical models and testable hypotheses, which lead to many challenges. For example, research with data at terabyte or petabyte scale challenges how we work with assumptions and theories.

Urban Computing

In a survey about "Urban Computing" [8], the authors argued that computer sciences can also be applied in conventional city-related fields, like transportation, civil engineering, environment, economy, ecology and sociology in the context of urban spaces. They summarized the typical technologies that are needed in urban computing, which are about urban sensing, urban data management, knowledge fusion across heterogeneous data, and urban data visualization. In addition, some missing research topics are introduced for inspiring further research work.

Geographic Information System

Since our work is going to mine the knowledge of locations, another related technology is Geographic Information System (GIS), which is a large domain within the broader academic discipline of geoinformatics. Location intelligence is one of them, which is a process of deriving meaningful insight from geospatial data relationships to solve a particular problem. The corresponding problems include how to visualize, question, analyze, and interpret data to understand relationships, patterns, and trends.

In the context of the big picture above, our work aims to jointly consider "Users", "Locations", and "User-Location Relations" from LBSNs to mine various knowledge. Our work has made new contributions to solving social problems including sightseeing, city planing, geo-social recommendation, and so on.

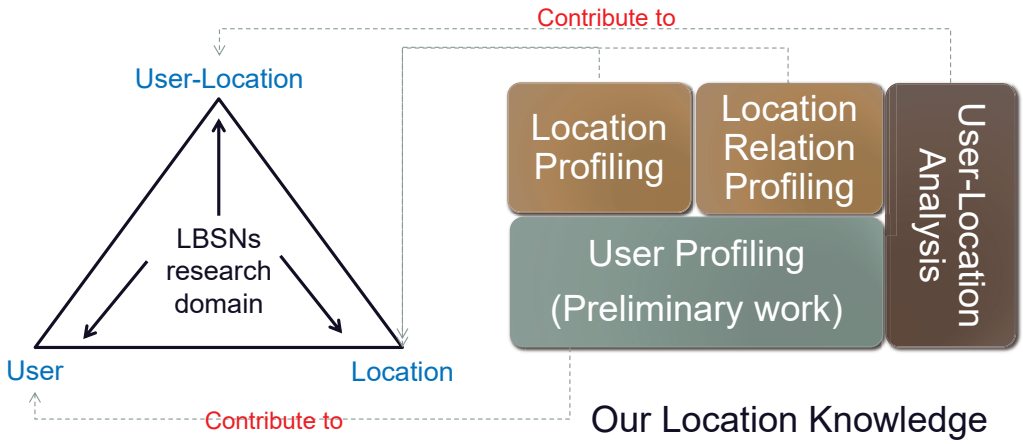


Figure 1.4. The "Location Knowledge".

1.2 Overview of the Research

In this section, we first explain what are *user activities* and *location knowledge*. Then, from a high technical perspective, the key idea (i.e., major contribution) of our research will be described. After that, from a high application based perspective, we will also discuss the corresponding problem domains to which our work can be applied, followed by the impact of our work (i.e., a benign cycle).

1.2.1 User Activities & Location Knowledge

User Activities. Figure 1.3 shows the definition of *user activities*, i.e., user activities are the interactions among users and locations. Specifically, on the basis of the collected raw data in our research, the user-user interactions are represented by the social network, whereas the user-location interactions are represented by geo-tagged images, check-ins, GPS trajectories, and so on. Accordingly, all the UGC could be regarded as different kinds of user activities.

Location Knowledge. Figure 1.4 shows the definition of *location knowledge*. Specifically, the user profiling knowledge is the prior knowledge. On the basis of the prior knowledge, a single location's and location-location relation's knowledge could be mined. Finally, we will also mine the user-location interaction knowledge. Since all of the location knowledge related components shown in Figure 1.4 are interdependent, regarded as a whole, we call them the location knowledge.

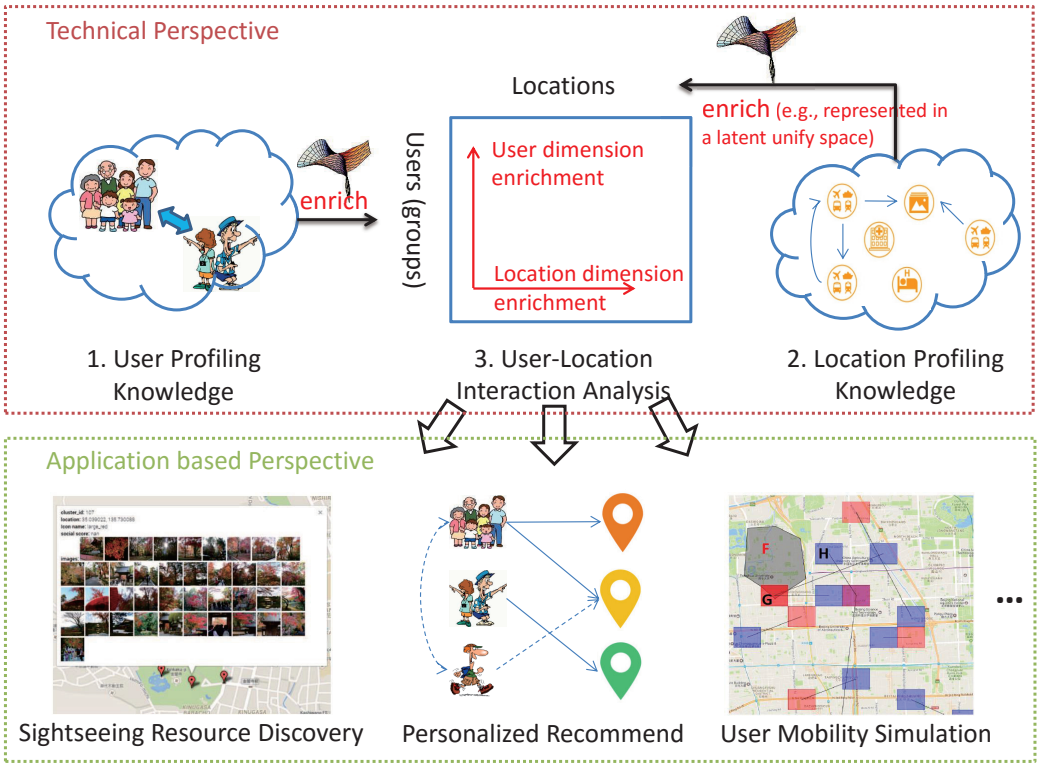


Figure 1.5. Overview of our research.

1.2.2 Major Contribution

In this subsection, we will introduce the major contribution of our work from both (1) technical view and (2) application based view. Figure 1.5 shows the overview of our contribution. Correspondingly, we first describe the technical contribution, followed by the application contribution. Furthermore, finally we will show a benign cycle to indicate that the work we have done did improve our daily lives.

From a technical perspective. In one word, we have made the following technical contribution:

1. As shown in Figure 1.5, from a technical perspective, we proposed a framework of jointly considering the enrichment of both user and location dimensions for better user-location interaction analysis. Novel Matrix Factorization (MF) related techniques were proposed for the idea's implementation. In Chapter 2, we did a brief review about related MF techniques for reference.

Specifically, for the user and location dimensions, we first mined useful knowledge

for them, respectively (a detailed description about the knowledge that we have mined will be presented in Section 1.3). Accordingly, we called these two mining processes *User Profiling* and *Location Profiling*. After obtaining the knowledge from both user and location dimensions, on the basis of a particular application problem (e.g., sightseeing resource discovery, personalized recommendation, user mobility simulation, and so on), we devised novel generative algorithms to map the knowledge into a latent high dimensional space, in which both the user and location dimensions have a same metric. For example, the space could be a Hilbert space. Finally, in this unify space, the interactions between users and locations could be analyzed. We called the mapping and interaction analysis *User-Location Interaction Analysis* shown in Figure 1.5.

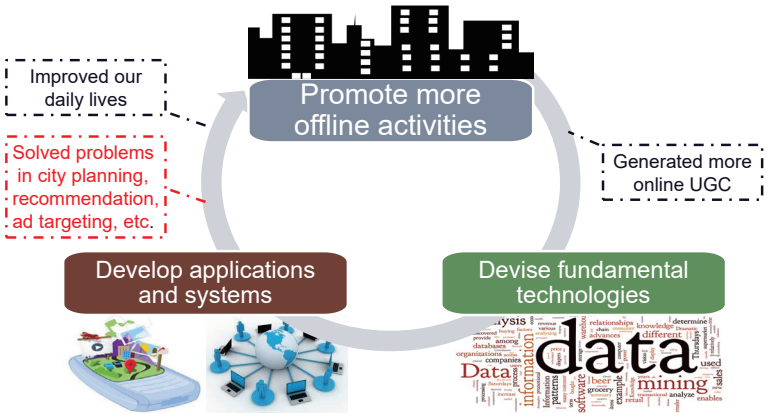
From an application based perspective. Furthermore, by utilizing the framework proposed above, we tried to solve real social problems. In one word, our current work have been applied in following social problems:

1. Our framework has been applied to discover both popular and obscure sightseeing resources in a city. On the basis of the survey [9], how to discover and recommend obscure (not so well-known) sightseeing resources is a big challenge. Our work is among the first of its kind to attack this challenge;

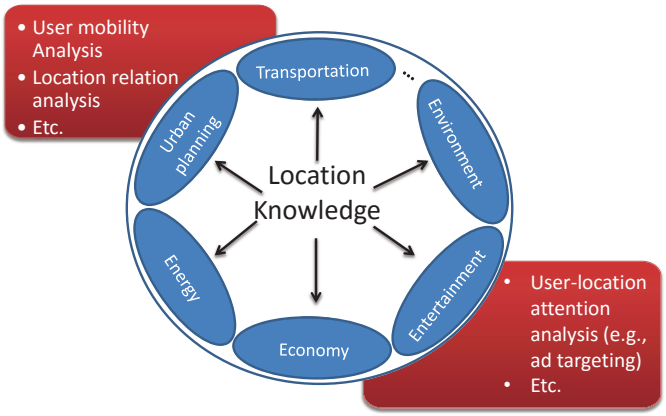
2. Our framework has been applied to learn user movement patterns in a city. Although a lot of related work have been proposed to analyze the user mobility [4] [10] [11], our work constructed a comprehensive user movement knowledge graph. On this graph, not only a single location's attributes but also the relations among different locations were learned. As a result, our work obtained better performance compared with the baselines.

3. Our framework has been applied to predict the extents of user attention to locations in a city. In essence, it is a collaborative filtering based matching process. That is, given an active user, the application aims first to find like-mined users and then based on these like-mined users' attention patterns, to calculate the prediction for the target active user. Benefiting from our proposed framework, since we can obtain more useful knowledge from *User Profiling* and *Location Profiling* processes, a much better matching result was obtained in our experiments.

To draw a brief conclusion about our major contribution, (1) from a technical standpoint, by mapping users' and locations' mined knowledge into a high dimensional space, we proposed a new framework of jointly considering both user and location



(a) A benign cycle



(b) Related problem domains

Figure 1.6. The impact of our work.

dimensions for better user-location interaction analysis; (2) from an application standpoint, by applying the proposed framework, we solved some new social problems as well as improved the performance of some already studied challenges.

A benign cycle. Figure 1.6a shows a benign cycle that we want to achieve. In this cycle, we mainly focus on the stage of devising fundamental technologies. Then, by using these novel technologies, various applications and systems would be developed for solving problems in different domains. As shown in Figure 1.6b, our work could be applied in city planning, transportation, economy, and so on. After improving users' experiences, more offline activities would be promoted. As a result, we could collect more UGC for improving the fundamental technologies. Therefore, we call the whole process a benign cycle.

1.3 Motivation and Major Tasks

Since detailed motivations will be respectively described in each chapter below, in this section, we briefly introduce why we want to do the research work presented in this dissertation followed by the corresponding major tasks that we have studied on in each of our work. To have a conclusion, four major tasks have been conducted:

1.3.1 User Profiling

User profiling is a problem of significant importance, the mined knowledge of which have already been successfully applied in different applications. For instance, major search engines, such as Google, Bing, and Yahoo, now incorporate user profiling related technologies to obtain better personalized recommendation results.

In our work, by observing the user mobility of different user groups in different cities (please see Figure 3.1 for details), we find that users from different groups have different location authority. Namely, there exits the information asymmetry about locations between different user groups. Therefore, our work propose a robust framework that includes five different methods to estimate users' location authority.

Location authority is a new concept, which is useful in different problem domains like geo-social recommendation, urban computing, ad targeting, and so on. To verify its utility, in our application, we use *location authority* to quantify the popularity of a location.

1.3.2 Single Location Profiling

In location profiling, we propose two dimensions, (1) user distribution based popularity and (2) quality, to describe a location. By doing so, we find that most of the conventional research work focus on discovering or recommending the locations that have high popularity and high quality. As shown in Figure 4.1, corresponding to the four quadrants, locations are classified into four clusters.

In our work, we focus on discovering and ranking obscure locations that have low popularity. Since the obscure locations often have very few user content generated there (i.e., lack of raw data), it is a big challenge for us to quantify the popularity and quality of them. Several novel and heuristic methods are proposed for this purpose.

For estimating the popularity, we utilize the users' location authority to do the calculation. For evaluating the quality, since "quality" contains too many aspects, we

propose two methods for analyzing the scenery quality of a location. In the experiments, verified by field visits, our methods do discover some obscure cherry blossom and maples locations in Kyoto. Most of them are worthy of visiting.

1.3.3 Location Relation Profiling

In the LBSNs related research area, the problems are addressed, in approximate terms, by considering three key relations: (1) user-location relation; (2) user-user relation; and (3) location-location relation. While the first user-location relation and the second user-user relation are well studied by related work, we find that the third location-location relation has not been well investigated yet.

In our work, we devise novel methods to construct a location knowledge graph to describe the relations among locations. By analyzing the GPS trajectory data from temporal, spatial, and temporalspatio views, two new unsupervised relation extraction methods are devised.

As a result, locations' functionality (e.g., for residence, education, shopping, etc.) can be represented as nodes on this knowledge graph, whereas the relations among locations are represented as edges. In the experiments, by visualization, we find that the movement patterns (lifestyles) can be detected on our knowledge graph.

1.3.4 User-to-Location Attention Prediction

As mentioned in Section 1.3.3, although the user-location relation has been well studied in related work, we find that few work has been done to simulate the user attention dynamics in a city. By seeing the statistics shown in Figure 6.1, we find that the user attention dynamics are exactly existing in a city. That is, the users from different groups pay attention to different locations in a city. For example, students always pay attention to the educational areas and shopping areas, whereas office workers pay more attention to the residential, working, and shopping areas. For office workers, there exists a strong movement pattern of *home* $\leftrightarrow$ *work*.

In our work, by utilizing the knowledge mined from location profiling, we propose two new probabilistic generative algorithms to simulate (predict) the user attention to locations in Beijing. The algorithms are first trained on the observed user-to-location attention matrix, and then used to predict the unobserved entries in this matrix.

In the experiments, by comparing with state-of-the-art baselines, our algorithms obtain the best performance with a big margin. Our work is among the first of its kind to

enrich the latent location features using a large amount of GPS trajectories. In other words, our algorithms organically combine the knowledge graph learned in Chapter 5 and the user-to-location attention matrix in Chapter 6 together. Furthermore, the algorithms are general ones that can be utilized in different domains.

1.4 Dissertation Structure *

The structure of this dissertation is as follows. In Chapter 2, a review of related work will be described. Then, Chapter 3 introduces the content about user profiling. By analyzing the various available UGC, users' location authority is estimated. In Chapter 4, on the basis of (1) user distribution based popularity and (2) scenery quality, a single location's characteristics are quantified. In addition, by applying the proposed methods, a demo is implemented to discover obscure sightseeing locations in Kyoto. In Chapter 5, the location relations are extracted by using two novel unsupervised Matrix Factorization (MF) methods. Accordingly, a location relation knowledge graph is constructed, which could be applied to a wide of problem domains. In Chapter 6, by utilizing the knowledge obtained from location profiling, two new algorithms are proposed to predict user attention to locations in a city. Chapter 7 concludes the whole dissertation.

*In some chapter titles of this thesis (i.e., Chapters 3, 4 and 5), for formal unity, we used the word "Profiling" to denote multiple attributes mining. Namely, in Chapters 3, 4 and 5, users', locations' and location relations' different attributes are mined, respectively.

CHAPTER 2

RELATED WORK

In this chapter, we first introduce the two major related research domains: (1) computational social science and (2) urban computing. From an application standpoint of solving social problems, since our work can also be regarded as a part of these two big research areas, we briefly present some related work that have been done in them.

Since our major technical contribution is to devise novel Matrix Factorization (MF) based generative algorithms, in this chapter, we further make a brief review of MF. One purpose is that the review could make readers quickly understand the evolution of the MF related technologies. The other purpose is to emphasize the differences between our proposed novel methodologies and these state-of-the-art MF technologies.

In this chapter, since we will introduce the related work from a high perspective, the detailed comparisons between our work and related work will be discussed by each of our tasks in later Chapters 3, 4, 5, and 6.

2.1 Computational Social Science

On the basis of a survey [12], computational social science is an interdisciplinary field at the intersection of the social science and computational science. It includes many research topics, such as information extraction, social network analysis, social geographic information system (GIS), social simulation models, and so on. Related to our work, we choose some of them for discussion.

Social Network Analysis. The foundations of morden social network analysis are found in the mathematical theory of graphs, like [13]. Being specific to our related problems, a social network consists of a set of nodes (users) and a set of edges (e.g., friendship relations), each defined by a set of attributes. By observing such a network, a lot of work have been proposed for different purposes.

For instance, in [14], the social network analysis was used to improve the recommendation performance. Their approach assumed a social network among users and makes recommendations for a user based on the ratings of the users that have direct or indirect social relations with the given user. As one of the major benefits, their social network analysis based approach could reduce the problems with cold start users. In [15], the authors found that the geography and online social relationships are inextricably intertwined. That is, the people we interact with on a daily basis almost always live near us. On the basis of this observation, the authros in [16] proposed a location propagation algorithm to predict users' residence locations based on users' social relationships. In our task introduced in Section 3.3.1, by devising a novel location authority propagation algorithm, we tried to estimate whether a user is familiar or unfamiliar with a location.

Information Extraction. Related to our work of discovering location knowledge, several related work have been conducted to mine the knowledge about a location's functionalities and quality.

To extract the functionality information of a location, in [17], the authors proposed a supervised multi-output regression method to estimate the functional composition of urban spaces. In [18], the authors proposed a topic-based inference model, which regards a location as a document, a function as a topic, to discover different functionalities. Unlike these related work of analyzing the semantic information (e.g., the categories of locations), by only observing GPS trajectories, we devised novel algorithms for functionality extraction in Section 5.4.

To evaluate the quality of a location, a lot of related work have been proposed for famous or hot locations [19]. For instance, in [3], the authors proposed a joint authority analysis framework to discover areas of interest (i.e., popular locations) with geo-tagged images and check-ins. In [20], the authors modeled the relationships of scene/landmark and scene/authorship as a graph and adopted two popular link analysis methods, PageRank and HITS, to mine representative landmarks. However, a problem of evaluating the quality of a not well known location has not been well investigated yet.

In our work, we will also evaluate the quality of a location. Unlike the related work introduced in [19] of ranking popular locations, in Section 4.5, we proposed two new methods to rank the obscure locations that have fewer raw data for analysis. In addition to the location functionality, in Section 5.4, we further extracted the relations (i.e., moving patterns) among different locations in a city.

Social Simulation Models. In this subarea, a number of related studies have been conducted for modeling the user mobility behavior. Unlike the related work that always focus on analyzing the individual user mobility behavior, in Sections 6.3 and 6.4, we proposed two robust methods that were able to model both individual and collective user mobility behavior. Furthermore, we further used the user mobility knowledge to predict the extents of user attention to different locations in a city. For reference, we list the following related work.

To model the individual mobility behavior, several related work have been conducted. For instance, in [21], a Hidden Markov Model (HMM) was proposed; in [11], a Conditional Random Field (CRF) was proposed. Recently, several studies have focused on predicting the places where users will go next, on the basis of their previously generated content. In [22], the authors explored the patterns of user check-ins and built a predictive model for user check-in behavior. They found that friends of users tend to go to similar locations, and that the users' visits follow a power law distribution. This means that users tend to visit only few places multiple times and many places only a few times. In [4], the authors developed a periodic and social mobility model. Their model is not designed to detect the home location of Twitter users, but rather their mobility patterns, e.g., when the user is at home and when the user is at work.

2.2 Urban Computing

Since the "Locations" defined in our work are mainly located in a city, we also would like to introduce some related work in the area of urban computing. On the basis of a survey [8], we introduce the following research topics.

Urban Planning. Traditionally, urban planners relied on labor-intensive surveys to inform their decision making in urban planning. For example, to understand urban commuting patterns, a series of research studies have been done based on travel survey data [23] [24]. Recently, benefiting from the widely available human mobility data generated in urban spaces, some modern models have been devised to automatically solve planning problems. The development of a city gradually fosters different func-

tional regions, such as educational areas and business districts, which support different needs of people's urban lives. The understanding of functional regions can calibrate urban planning.

For instance, in [18], the authors proposed a framework, DRoF, to discover regions of different functions in a city using human mobility between regions and POIs located in a region. In [25], the authors utilized call detail records to infer the land use of a region based on the dynamic phone activity patterns in the target region.

Urban Environment. Since we have witnessed an increasing trend of pollution in different aspects of the environment, recently, a number of researchers have focused on improving the air quality and reducing noise pollution.

For instance, in [26], the authors presented a vehicular-based approach for measuring fine-grained air quality in real time. They devised a mobile device consisting of a GPS receiver, CO sensor, and cellular modem. Installing such a device on multiple vehicles, they would be able to monitor the concentration of CO throughout a city. Different from the above solution, in [27], the authors inferred the real-time and fine-grained air quality information throughout a city based on the historical and real-time air quality data reported by existing monitor stations. In addition, a variety of data sources observed in a city, such as traffic flow, human mobility, structure of road networks, and POIs, were also taken into consideration for the inference.

Since exposing to high levels of noise pollution could cause serious illness ranging from hearing impairment to negatively influencing productivity, a number of countries, such as the United States, the United Kingdom, and Germany, have started monitoring noise pollution. As a result, a number of studies have been conducted to reduce the noise pollution, such as [28] [29] [30]. Since our work does not have much relationship with this research topic, we omit the detailed discussion.

Social Applications. Similar to our work, in urban computing, a lot of work have been conducted by analyzing the Location Based Social Networks (LBSNs). For example, the research topics include: estimating user similarity [31]; finding local experts in a region [32]; recommending locations [33]; understanding life patterns and styles [34] [35]; and so on. Therefore, we choose some work related to ours for discussion.

In [32], the authors were aiming to identify the local experts who have richer knowledge about a region than others. Since the experts' travel experiences are more accountable and valuable, they used them for travel recommendations. By analyzing the geo-tagged tweets, photos, and check-ins, a number of studies have also been conducted to understand a city's dynamics [34] and user behavior patterns in a city [35].

Above, we have listed some particular related studies in urban computing, which were aiming to solve the engendered challenges along with the rapid urbanization. Similar to these work, our work can also be applied into various problem domains, such as geo-social recommendation (e.g., sightseeing location recommendation in Section 4.7), urban planning (e.g., user attention prediction in Chapter 6), personal digital assistant (e.g., user mobility analysis in Chapter 5), and so on. By devising much more robust and efficient models (e.g., the novel idea of joint consideration of both user and location knowledge), our work obtained better performance in solving such social problems. In addition to the application based related work, in the next, we will further discuss the major related work from the view of technique.

2.3 Matrix Factorization

Among MF techniques, nonnegative matrix factorization (NMF) has received considerable attention due to its successful application to many recommender systems. The conventional NMF methods use ratings given to items by users as the sole source of information for learning to make recommendations.

Nonnegative Matrix Factorization. Given a user-item matrix $X \in \mathbb{R}^{M\times N}$, each column of X records an item's ratings given by different users. NMF aims to find two nonnegative matrices $U = [u_{ik}] \in \mathbb{R}^{M\times K}$ and $V = [v_{jk}] \in \mathbb{R}^{N\times K}$ whose product can well approximate the original matrix X. A commonly used cost function, the sum-of-squared-errors objective function, is presented as follows.

$$min_{U\geq 0, V\geq 0} \sum_{i,j} (x_{ij} - u_i v_j^T)^2 + \lambda_u \sum_{i=1}^{M} \|u_i\|^2 + \lambda_v \sum_{j=1}^{N} \|v_j\|^2, \tag{2.1}$$

where $\|\cdot\|^2$ denotes the Frobenius norm.

Probabilistic Matrix Factorization. Furthermore, researchers have found that the MF related problems can also be solved in a probabilistic manner. For instance, in [36] authors propose a probabilistic matrix factorization model. Algorithm 6 shows its generative process and all the notations are explained in Table 6.1. In accordance with the results in [36], maximizing the log-posterior over item and user features with hyperparameters (i.e., λ_u and λ_v in Algorithm 6) kept fixed is equivalent to minimizing the objective function Eq. 2.1.

Extended Probabilistic Matrix Factorizations. Recently, the above generative process has been further improved using item content information, i.e., a hybrid model.

Algorithm 1 The generative process for MF.

1: **for** each user $\vec{u}_i \in U$ **do** ▷ Stage 1
2: draw user group latent vector $\vec{u}_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$
3: **end for**
4: **for** each item $\vec{v}_j \in V$ **do** ▷ **Stage 2**
5: draw item latent vector $\vec{v}_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$
6: **end for**
7: **for** each user-item pair $(\vec{u}_i, \vec{v}_j) \in X$ **do** ▷ Stage 3
8: draw the rating $x_{i,j} \sim \mathcal{N}(\vec{u}_i^T \vec{v}_j, c_{i,j}^{-1})$
9: **end for**

Most of these studies focus on Stage 2 in Algorithm 6. That is, by analyzing the auxiliary content information, they aim to learn the more effective latent representation V. For example, in [37], Stage 2 is defined as:

$$\vec{v}_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K) + Dirichlet(\alpha). \tag{2.2}$$

Given words in scientific articles, this collaborative topic regression model combines traditional collaborative filtering with topic modeling. In [38], the authors define Stage 2 as:

$$\vec{v}_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K) + \mathcal{N}(X_{\frac{L}{2}, j*}, \lambda_n^{-1} I_K). \tag{2.3}$$

By utilizing the deep learning model, i.e., Stacked Denoising Autoencoders (SDAE) [39], they expect to enrich latent representation to boost the performance. By viewing the second entry in Eq. 2.3, the output of middle layer of the SDAE (i.e., $X_{\frac{L}{2}, j*}$) is used for enrichment.

Collective Matrix Factorization. Collective Matrix Factorization (CMF) [40] defined a high-level framework, by which two or more matrices could be jointly factorized. By introducing the Bregman matrix factorization [41], the loss function L of factorizing two matrices X and Y is defined as:

$$\begin{aligned} L_1(U,V|W) &= \mathbb{D}_{F_1}(UV^T||X,W) + \mathbb{D}_G(0||U) + \mathbb{D}_H(0||V), \\ L_2(V,Z|\tilde{W}) &= \mathbb{D}_{F_2}(VZ^T||Y,\tilde{W}) + \mathbb{D}_H(0||V) + \mathbb{D}_I(0||Z), \\ L(U,V,Z|W,\tilde{W}) &= \alpha L_1(U,V|W) + (1-\alpha) L_2(V,Z|\tilde{W}), \end{aligned} \tag{2.4}$$

where $\mathbb{D}_*$ are Bregman divergences; W and $\tilde{W}$ weight the observed X and Y, respectively; and $\alpha \in [0,1]$ weights the relative importance of X and Y. The F_1-divergence

tends to pull UV^T towards X, while the G- and H-divergences favor small U and V. Since CMF is a high-level framework, it is difficult to directly apply CMF for solving a particular problem. Mentioned in [41], the choice of Bregman divergences (e.g., squared euclidean distance, KL-divergence, Itakura-Saito Dist, etc.) is where we should inject domain knowledge. As a result, several instances of CMF have been developed in different problem domains [42] [43].

Instance of Collective Matrix Factorization. The Graph Regularized Nonnegative Matrix Factorization (GNMF) proposed in [42] is one of them. Instead of penalizing the norms of U and V (i.e., the second and third entries in Eq. 2.1), the authors introduce a geometrically based regularizer on V. By incorporating the nearest neighbor graph $W \in \mathbb{R}^{N\times N}$, the objective function of GNMF is as follows:

$$min_{U,V} \sum_{i,j} (x_{ij} - u_i v_j^T)^2 + \frac{\lambda}{2} \sum_{j=1}^{N} \sum_{l=1}^{N} \left\| v_j - v_l \right\|^2 W_{jl}. \tag{2.5}$$

By minimizing Eq. 2.5, they expect that if two items are close (i.e., W_{jl} is big), the latent representations of these two items, v_j and v_l, are also close to each other. We find that Eq. 2.5 can be regarded as jointly factorizing two matrices, $X \approx f(UV^T)$ and $W \approx g(VV^T)$, in a context of the assumption defined above.

GeoMF. Similar to our application problem, GeoMF [2] is a state-of-the-art method for POI recommendation. The authors argument users' and POIs' latent factors (i.e., P and Q in Eq. 2.6) in the factorization model with activity area vectors of users (i.e., X in Eq. 2.6) and influence area vectors of POIs (i.e., Y in Eq. 2.6), respectively. By capturing the spatial information (i.e., X and Y), GeoMF can deal with the challenge from the sparsity of observed matrix R. We compare our method with GeoMF in experiment. A detailed discussion is presented in Chapter 6.

$$min_{P,Q,X} ||W \odot (R - PQ^T - XY^T)||_F^2 + \gamma(||P||_F^2 + ||Q||_F^2) + \lambda ||X||_1. \tag{2.6}$$

CHAPTER 3

USER PROFILING *

Unlike the conventional user profiling problems mentioned in the related work section 3.2, our novel work is among the first of its kind to analyze users' location authority. That is, we are going to estimate user's familiarity with a given city. By analyzing different kinds of raw data (i.e., social network, time information, geographical information), we totally proposed three different methods for the location authority estimation. Furthermore, two ensembles of these three methods are devised, which obtained better experimental results. Regarded as a preliminary work, the user profiling will also help to mine location's attributes introduced in Chapter 4.

3.1 Motivation

The explosion of social networking services (SNSs), such as Twitter, Facebook, and Flickr, has led to a wealth of research into using social media content and various social graphs. For example, major search engines, such as Google, Bing, and Yahoo, now incorporate user-generated content (UGC) and trend analysis in their results. Moreover, in order to meet the diverse requirements of users, a significant effort has been dedicated to developing new applications.

In this context, one problem of significant interest is that of *location-authority-based*

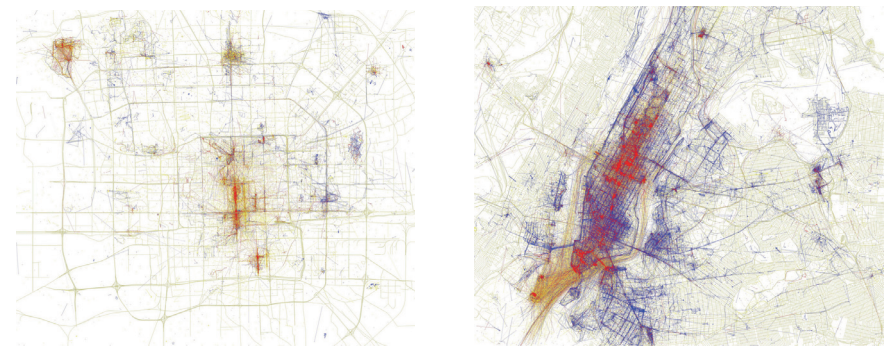
*Copyright @ 2017 MTA. The final publication of the content in this chapter is available at http://link.springer.com/article/10.1007/s11042-016-4034-6

user profiling—i.e., mining the information asymmetry regarding a location among various online user groups. That is, the "location authority" measures a user's familiarity with a particular location. In our work, by analyzing different kinds of user behavior that are the results of information asymmetry, location authority can be measured. These vital user models can help in different application scenarios, including the following:

Geo-social recommendations: according to a survey by Zheng et al. [9], the growing geo-referenced and community-contributed media resources have generated considerable amounts of detailed location and event tags, covering not only famous landmarks but also obscure locations. Although obscure locations are less well known, they are often still worth visiting, and they are a valuable sightseeing resource for development and promotion. Recently, several studies [44, 45, 20, 3] have shown the potential of UGC mining for discovering popular or famous locations and landmarks. Nevertheless, this area of research is relatively unexplored. Because the existence of obscure locations is discovered from the information asymmetry among different user groups, users' location-authority information can be used to discover them. For example, some beautiful locations may be popular with residents but relatively unknown to tourists who are unfamiliar with the city.

User profiling: a number of algorithms have been proposed for detecting the different attributes of users. For instance, conventional methods for location inference have been proposed from a wide variety of social media, such as Tweets on Twitter [46], geo-referenced pages on Wikipedia [47], image tags on Flickr [48], and social-network structures on Facebook [15]. Unlike these related methods, location authority focuses on estimating the authority of a user regarding some particular topic.

An intuitive example of this is shown in Figure 4.1. By analyzing the time information in images, Fischer presented the city-wide mobility behavior of Flickr users (`www.flickr.com/photos/walkingsf/sets/72157624209158632/`). As seen in these four cities, locals have significantly different mobility behaviors when taking photographs compared to tourists. Further analysis shows that the reason for this is the information asymmetry between these user groups. In this example, those familiar with the city (i.e., locals) have more choices for sightseeing (e.g., enjoying cherry blossom). That is, they are the authoritative users of the target city. By analyzing image content, we can further detect the topics with which they are familiar. However, visitors are sometimes more authoritative than residents, and vice versa. For example, some locations are better known to professional photographers than they are to resi-



(a) Beijing (b) New York

(c) Kyoto (d) San Francisco

Figure 3.1. Mobility behavior of different user groups. Red: tourists; blue: locals; yellow: either.

dents. Accordingly, location-authority analysis cannot simply depend on the location information specified in online user profiles. Moreover, according to our statistics, most users do not disclose information regarding their residency.

Therefore, in this work, we mainly focus on the location authority based user-classification task. By analyzing geo-tagged UGC, three complementary models are proposed. First, we introduce a social-network-driven model, where a graph-based method is used to determine whether a user is living in the target city. Second, we introduce a time-driven model, which estimates the user's location authority based on the frequency of visits. Third, considering the fact that different users have different mobility behaviors, a location-driven model is devised. Furthermore, in response to the shortages of each model, two ensemble methods are discussed. Finally, utilizing these methods, an application is implemented to discover points of interest (POIs) that

are not well known, even though they are worth visiting. To the best of our knowledge, this is the first attempt at discovering such POIs.

In summary, we make the following major contributions in this work:

- From a theoretical standpoint, we devise a comprehensive solution to classify SNS users on the basis of their location authority. Thus, the influence of the social network and the temporal and spatial influence from geo-tagged UGC are analyzed. We further propose two ensembles of these models, in which each user's available information is dynamically considered in order to choose models with higher discriminative ability.

- We evaluate the proposed methods with experiments on a real-world dataset collected from three cities on Flickr.

The remainder of this chapter is organized as follows: Section 3.2 places our research in the context of previous related work; Section 3.3 presents the proposed models and their ensemble methods; Section 3.4 outlines our experiments and presents an in-depth quantitative and qualitative analysis of the obtained results.

3.2 Related Work

In this section, we discuss some research related to user-attribute detection.

A number of algorithms have been proposed for estimating the home location of Twitter users using content analysis. For instance, Eisenstein et al. [49] built geographic topic models to predict the locations of users in terms of regions and states. Using a Gaussian Mixture Model and Maximum Likelihood Estimation, Chang et al. [46] proposed a city-level location-estimation method. Estimating at city level is more challenging than estimations at higher granularities, such as states or countries. In addition to the analysis of Tweets, other methods for inferring location have been proposed based on social media, including geo-referenced pages on Wikipedia [47], image tags on Flickr [48], and social network structures on Facebook [15].

Recently, several studies have focused on predicting the places where users will go next, based on their previously generated content. For instance, Gao et al. [22] explored the patterns of user check-ins and built a predictive model for user check-in behavior. They found that friends of users tend to go to similar locations, and that the users' visits follow a power-law distribution. This means that users tend to visit only a

few places multiple times and many places only a few times. Cho et al. [4] developed a periodic and social mobility model. Their model is not designed to detect the home location of Twitter users, but rather their mobility patterns—e.g., when the user is at "home" and when the user is at "work."

In addition to location information, a variety of other user attributes have been investigated using natural language processing, including gender, age, and even political orientation [50, 51]. The problem of automatically constructing user profiles has been commonly regarded as important. Nevertheless, to the best of our knowledge, estimating a user's location authority by analyzing online generated social media has not yet been thoroughly investigated.

3.3 Users' Location-authority Analysis

In this section, we describe in detail three types of information that can help to characterize a user's location authority: *social network*, *visiting frequency* and *mobility behavior*. In accordance with these three types of information, three models (i.e., social-network-driven, time-driven, and location-driven models) are introduced in order to estimate SNS users' authority with a particular city. Our goal is two-fold: first, to provide an assessment of the robustness and generalization potential of features for authority-based user-classification purposes; and second, to explore methods of combining these three models for higher classification performance. We first describe each individual model before discussing their combination.

3.3.1 Social Network: "Is the user living in the target city?"

For target city c, the most direct way of identifying a user's authority is to retrieve residency information from the user's profile. However, few users are willing to disclose this information or even their country of residence. In accordance with the statistics from our experimental dataset, 64.8% of the 1,827,500 Flickr users did not specify any information regarding their residency. Moreover, although the other 35.2% users did provide location information, they were not necessarily familiar with the location specified. A reasonable explanation for this exception is that users commonly provide their hometown rather than their current location.

Therefore, we propose a social-network-based method for calculating the authority of a user with given city c, by which "authoritative communities" can be detected on a

network. The assumption for this model is that a user with many friends who are familiar with city c will also be familiar with c. The statistics compiled in [15, 16] confirm that geographic and online social relationships are inextricably intertwined. Generally, the likelihood of friendship is inversely proportional to distance. Unlike the related studies [15, 16] that investigated pairwise friendship between two users using a maximum likelihood approach, our algorithm further aggregates collective relationships by information propagation on the social network. Before introducing the algorithm, we first present the definition of a social network.

Definition 1. *A social network N is a directed graph $N(V,E)$, where V is the user set of v_i, and E is the friendship set of $e(v_i,v_j)$ from v_i to v_j. In our implementation, the network is represented as a transition matrix:*

$$N(v_i,v_j) = \begin{cases} 0 & if(v_j,v_i) \notin E, \\ 1/outdegree(v_j) & if(v_j,v_i) \in E. \end{cases} \tag{3.1}$$

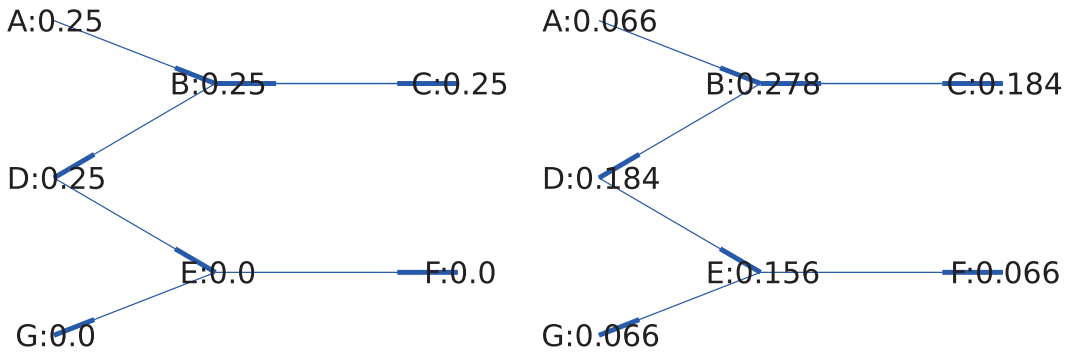
We then devise a biased authority-propagation algorithm, Algorithm 2, to model this.

Algorithm 2 Biased Authority-propagation Algorithm.

1: calculate d_c ▷ based on whether or not v_i is a local user
2: $M_B = 0; \alpha_B = 0.85$
3: $F_s(c) = d_c$ ▷ initialize $F_s(c)$
4: **while** $M_B < 1000$ *and* $F_s(c)$ *not converged* **do**
5: $\quad F_s(c) = \alpha_B \cdot N \cdot F_s(c) + (1-\alpha_B)d_c.$
6: $\quad M_B++$
7: **end while**

By iterating the biased algorithm on network N, the authority value of a parent node is split among its children and the authority value of a child node is the sum of the authority values propagated over its links. This is exactly consistent with our assumption.

For the normalized bias distribution d_c in Algorithm 2, in contrast to TrustRank [52], our method can automatically select good seeds based on whether user v_i is a local of city c, as detected from the user's profile. The entries for vector d_c that correspond to good seeds sum up to 1. Given city c, $F_s(c)$ is initialized to d_c before the iteration. Here, α_B is the decay factor. Figure 3.2 illustrates an example when nodes [A, B, C,



(a) Before iteration

(b) After iteration

Figure 3.2. Social-network-driven method. Each node represents a user. Edges denote a friendship between two users. The scores measure the authority levels.

D] are good seeds. After the iteration, we find that node B has obtained the highest score because it was recommended by more good seeds.

Because of the exception above (note that a user may not be familiar with the location specified in their profile), some nodes can be wrongly selected as good seeds in d_c. Fortunately, our method can significantly reduce this impact. Our iterative algorithm not only aggregates the "authoritative communities" (e.g., [B, C, D, E] in Figure 3.2), it also filters out the noise in d_c (e.g., [A]).

Although this model is effective in most cases, it cannot cover situations where potentially good seeds are sparsely distributed (e.g., [F, G]). In other words, if there are many users with few friends on the social network, the sensitivity (or "recall") of this model will be low. We discuss this issue in detail when evaluating the proposed methods.

3.3.2 Visiting Frequency: "Does the user visit the target city?"

The second method is a time-driven method. This method considers the frequency of visits made by users. Intuitively, if a user has visited target city c frequently and recently, the user is familiar with c. First, for each user v_i, we define matrix M_i to record the user's visitation history.

Definition 2. *Matrix M_i is a mapping between user v_i and all the cities C_i that the user has been to. For example, retrieving v_i's geo-tagged images, $M_i[y = 2014, c = kyoto] = 10, c \in C_i$ means that among v_i's images taken in Kyoto in 2014, there are 10 different*

dates detected.

On the basis of matrix M_i, we calculate the user's authority with each city $c \in C_i$. To do so, three factors must be considered:

1. For each year, the proportion of days each c accounts for is recorded in M_i;
2. The revisitation to cities is denoted as matrix r_i. Assuming that we collect 10 years' worth of raw data for v_i, for each year x, $r_i[x,c] = \frac{5}{10}$ when we detect that v_i went to c during five of those years;
3. The staleness of information. To identify whether v_i has been to these cities recently, we introduce diagonal matrix $\omega_i(x,y)\ \forall x,y \in \{1,2,...,n\}$, to characterize this feature. Here, n denotes the number of detected years.

The time-driven method is then used to compute authority score F_t as follows:

$$F_t(v_i,c) = diag(r_i^T \cdot (w_i \cdot M_i))[c], \tag{3.2}$$

where *diag* denotes the matrix's diagonal vector and $[c]$ is used to obtain the authority score of target city c.

Obviously, the accuracy of user v_i's F_t depends to a considerable extent on the sparsity of matrix M_i. For a user who has uploaded few geo-tagged UGC (e.g., Tweets, images, check-ins), this method will be ineffective.

3.3.3 Mobility Behavior: "Where do users visit in a target city?"

In addition to the social network and time information introduced above, the geographic information regarding where a user has been in city c is also essential for estimating location authority. Furthermore, unlike our previous two models and related approaches [46] [15] [4], which focus on individual behavior, this third model simulates collective mobility behavior. By introducing latent variable *group* g this model provides a more general solution to simulate the mobility behavior of different user group gs. In the next part, we first describe the modeling process followed by the model's parameter inference.

Mobility Behavior Modeling

Table 3.1 shows the notations used for this model. To generate the map shown in Figure 3.1, Algorithm 3 uses a generative process. By introducing Bayes' theorem, our algorithm models users' mobility behavior based on the following observations.

1. **Observation 1:** For each user v_i, given city c, there is an inherent property that describes their authority with the target city. For instance, there are three different groups, labeled red, blue, and yellow in Figure 3.1, which represent the different mobility behavior of these groups. Accordingly, with our location-driven model, we introduce latent variable *group*, denoted as g, to represent this property.

2. **Observation 2:** A region r (see Definition 3, below) is visited by groups g with mixture distributions. That is, users from different groups prefer visiting different regions of a city. In Figure 3.1, it is clear that some regions appeal to more users from the *red group*, while others attract many users from the *blue group*. In Algorithm 3, this observation corresponds to the generative process from lines 5 to 12.

3. **Observation 3:** A group g is a mixture distribution of users v. Similar to the concept of a fuzzy set, given group g, each v_i has a probability that indicates their degree of membership to g. In Algorithm 3, the generative process from lines 2 to 4 represents this observation.

Definition 3. *Region R is a collection of regions clustered by Gaussian Mixture Model (GMM), as shown in line 1 of Algorithm 3. GMM is used because it is widely applied to simulate users' mobility behavior [46] [4]. Another, more important reason for applying a GMM is to have a unified parameter inference process in Section 3.3.3 that operates in a probabilistic manner.*

The whole generative process in Algorithm 3 can be explained using the following joint distribution:

$$p(t,g \mid \vec{\alpha},\vec{\beta},\Lambda) = p(l_t \mid r,\Lambda)p(v_t \mid g,\vec{\alpha})p(g \mid \vec{\beta}). \tag{3.3}$$

Given priors $\vec{\alpha},\vec{\beta}$, and Λ, by introducing latent variable *group* (g), the model aims to generate each tuple $t = \{v_t, l_t\}$ (e.g., t represents an image, Tweet, or post) in city c. Here, Λ denotes all the priors of the GMM. More specifically, there is a Dirichlet distribution over the mixing coefficients π_r, and a Gaussian–Wishart distribution governing the mean μ_r and precision Σ_r^{-1} of each Gaussian component.

Parameter Learning

In accordance with our goal of identifying user authority (i.e., the probability a user belongs to group g), the target of model inference is distribution $p(g \mid t)$, where the hy-

Table 3.1. Notations used for the location-driven model.

	Size	Description
V	$[1,V]$	User set, where v is a particular user in V
R	$[1,R]$	Region set, where r is a region in R
G	$[1,G]$	*Group* set, where g is a group in G
T	$[1,T]$	$t \in T$ is a user-coordinate tuple: $\{v_t, l_t\}$
μ_r	$\mathbb{R}^2$	Mean location of latent region r
Σ_r	$\mathbb{R}^{2\times 2}$	Covariance matrix of latent region r
$\vec{\vartheta}_r$	$1 \times \lvert G\rvert$	Region-dependent group distribution
$\vec{\varphi}_g$	$1 \times \lvert V\rvert$	Group-dependent user distribution
$\vec{\alpha}$, $\vec{\beta}$		Hyperparameters of Dirichlet priors
Ω	$\lvert R\rvert \times \lvert G\rvert$	$\Omega_{r,g}$: count of tuples in r that are assigned to g
Ψ	$\lvert G\rvert \times \lvert V\rvert$	$\Psi_{g,v}$: count of times when v is assigned to g
$\mathcal{N}$		Normal distribution
Dir		Dirichlet distribution
$Mult$		Multinomial distribution

Algorithm 3 Probabilistic generative process.

1: $R \sim \sum_{r=1}^{R} \pi_r \mathcal{N}(x \mid \mu_r, \Sigma_r)$ ▷ initialize R using GMM
2: **for** all groups $g \in [1,G]$ **do** ▷ Observation 3
3: sample mixture components $\vec{\varphi}_g \sim Dir(\vec{\alpha})$
4: **end for**
5: **for** all regions $r \in [1,R]$ **do** ▷ Observation 2
6: sample mixture proportion $\vec{\vartheta}_r \sim Dir(\vec{\beta})$
7: **for** all tuples $t \in [1,N_r]$ in region r **do**
8: sample a coordinate $l_t \sim \mathcal{N}(\mu_r, \Sigma_r)$
9: sample a group $g_{rt} \sim Mult(\vec{\vartheta}_r)$
10: sample a user $v_{rt} \sim Mult(\vec{\varphi_{g_{rt}}})$
11: **end for**
12: **end for**

perparameters are omitted. In Eq. 3.3, when region r is given at the initialization stage, g is independent ($\perp\!\!\!\perp$) of l_t. Intuitively, we can regard all tuples t in a specific region r as a *bag of instances of different users* by ignoring the location information. Note that, despite $g \perp\!\!\!\perp l_t \mid v_t, r$, our model is sufficiently flexible to describe a situation where different instances v_t of the same user v can be assigned to different gs. Accordingly, we divide the inference procedure into the following two steps.

Step 1. $l_t - generation$**:** Because city c is described by a GMM and each region r is a component of the GMM, we utilize a variational Bayesian machinery for the inference of the GMM. As mentioned above, because it is unnecessary to identify the exact location information given r, a random coordinate is assigned to each l_t in r.

Therefore, we simply utilize the implementation introduced in Section 10.2 of [53]. This offers the advantage that the number of mixture components (i.e., $|R|$) can be automatically identified with a relatively large initial value.

Step 2. $v_t - generation$**:** After eliminating the l_t, the joint distribution in Eq. 3.3 is simplified as follows:

$$p(t, g \mid \vec{\alpha}, \vec{\beta}) = p(v_t \mid g, \vec{\alpha}) p(g \mid \vec{\beta}). \tag{3.4}$$

Three strategies can be applied to solve our inferential problem: EM with variational inference, EM with expectation propagation, and Gibbs sampling. We selected the Gibbs sampling method because its performance is comparable with the other two, but it is more tolerant to local optimization.

We derive the Gibbs updating rule for Algorithm 4, i.e., a conditional distribution for a tuple with index $i = (r,t)$, as follows. Note that $T = \left\{t_i, T^{\neg i}\right\}$, $G = \left\{g_i = x, G^{\neg i}\right\}$, and the priors $\vec{\alpha}$ and $\vec{\beta}$ are omitted. Because $t_i \perp\!\!\!\perp T^{\neg i} \mid G^{\neg i}$, we have:

$$\begin{aligned} p(g_i = x \mid G^{\neg i}, T) &= \frac{p(G,T)}{p(G^{\neg i}, T)} \\ = \frac{p(T \mid G)}{p(T^{\neg i} \mid G^{\neg i}) p(t_i)} \cdot \frac{p(G)}{p(G^{\neg i})} &\propto \frac{p(G,T)}{p(G^{\neg i}, T^{\neg i})}. \end{aligned} \tag{3.5}$$

The joint distribution from Eq. 3.4 can be derived as:

$$\begin{aligned} p(G,T) &= p(T \mid G, \vec{\alpha}) p(G \mid \vec{\beta}) \\ &= \prod_{g \in G}^{G} \frac{B(\Psi_g + \alpha_g)}{B(\alpha_g)} \cdot \prod_{r \in R}^{R} \frac{B(\Omega_r + \beta_r)}{B(\beta_r)}, \\ &\text{where } B(\vec{x}) = \frac{\prod_{k=1}^{dim \vec{x}} \Gamma(x_k)}{\Gamma(\sum_{k=1}^{dim \vec{x}} x_k)}. \end{aligned} \tag{3.6}$$

Algorithm 4 Gibbs sampling algorithm for learning Eq. 3.4.

zero all count matrices Ω, Ψ; ▷ initialization
for all regions $r \in [1,R]$ **do**
  for all tuples $t \in [1,N_r]$ in region r **do**
    sample a group $g_{rt} \sim Mult(1/|G|)$ for t
    update region-group count: $\Omega[r,g_{rt}]+1$
    update group-user count: $\Psi[g_{rt},v_t]+1$
  end for
end for
while not finished **do** ▷ burn-in and sampling period
  for all regions $r \in [1,R]$ **do**
    for all tuples $t \in [1,N_r]$ in region r **do**
      exclude the target t: $\Omega[r,g_{rt}]-1$; $\Psi[g_{rt},v_t]-1$
      sample a new group $\tilde{g_{rt}}$, using Eq. 3.7
      update the counts: $\Omega[r,\tilde{g_{rt}}]+1$; $\Psi[\tilde{g_{rt}},v_t]+1$
    end for
  end for
  if converged and sampling iterations are finished **then**
    obtain the parameters $\vec{\vartheta}_r$ and $\vec{\varphi}_g$, using Eq. 3.8
  end if
end while

Here, we first assume symmetric Dirichlet priors, with α and β each having a single value. Then, by substituting and simplifying Eq. 3.5, we obtain the Gibbs updating rule:

$$\begin{aligned} p(g_i = x \mid G^{\neg i}, T) &= p(g_i = x \mid G^{\neg i}, t_i = v, T^{\neg i}) \\ &\propto \frac{\Psi_{x,v} + \alpha_v - 1}{\sum_{v' \in V}^{V} \Psi_{x,v'} + |V|\,\alpha_{v'} - 1} \cdot \frac{\Omega_{r,x} + \beta_x - 1}{\sum_{x' \in G}^{G} \Omega_{r,x'} + |G|\,\beta_{x'} - 1}. \end{aligned} \tag{3.7}$$

This result is intuitive: the first ratio expresses the probability of t_i under group g_i, and the second ratio expresses the probability of group g_i in the region r to which the target sampled t_i belongs.

Finally, with a set of samples from the obtained posterior distribution $p(g \mid t)$ (i.e., the converged Ω and Ψ), the multinomial parameters, $\vec{\vartheta}_r$ and $\vec{\varphi}_g$, can be computed as

follows:

$$\varphi_{g,v} = \frac{\Psi_{g,v} + \alpha_v}{\sum_{v' \in V}^{V} \Psi_{g,v'} + |V|\,\alpha_{v'}},$$
$$\vartheta_{r,g} = \frac{\Omega_{r,g} + \beta_g}{\sum_{g' \in G}^{G} \Omega_{r,g'} + |G|\,\beta_{g'}}. \tag{3.8}$$

These values correspond to the state of the Markov chain in Gibbs sampling. Insofar as they are conditioned by the past T and G, they are used to estimate the predictive distribution over new tuples and groups.

3.3.4 Model Integration

The most important reason for integrating the models is to identify latent groups generated by the location-driven model. Because only the user IDs and geo-coordinates are used, the system cannot automatically know the meanings of the groups when this model is exclusively used (i.e., it is unclear whether the group is familiar or unfamiliar with the city). Within this context, the other two models can be utilized to identify latent groups.

Another reason to integrate the models is that they use distinct kinds of raw data. Integrating them thus makes the classification task much more stable and robust. That is, a more general model is obtained for responding to different situations. To this end, we propose two ensembles: a *support vector machine (SVM)* and *dynamic weighted ensemble (DWE)*. These two ensembles enable our solution to utilize the results obtained from the location-driven model, and they enable us to improve its performance.

SVM: With this method, we simply use the scores obtained from the three models as features to train multiple classification models. For instance, if the number of groups $|G| = 2$ in the location-driven model, there will be at most four features for SVM model training. However, because the sum of the two scores from the location-driven model is 1.0, in practice three features are sufficient. In the experiments, combinations of these features, taken two at a time, were tested in order to detect key features.

For the implementation, we utilize SVM with the RBF kernel. The grid-search method introduced in [54] is then applied with three-fold cross-validation to find the best RBF kernel parameter γ and penalty parameter C in the SVMs. Finally, the most powerful model is selected by comparing the average classification accuracy.

DWE: In machine learning, multiple classifiers are always combined in an ensemble, and this is typically more accurate than using an individual model. The most standard

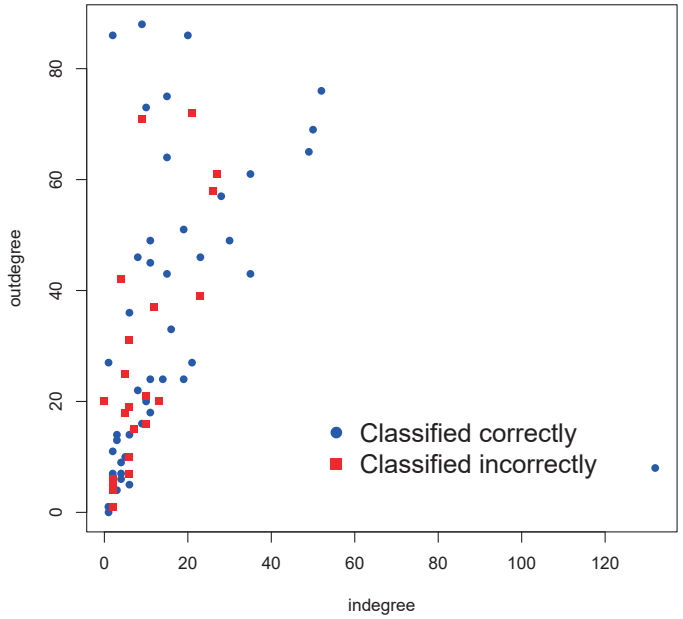


Figure 3.3. Indegree and outdegree distributions of 94 samples from the social network N (see Definition 1). Here, 47 blue samples are correctly identified, while the other 47 red samples are misidentified.

method for doing so is majority voting, where the final classification is the class that receives the most votes from the individual classifiers [55]. In addition, other methods have been proposed, such as bagging [56] and boosting [57].

Before introducing our ensemble strategy, we first discuss the shortcomings of each of the proposed models. For the first social-network-driven model, Figure 3.3 shows 94 nodes (users) in the friendship-based social network. Among them, 47 blue nodes are correctly classified, while the other 47 red nodes are misclassified. Obviously, there are differences in the available information ("indegree and outdegree" in this case) that can influence the model's performance. Intuitively, for users with few friends (i.e., lower indegree and outdegree), this model would, to a considerable extent, have insufficient discriminative power. The experiment results shown in Figures 3.6a and 3.6b further verify this observation. Its performance decreased sharply for the inactive users. Similarly, our experimental results in Figures 3.6c and 3.6d show that the second time-driven model's performance decreases sharply for inactive users who generate little content (i.e., images in our case) on the Internet. The performance of the third

location-driven model, as shown in Figures 3.6e and 3.6f, although not the highest, is the most stable. These observations suggest that we should take the different levels of available information for each user into consideration.

Therefore, in our solution, we use a DWE, the idea for which was first introduced in [58]. In contrast to other methods, this method can account for the differences in the available information for each user. For instance, some users may have more generated data (e.g., images, or Tweets), favoring the time-driven model, whereas other users may have a lot of friends, favoring the social-network-driven model. Moreover, by regarding as evidence users who obtained high weights in the first two models, the DWE can label the latent groups generated by the location-driven model as "familiar with the city" or "unfamiliar with the city."

Accordingly, using a user's indegree and outdegree as features, we train learning model P_s to predict the discriminative ability of the social-network-driven model. Similarly, using the amount of UGC as a feature, we train another model P_t for the time-driven model. AdaBoost [57] was used for learning, because it fits a sequence of weak learners. In our case, the learners (i.e., P_s and P_t) are decision trees with $max_depth = 3$.

$$\begin{aligned} F(v_i) &= [F_s, F_t, F_l] \cdot \vec{en}^T, \\ where\ \vec{en} &= [0.5P_t, 0.5P_s, 1 - 0.5(P_t + P_s)]. \end{aligned} \tag{3.9}$$

Eq. 3.9 is used to calculate the assembled authority score of user v_i. Here, F_s, F_t, and F_l are the authority scores or probabilities obtained from the three models, respectively. In order to maintain the same scale, they are first pre-normalized. Ensemble weights $\vec{en}$ are calculated using prediction results of the models P_s and P_t. Both P_s and P_t contain two predictions: "1" denotes high classification strength and "0" denotes low classification strength. For example, if neither the social-network-driven model nor the time-driven model have sufficient classification strength, the DWE only uses the stable location-driven model.

In summary, because the three original models are established using completely different information, we expect that their combination will improve the performance. The drawback to such ensembles, however, is that they require a training dataset. Although future research could be directed towards establishing a more comprehensive training dataset, we currently rely on manually labeled random samples. Moreover, in the next section we experimented with cross-validation in order to render the dataset statistically significant.

Table 3.2. Datasets collected from the three cities: Beijing, Kyoto, and San Francisco.

	Beijing	**Kyoto**	**San Francisco**
Number of geo-tagged images	4,002	6,121	3,989
Number of target Flickr users	230	300	536
Percentage of images from the top 20%	85.7%	84.4%	78.6%

Table 3.3. Contextual data used in Algorithm 2 and Eq. 3.2.

Algorithm 2:	Friendship-based directed social network N (see Definition 1). Nodes: 837,633; Edges: 3,082,408.
Eq. 3.2:	1,066 matrices M (see Definition 2) recording users' visitation history. Year range: 2005–2015; Cities: 17,460.

3.4 Experimental Evaluation

In this section, we outline the retrieved datasets and parameter selections for our experiments. All five user-classification methods (i.e., Algorithm 2, Eq. 3.2, Eq. 3.3, SVM, and DWE) are then compared and discussed quantitatively. Finally, some case studies are visualized in order to demonstrate the feasibility of our methods.

3.4.1 Data Preparation

Table 3.2 summarizes the target datasets for Beijing, Kyoto, and San Francisco, which were collected from Flickr. Unsurprisingly, the 80–20 rule persists on Flickr. That is, for all three datasets, the top 20% Flickr users were prolific, whereas the bottom 80% of the users remained relatively inactive.

Table 3.3 details the contextual data used by the social-network-driven model (Algorithm 2) and the time-driven model (Eq. 3.2). We collected a total of 837,633 Flickr users in order to construct social network N, in which all the 230 + 300 + 536 target users were included. Furthermore, we retrieved all of the images uploaded by the target users from Flickr. These included all of the geo-tagged images and some other images that were taken just before or after the geo-tagged images. The images were then used to create matrix M.

3.4.2 Parameter Selection

To make our models more reproducible, we list all the parameters and their corresponding values.

Social-network-driven Model: We applied our biased authority-propagation algorithm by setting the decay factor $\alpha_B = 0.85$, which is often considered to be the default value for PageRank-like calculations in the literature. The authors of [59] further found that when the decay factor changes, the top sections of the ranking changes only slightly. As we are especially interested in "authority communities," i.e. the top sections, the impact of the decay factor's choice is limited. We set the maximum number of iterations $M_B = 1,000$.

Time-driven Model: Because the heuristic parameter w_i in Eq. 3.2 is used to characterize the information staleness, on the basis of its actual effectiveness, we set it to exponential decay: $0.8^{(\lambda - x)}$. That is, for any year x, more time passing between x and current year λ leads to higher decay.

Location-driven Model: We set the number of groups $|G| = 2$. Based on the experimental results below, these two latent groups can be explained as a "tourist-like group" and "local-like group." In accordance with the conclusion in [60], because of the mutual relationship between hyperparameters and the group number, we set the symmetric Dirichlet priors as $\alpha = 0.01$ and $\beta = 50/|G|$. A relatively small value of α can be expected to result in a fine-grained decomposition of the users in region r into groups g that address familiarity. For β, by keeping constant the sum of the Dirichlet hyperparameters (50 in our case), it can be interpreted as the number of virtual samples contributing to the smoothing of the region-dependent group distribution ϑ. The maximum number of sampling iterations in Algorithm 4 was set to $1,000$.

3.4.3 User-classification Evaluation

ROC-based Evaluation: The receiver operating characteristic (ROC) curve is a graphical plot that illustrates the performance of a binary classifier model as its discrimination threshold is varied. In our case, although we attempted to classify users into two groups, we obtained a score or probability for each user rather than a label with two options (i.e., familiar or unfamiliar). Because the determination of an ideal threshold (also known as a cut-off value) is always a tradeoff between sensitivity (true positives) and specificity (true negatives), the ROC curve offers a graphical illustration of these tradeoffs at each threshold.

Table 3.4. Calculating the true-positive rate and the false-positive rate.

True class	Familiar (high authority)	Unfamiliar (low authority)
Predicted class		
Familiar	True Positives (TPs)	False Positives (FPs)
Unfamiliar	False Negatives (FNs)	True Negatives (TNs)
True–positive rate = TP/P;		
False–positive rate = FP/N.		

Table 3.5. Labeling results of the top 100 active users in each city, denoted by $Active_{set}$.

Cities	**Number of familiar users**	**Number of unfamiliar users**
Beijing	≈ 36	≈ 64
Kyoto	≈ 31	≈ 69
San Francisco	≈ 76	≈ 24

In our experiments, we used two criteria for the evaluation, as shown in Figure 3.6: sensitivity (true-positive rate) and 1 - specificity (false-positive rate). Table 3.4 shows how these these two criteria were calculated.

Using these two criteria, the area under the ROC curve (AUC) is recognized as a measurement of a test's discriminatory power. The maximum AUC value is 1.0, indicating 100% sensitivity and 100% specificity, and an AUC value of 0.5 indicates the absence of any discriminative power.

Ground Truth: It is almost impossible to acquire the exact ground truth regarding whether a user is familiar with a city. For example, a resident of Beijing is not necessarily familiar with the city. For our experiments, we employed manual efforts to collect an approximate ground truth. This was done by adopting the following procedures.

We invited nine subjects to manually score users' authority over a specific city from the users' online information. For each city (i.e., Beijing, Kyoto, and San Francisco), three independent subjects were assigned to label 200 users that had been selected. In order to obtain a comprehensive evaluation, we selected both users who had uploaded many images to Flickr and those who had not. Table 3.5 shows the labeling results for 300 active users, from which approximately 36, 31, and 76 familiar users were returned by majority voting. Similarly, Table 3.6 shows the labeling results for 300 inactive users.

Table 3.6. Labeling results of the bottom 100 inactive users in each city, denoted by $Inactive_{set}$.

Cities	Number of familiar users	Number of unfamiliar users
Beijing	≈ 44	≈ 56
Kyoto	≈ 15	≈ 85
San Francisco	≈ 59	≈ 41

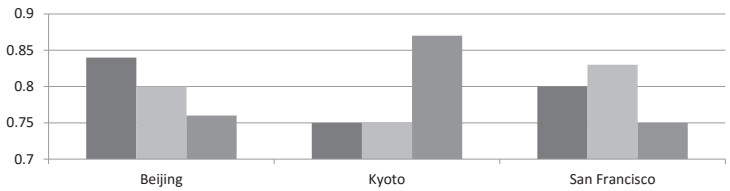


Figure 3.4. Labeling variances on the top 100 active users in each city, calculated using the Jaccard index.

To ensure that the labeling results were credible, all of the subjects were familiar with their corresponding cities, and they recorded the reasons for all of their labels. For each city, Figures 3.4 and 3.5 present the labeling variances between any two of the three subjects according to the Jaccard index. Indeed, all of the subjects provided high-quality outcomes, and their results were similar. Next, all of the models were analyzed using the ground truth from subjects' votes—i.e., by majority.

Experimental Results: The overall results for the location-authority estimation are reported in Figure 3.6. The results show that the proposed framework generally performed well. However, the results varied across models. The details are as follows.

The social-network-driven model (SNDM) fluctuated considerably. By comparing the ROC curves shown in Figures 3.6a and 3.6b, the decline in user activity had a neg-

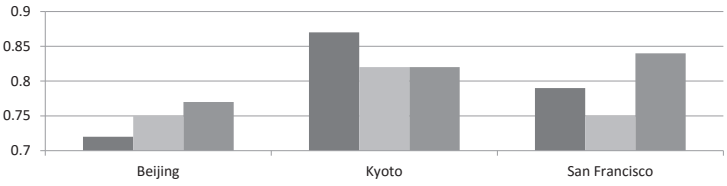
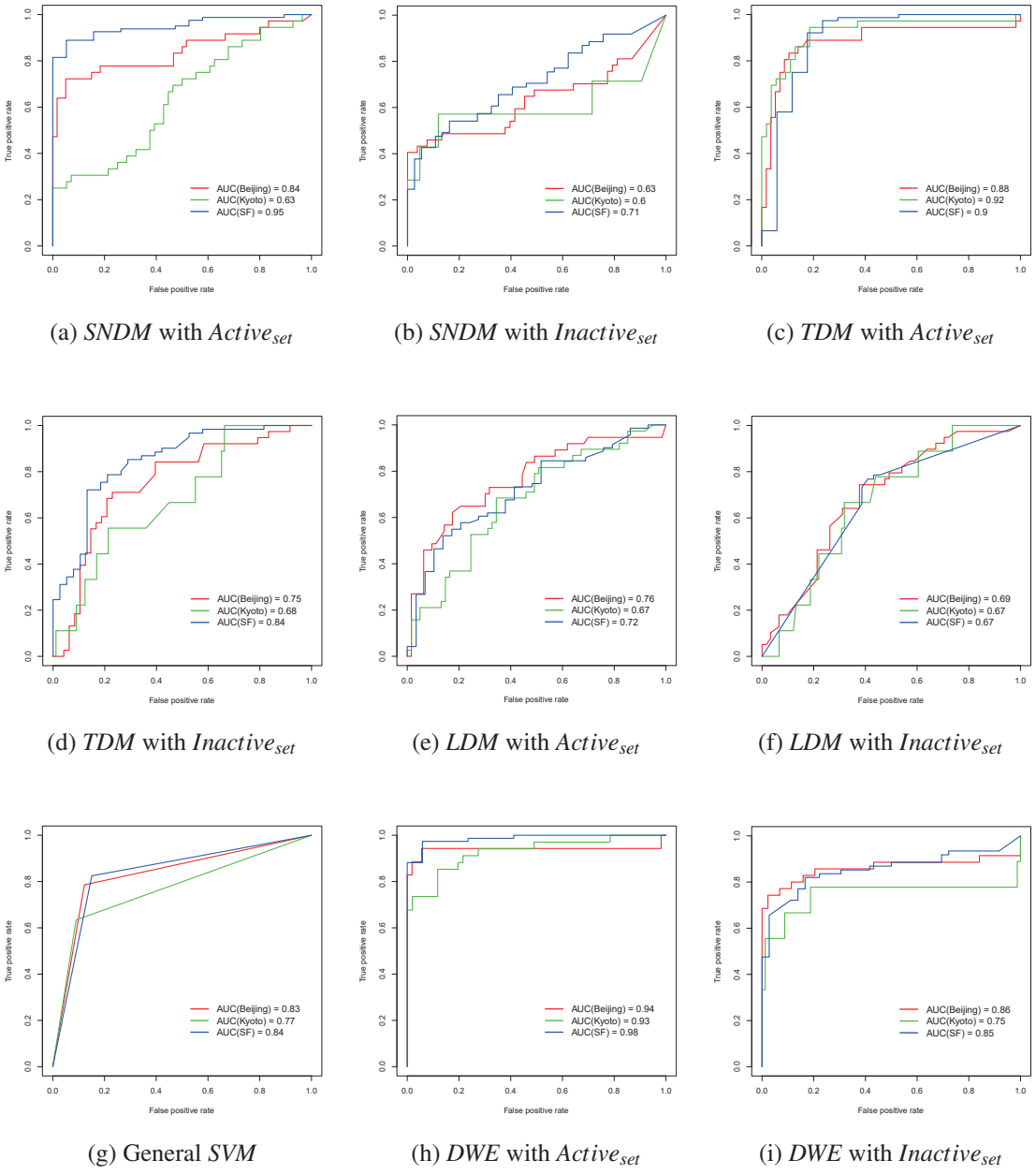


Figure 3.5. Labeling variances on the bottom 100 inactive users in each city, calculated using the Jaccard index.



(a) *SNDM* with $Active_{set}$ (b) *SNDM* with $Inactive_{set}$ (c) *TDM* with $Active_{set}$

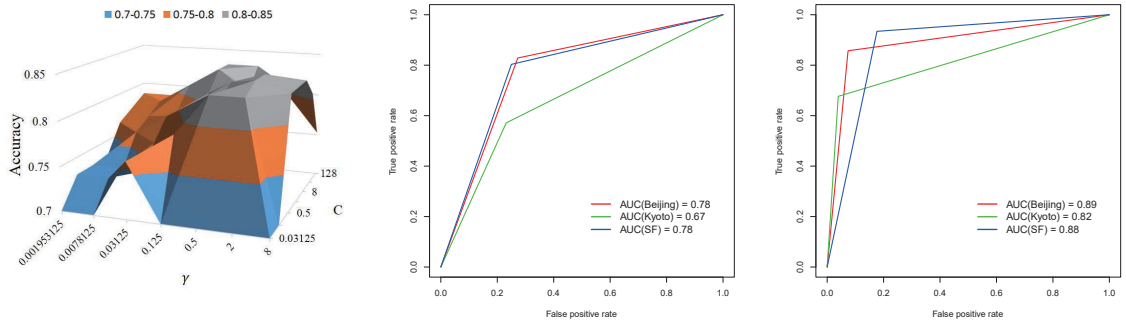
(d) *TDM* with $Inactive_{set}$ (e) *LDM* with $Active_{set}$ (f) *LDM* with $Inactive_{set}$

(g) General *SVM* (h) *DWE* with $Active_{set}$ (i) *DWE* with $Inactive_{set}$

Figure 3.6. Overall results for the location-authority-based user-classification task. By drawing the ROC curves, the five models' performance on both active (upper bound) and inactive (lower bound) datasets are presented.

Table 3.7. Classification accuracy of the general SVM models when using different numbers of features. $\mathfrak{D}_1$: $Active_{set}(Beijing)$; $\mathfrak{D}_2$: $Active_{set}(Kyoto)$; $\mathfrak{D}_3$: $Active_{set}(SanFrancisco)$; $\mathfrak{D}_4$: $Inactive_{set}(Beijing)$; $\mathfrak{D}_5$: $Inactive_{set}(Kyoto)$; $\mathfrak{D}_6$: $Inactive_{set}(SanFrancisco)$;

Datasets	**One feature**			**Two features**			**Three features**
	SNDM	TDM	LDM	SNDM & TDM	SNDM & LDM	TDM & LDM	SNDM & TDM & LDM
$\mathfrak{D}_1$	0.854	0.876	0.674	0.876	0.854	0.865	**0.899**
$\mathfrak{D}_2$	0.682	0.847	0.647	0.824	0.694	0.847	**0.847**
$\mathfrak{D}_3$	0.828	0.882	0.641	**0.935**	0.774	0.871	0.892
$\mathfrak{D}_4$	0.747	0.709	0.620	0.772	0.747	0.684	**0.772**
$\mathfrak{D}_5$	0.744	0.798	0.764	0.775	0.794	0.865	**0.888**
$\mathfrak{D}_6$	0.759	0.773	0.557	**0.784**	0.567	0.660	0.701



(a) Tuning parameters for the *SVM* (b) Train: $Active_{set}$; Test: $Inactive_{set}$ (c) Train: $Inactive_{set}$; Test: $Active_{set}$

Figure 3.7. Detailed results for the SVM model: (a) the parameter-tuning process for obtaining the best classification model; (b) and (c) results from respectively using the $Active_{set}$ or $Inactive_{set}$ as the training set and the other for testing. Because the SVM model has only two outputs (high or low authority), there is only one threshold for the ROC curves.

ative effect on the classification results. Since inactive users always have few friends, SNDM is not so good at dealing with these users. The classification results shown in Figure 3.3 further verified this observation. For users with few friends, i.e., lower indegree and outdegree on the social network (see Definition 1), SNDM has insufficient discriminative power. Moreover, we analyzed the situations in different cities in order to gain insight into this model. By investigating the lower AUC values for Kyoto, we found that, because Kyoto is a traditional city, relatively few online social relationships are established there, compared to modern cities like San Francisco. In this sense, the performance of this model depends on different kinds of social culture.

The time-driven model (TDM) always performed well. Because the users in $active_{set}$ uploaded many images, this model obtained considerably good results that closely approximated the ground truth, as shown in Figure 3.6c. However, for users with few images, the model's performance decreased sharply. Figure 3.6d shows the model's performance when classifying inactive users. With Kyoto, for example, the low performance ($AUC = 0.68$) was due to calculations on a sparse matrix M, owing to the inactivity of users who upload few images on Flickr. Of the bottom 100 users from $Inactive_{set}(Kyoto)$, each of them uploaded an average of 89 images. In this sense, for users with few images, i.e., high sparsity of the visiting matrix (see Definition 2), TDM has insufficient discriminative power.

The location-driven model (LDM) returned the most stable results. Figures 3.6e and 3.6f show the performance of the location-driven model. Although this model did not perform as well as the time-driven model, it did prove that location information is an important feature for stable results. That is, no matter what kind of raw data (e.g., how many friends, how many images, etc.) we collected for each user, LDM always performs stably. A more important characteristic is that combining it with other features improves significantly over using them alone, suggesting that the LDM adds important value to this classification task.

The SVM model provides a solid baseline when integrating the above three models. By using $Active_{set}$ and $Inactive_{set}$ as a whole, a three-fold cross validation was applied to train multiple SVM classification models. Figure 3.7a shows the grid-search method for tuning the parameters. The X-axis represents the RBF kernel's parameter γ, whereas the Y-axis denotes the penalty parameter C in the SVMs. After choosing the best model from the parameter-tuning results, Figure 3.6g shows the general SVM model's superlative classification performance. Rather than randomly sampling training and testing data sets, Figures 3.7b and 3.7c show a cross-validation on $Active_{set}$

and $Inactive_{set}$. Because the model trained by $Inactive_{set}$ performed better than the model trained by $Active_{set}$, it is worthwhile exploring and implementing sophisticated features, rather than treating all of the features equally.

Table 3.7 further confirms that more features do not necessarily lead to better results. With San Francisco, for example, the SVM model obtained the best results when only two features were used. The relatively low performance from combining more features is due to the probability of existing disagreements between different models. This result has further verified our observation that SNDM, TDM, and LDM are good at dealing with different available raw data. Therefore, the DWE model is one of the best solutions to this problem.

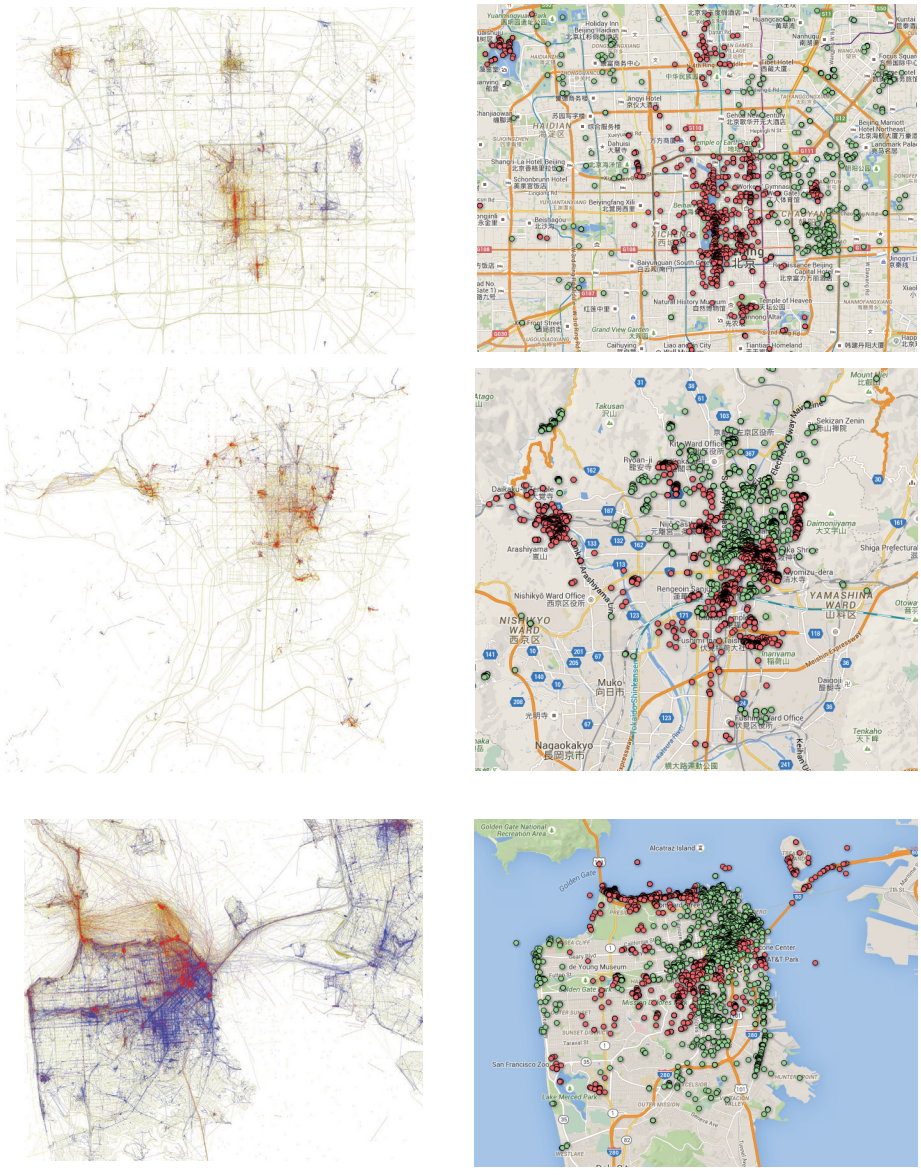
The DWE model performed best. This model is an advanced version of the SVM model. The DWE model intelligently selects the most discriminative features for each user. By considering the differences in the available information for each user, dynamic weights were assigned to each feature. As a result, those models (SNDM, TDM, or LDM) that have sufficient discriminative power are automatically selected by DWE to classify each user. Ultimately, this model obtained the best results for both the $Active_{set}$ (Figure 3.6h) and $Inactive_{set}$ (Figure 3.6i).

3.4.4 Visualizing the Classification Results

Because our location-driven model, introduced in Section 3.3.3, was devised in accordance with the statistical observations of [1], we visualized our results to show that our model could generate similar distributions for the three cities Beijing, Kyoto, and San Francisco. Note that they used datasets that differ from ours, and neither our results nor theirs can be regarded as correct.

Although temporal information is an essential feature of such classifications, our results confirm that using only latitude–longitude coordinates can also perform well. Figure 3.8 visualizes the results. The red icons represent geo-tagged images taken by Flickr users who are very likely to be unfamiliar with the city, whereas the green icons represent images probably taken by those who are familiar with the city. Through such a classification, we can easily find popular sightseeing locations (i.e., the red areas) in each city. Thus, the location-driven model not only improves the time-driven model, it also models the relationships between people and space by introducing the inherent *authority* attribute.

To explain the latent variable *group*, Table 3.8 shows the number of votes obtained



(a) Results in [1] (b) Our results

Figure 3.8. Visualization of different user groups' mobility behavior in Beijing, Kyoto, and San Francisco. Temporal information is analyzed in (a), whereas location information is utilized in (b).

Table 3.8. Representative users for the two latent groups.

Groups	**Votes for the top ten representative members in each group**		
Tourist-like group	Beijing: 22/30;	Kyoto: 23/30;	San Francisco.: 26/30.
Local-like group	Beijing: 22/30;	Kyoto: 25/30;	San Francisco: 21/30.

by the representative members from each group. The more votes a user has (with a maximum of three votes from the subjects), the more likely the user belongs to the corresponding group. By analyzing the votes in Table 3.8 and the distributions in Figure 3.8, we found that the representative members accurately reveal the underlying meaning of each group, i.e., a tourist-like group and a local-like group.

To draw a conclusion, in this chapter, we proposed a new concept, *location authority*, for mining information asymmetries among different online user groups. By respectively considering a user's social, temporal, and spatial information, we devised three models for estimating users' location authority: a social-network-driven model, time-driven model, and location-driven model. We then introduced two ensembles of these models, which demonstrated robust performance across different classification tasks. In the experiments, we manually collected the ground truth and compared the performance of each proposed method. We also visualized the experimental results in order to show the broader applications of our methods. Our work is intended to inspire more interest in information asymmetry analysis, user profiling, and geo-social recommendations.

CHAPTER 4

SINGLE LOCATION PROFILING

In this chapter, a location's characteristics will be discussed on the basis of user activity analysis. Two dimensions, i.e., (1) user distribution based popularity and (2) scenery quality will be analyzed. Compared with the conventional related work aiming to discover hot or famous Points of Interests (POIs), our work is among the first of its kind, which can not only evaluate the popular POIs but also the obscure ones. By applying the location profiling methods introduced in this chapter, we further developed a demo of recommending obscure locations in Kyoto.

4.1 Motivation

Travel, as an indispensable part of the service sector, has already been well developed worldwide and plays an important role in people's lives. Powered by the Internet, abundant tourism information, including traditional events, accommodation, transportation, suggested itineraries, etc., is uploaded and shared on websites. On conventional travel websites, information about the most famous sightseeing resources is stressed, well organized and can be easily referred to by tourists. However, the information about obscure sightseeing locations known by a small number of people is rarely organized or even described on the Internet. Since personalized tours are becoming popular, more attention is focusing on obscure sightseeing locations that are less well-known while still worth visiting. Moreover, these obscure locations are potential sightseeing

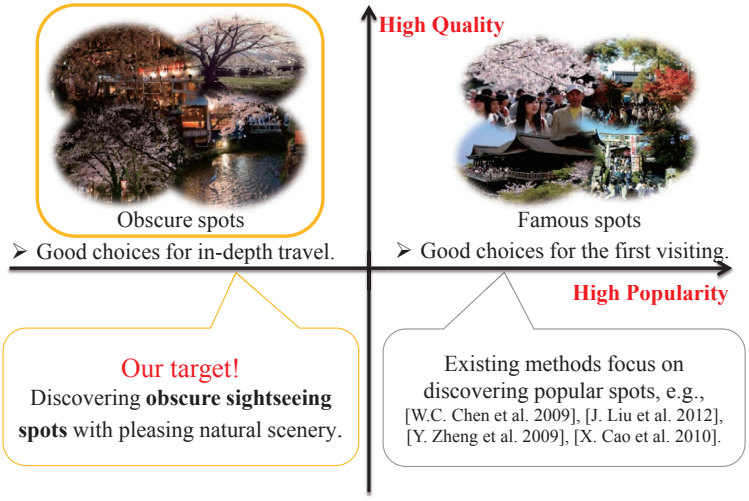


Figure 4.1. A sightseeing location can be evaluated from two aspects, i.e., (1) popularity and (2) scenery quality.

resources valuable for developments and promotions to governments and travel agencies. As a result, there exists a challenge to effectively discover and recommend such locations.

Benefiting from Social Networking Service (SNS) and advances in mobile devices, people can upload and share their experiences on the Internet. Geo-tagged images are one of the typical content generated by users. The vital information it contains provides researchers with excellent opportunities for discovering landmarks and moving patterns [61][44][62][63][64][65]. More importantly, based on a survey by Zheng et al. [9], the growing geo-referenced and community-contributed media resources have generated huge amounts of detailed location and event tags, covering not only famous landmarks but also obscure locations. However, to the best of our knowledge, the research on discovering obscure sightseeing locations has not been well studied before, i.e., most existing works [44][45][20][3][66] pay more attention to hot and famous areas and landmarks.

As shown in Figure 4.1, we generally divided sightseeing locations into four quadrants on the basis of their "popularity" and "scenery quality". Located in the quadrant with high sightseeing quality but low popularity, an obscure sightseeing location is a considerable choice for in-depth travel to not only enjoy the beautiful scenery but also experience local culture, especially for the repeat tourists who have already visited the most famous places there.

In this chapter, to a given scenery object, we propose different methods to discover and rank the obscure sightseeing spots with two novel metrics, "obscurity level" and "scenery quality", by analysis of geo-tagged images retrieved from Flickr.

Regarding the *obscurity level estimation*, it is somewhat a measurement of information asymmetry between different photographer groups. "Obscurity level" is a relative concept, which means a location can be popular to Group A (e.g., residents), but obscure to Group B (e.g., travelers). In this work, we try to discover the locations that are obscure to travelers who are unfamiliar with the corresponding cities.

Regarding the *scenery quality evaluation*, our research evaluates the aesthetic value of the target scenery object per each spot.

To summarize, we make the following major contributions in this work:

- We propose a novel mechanism to discover sightseeing spots from two aspects: obscurity level and scenery quality. The implementation of our proposed methods are validated on the basis of field visits in Kyoto.
- To estimate the user distribution based location popularity, by utilizing the user profiling methods introduced in Chapter 3, a novel information asymmetry based method is proposed.
- For the scenery quality dimension, we evaluate a location by analyzing user behavior related to the images shot around it. The behavior of both viewers who browsed the social images (i.e., social appreciation) and photographers (i.e., photographers' attention) are considered.

The remaining content of this chapter are as follows. Section 4.2 talks about some related work in this area of discovering and recommending POIs. Section 4.3 shows the overview of the problem and our preliminary work. In sections 4.4 and 4.5, a location's popularity and quality will be analyzed, respectively. In Section 4.6, our methods will be verified by field visits in Kyoto.

4.2 Related Work

Although the location profiling will also depend on the user profiling related work, since we have already discussed them in Chapter 3, we omit these discussions here. As a result, Our research in this chapter is related to the following three areas.

Discovering Popular Locations: Many related works study on landmarks that are always located in the quadrants with high "popularity" (see Figure 4.1). The attractiveness and popular appeal of landmarks result in a vast amount of landmark related media resources on the Internet. As a result, it has spurred much research into "building a landmark database", "landmark visual summarization and 3D modeling", and "landmark recognition". For instance, several studies have investigated building landmark databases at a city level [67][68]. Zheng et al. [66] first attempted to establish a world-scale landmark database. A live landmark recognition engine has been proposed that can visually recognize landmarks in a given image. With recent advances in mobile computing, research into visual localization or landmark identification on mobile devices is gaining interest. Chen et al. [61] published an extensive landmark dataset in the research area of mobile location recognition on a city scale. When a user sends a photo taken with his/her mobile phone, in return s/he receives information about the current position. As the number of landmark images grows rapidly, browsing and summarizing landmark photo collections become important. Researchers have attempted both summarizing landmarks with representative images [68][69] and 3D reconstruction of landmarks from internet images [70][5]. However, methods to discover obscure rather than popular sightseeing locations have not been well investigated.

Ranking Sightseeing Locations: In the research into estimating "sightseeing quality", a survey given by Luo et al. [19] shows that collections of geo-multimedia, which are a result of sightseeing experiences sharing by web communities, are widely used in trip recommendations. Ji et al. [20] modeled the relationships of scene/landmark and scene/authorship as a graph and adopted two popular link analysis methods, PageRank and HITS, to mine representative landmarks. Zhang et al. [32] aimed to mine interesting locations and classical travel sequences in a given geospatial region on the basis of multiple users' GPS trajectories. They first modeled multiple individuals' location histories with a tree-based hierarchical graph. Then, by using the graph, they proposed a HITS-based inference model that infers the interest of a location. In [65], the authors further developed a recommendation system. Instead of GPS traces, Liu et al. [3] proposed a joint authority analysis framework to discover areas of interest with geo-tagged images and check-ins. By utilizing geo-tags and user-tags of images, Chen et al. [44] developed a visualization application for automatically generating tourist maps. Hasegawa et al. [71] attempted to organize travel related tweets by considering the spatio-temporal continuity of user-behaviors during travel. By merging such fragmented tweets, users' travel experiences can be detected.

In these works, GPS traces, images, check-ins, and tweets are treated as different kinds of user votes to help gather tourism knowledge. Authority based analysis, like "rank-by-count" and "rank-by-frequency" in a vote manner, is the basis for most of these trip recommendation research. However, for an obscure location, not enough visits or votes are generated on the Internet. The classical authority based analysis, which is always used to recommend popular sightseeing locations, is not suitable.

Evaluating Aesthetic Value: Researchers have already proposed some methods to rank attractiveness of images [72][73]. In addition, Savakis et al. [74] presented a subjective evaluation of the significance of different visual aspects for the determination of the overall appeal of natural images. In [75], Winkler proposed different ways to quantify sharpness and colorfulness of images. He conducted further experiments to show the properties of these features as effective indicators of image appeal.

A lot of additional metrics, such as exposure, contrast, or texture, have also been extracted and used to measure an image's content. Nonetheless, an image may be attractive solely due to professional skills of a photographer rather than there being beautiful scenery at a location. To the best of our knowledge, how to evaluate the scenery quality of a given location by using the photos shot around there has not been well investigated so far.

4.3 Preliminary Work

In this section, we mainly introduce the data structures that will be used in the next two sections. Since some of the data structures have already been defined in Chapter 3, here, we use a figure to visualize them. Accordingly, it is much easier to understand the relationships between these data structures. On the other hand, the two tasks (i.e., user distribution based popularity estimation and scenery quality evaluation) will be describe briefly.

4.3.1 Data Construction

Through Flickr's APIs*, we first retrieve geo-tagged images and photographers' information into two data sets: *GeoImage* and *PProfile*. Then, three additional data structures are modeled as shown in Figure 4.2.

*https://www.flickr.com/services/api/

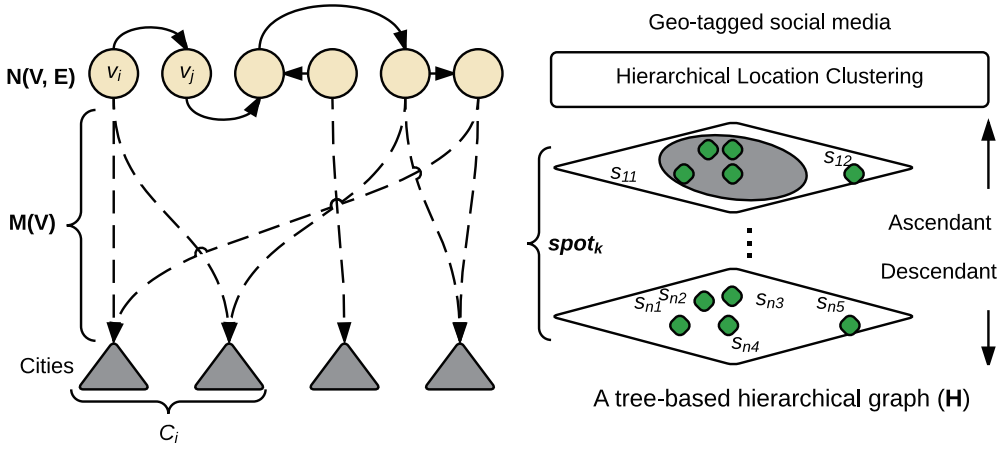


Figure 4.2. Data construction. Details are described in Definitions 1, 2 and 4.

By seeing the left side of Figure 4.2, it presents the relationships among social network N (see Definition 1 in Chapter 3), visiting frequency matrices M_i (see Definition 2 in Chapter 3), and cities C_i. It describes that users v_i have different friends on the network N. Also, different users will visit different cities C_i.

By seeing the right side of Figure 4.2, it describes the "locations" we focused on in this chapter. The figure shows that the "locations" have a hierarchical structure. Definition 4 below gives the detailed information.

Definition 4. *A tree-based hierarchical graph H is a collection of location clusters, denoted as $spot_k$, with a geographical hierarchical structure. A $spot_k$ is a candidate sightseeing place, which is obtained by a clustering algorithm using the retrieved images' geo-information. Correspondingly, the set of all images that are taken at $spot_k$ is denoted as I_k.*

We put together the geographic coordinates (i.e., latitude, longitude) from geo-tagged images into a dataset to obtain the tree-based hierarchical graph H, with which the system can select spots with different granularities to display them in different levels of details (LOD). In this chapter, hereinafter, a spot $spot_k$ denotes a location.

4.3.2 Overview of the Two Tasks

In accordance with the two dimensions, i.e., "popularity" and "sightseeing quality", shown in Figure 4.1, two corresponding challenges are addressed in this work:

User Distribution based Popularity Estimation: Given social network $N(V,E)$ and visiting frequency matrices M_i, we first estimate each photographer v_i's familiarity with a particular city $c \in C_i$, based on which all the photographers are classified into a *familiar group* and an *unfamiliar group*. Then, on the basis of visiting frequency asymmetry between the photographers from these two groups, we estimate the obscurity level of each $spot_k$. Intuitively, if more photographers from the familiar group than from the unfamiliar group visit a candidate spot, we assume that spot has a high probability of being obscure to travelers.

Scenery Quality Evaluation: For the second challenge, two kinds of user behavior are identified to evaluate the scenery quality in different ways. Regarding the behavior of users who viewed target images, we evaluate a $spot_k$'s social appreciation to present how much on-line community members like some scenery around there. On the other hand, by analyzing the behavior of photographers who took target images, we try to evaluate the attention they have paid on some scenery at $spot_k$.

4.4 User Distribution based Popularity Estimation

As have been mentioned in Section 4.1, since the "popularity" is a relative concept on the basis of different user groups (e.g., locals, foreign tourists, etc.), before calculating the extent of location popularity, we should first do user classifications to identify different user groups. Accordingly, we classify users (photographers) into two groups, i.e., the (1) familiar group and (2) unfamiliar group.

Then, by analyzing the information asymmetry between these two groups, we calculate the user distribution based location popularity. As a result, our work could discover both popular and obscure locations (spots) in a city.

4.4.1 User Classification

In this subtask, we will directly use the users' location authority score devised in Chapter 3 to classify photographers into the *familiar group* and *unfamiliar group* by considering 1) their profiles, 2) their friends' familiarity, and 3) visiting frequency to the target city.

Given a user (photographer) v_i and a target city c, we calculate a familiarity score $F(v_i,c)$ using different methods proposed in Chapter 3. The details are as follows:

- **Social Network $N(V,E)$ based familiarity score.** We assign the location authority score calculated by Algorithm 2 to $F(v_i,c)$.
- **Visiting Frequency Matrices M_i based familiarity score.** We assign the location authority score calculated by Eq. 3.2 to $F(v_i,c)$.
- **SVM based familiarity score.** We assign the location authority score calculated by SVM integration model in Section 3.3.4 to $F(v_i,c)$.
- **DWE based familiarity score.** We assign the location authority score calculated by DWE integration model in Section 3.3.4 to $F(v_i,c)$.

Please note that, because the location-driven model, i.e., Algorithm 3 in Chapter 3, cannot be applied alone (please see Section 3.3.4 for details), there are totally four different ways to do the user classification. Accordingly, in the experiments, by comparing these different strategies, we will discuss the advantages and disadvantages about them. On the other hand, the performance of location popularity estimation can be regarded as an indirect verification of our user profiling methods proposed in Chapter 3. Therefore, we will also discuss this issue in the experiment section.

4.4.2 Location Popularity Score Calculation

For a target city c, we first introduce a *fuzzy set* $\{U_c, m(v_i) = F(v_i,c)\}$ to specify each user v_i's familiarity on the basis of the familiarity score $F(v_i,c)$. Then, for a $spot_k$, we define two vectors, $\phi_f^{\vec{spot}_k}$ and $\phi_{unf}^{\vec{spot}_k}$, to represent the percentage of both the familiar group and unfamiliar group who have visited $spot_k$ in each year y:

$$\phi_f^{\vec{spot}_k}[y] = \frac{w_y}{\left|U_c^k\right|} \sum_{v_i \in U_c^k} m(v_i);\ \phi_{unf}^{\vec{spot}_k}[y] = w_y - \phi_f^{\vec{spot}_k}[y]. \tag{4.1}$$

Similar to the w_i in Eq. 3.2, w_y is an empirical parameter to characterize the staleness of information. In experiments, we find that setting $w_y = 0.8^{(\lambda - y)}$ obtained good performance.

Finally, to calculate the time sensitive familiarity before year λ, the entries in $\phi_f^{\vec{spot}_k}$ and $\phi_{unf}^{\vec{spot}_k}$ should include all the detected years before λ. We calculate $spot_k$'s obscurity score by comparing the norms of these two vectors.

$$ObscurityScore(spot_k, \lambda) = \left\| \phi_f^{\vec{spot}_k} \right\| - \left\| \phi_{unf}^{\vec{spot}_k} \right\|. \tag{4.2}$$

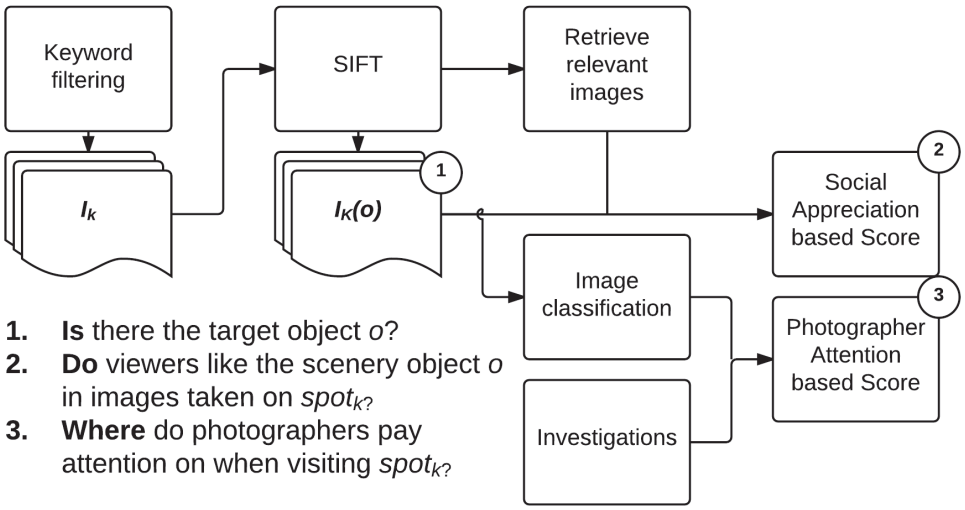


Figure 4.3. The framework of ranking spots. Corresponding to Sections 4.5.1, 4.5.2 and 4.5.3 respectively, our system uses three criteria (i.e., the three questions listed in this figure) for scenery quality evaluation.

On the basis of the tree-based hierarchical graph H (see Definition 4), the obscurity level of $spot_k$ with various granularities can be estimated, as long as in our retrieved database there is at least one photographer who uploaded the target images about $spot_k$.

4.5 Scenery Quality Evaluation

Unlike the conventional related work on evaluating hot or famous locations, our work faces with a challenge of evaluating obscure locations. To the best of our knowledge, our proposed method is among the first of its kind, which can evaluate the scenery quality of both popular and obscure locations.

The biggest challenge of ranking obscure spots (namely, having low popularity) is how to handle the lack of raw data including geo-tagged images, tweets or check-ins, because less relevant information is generated on the Web. In other words, the authority based analysis methods previously utilized [20][3][65][32] are not suitable in our cases. In this subtask, given a scenery object o (e.g., cherry blossom, maples, etc.), we use social images to rank $spot_k$ by evaluating the aesthetic value of the scenery object o. Compared with the existing very few geo-tagged tweets and check-ins, images describe a spot much more realistically and objectively.

Figure 4.3 presents the framework of evaluating a $spot_k$'s scenery quality. By ob-

serving the behavior of both viewers (i.e., who have viewed the images in I_k) and photographers (i.e., who have taken the images in I_k), we defined three progressive criteria. Along with this flow chart, the three criteria will be introduced in the remainder of this section.

4.5.1 Criteria 1: Image Labeling

Hereinafter, the set of all images about a particular scenery object o that are taken at $spot_k$ is denoted as $I_k(o)$. Benefiting from Flickr's well implemented keyword based search function[†], the filter has already obtained a good performance. For example, through manually statistics, the percentage of noise images in $I_k(o = cherryblossom)$ is about 7.72 % on average.

In our system, we further applied SIFT [76] to reduce noise images in $I_k(o)$. In this model (see Figure 4.3), we use the key-points detected by SIFT to describe an image. For calculating the similarity, a fast nearest-neighbor algorithm[‡] is used to do the matching by considering both numbers of matches and match distance. When filtering, $I_k(o)$ itself is used as a set of references. After mutual match, the system removes the images obtaining the lowest similarity scores. By sampling in our ranking experiments, the percentage of noise images finally dropped to about 5.86 %.

After noise image filtering, we have met the first and most elementary criteria that there is the scenery object o at $spot_k$. In our system, all the images in $I_k(o)$ will be displayed on the exploring map interface.

4.5.2 Criteria 2: Social Appreciation based Analysis

In this method, we use an intuitive value, the number of "Favs", to evaluate a scenery object's quality in a social appreciation manner. "Fave this photo" is a common user behavior on SNS websites including Flickr for friends and visitors to mark on a message or an image. In [73], Pedro et al. used the number of "Favs" as ground truth for building and testing their image attractiveness ranking models. However, our task is faced with two new problems:

1. Few images may be taken at $spot_k$;

[†]Flikcr API: flickr.photos.search

[‡]Library FLANN (http://www.cs.ubc.ca/research/flann/)

2. A higher "Favs" may mean a beautiful image rather than beautiful scenery included in this image.

Accordingly, we try to estimate the social appreciation based scenery quality in a relative manner by introducing two kinds of $I_k(o)$'s related images.

Similar Content Images: We first retrieve $I_k(o)$'s similar content images. Similar to a semi-supervised learning method having a small training set, we expand the existing data sets (i.e., $I_k(o)$) by detecting the similar content images. For each image in $I_k(o)$, we retrieve its most similar images by applying SIFT as shown in Figure 4.3. The newly expanded image set is denoted as $I_k^s(o)$. High social appreciation on $I_k^s(o)$ anyhow shows that the $spot_k$ is a good place for someone to take beautiful images about object o.

Context Images: We set the images from the same photosets to which an image $\eta \in I_k(o)$ belongs as its context images, denoted as C_η. Note that the image η itself is not in C_η. In this sense, both η and images in C_η have a similar chance of being accessed by viewers. In addition, all these images are likely taken by the same photographer as well as similar shooting skills. Under these contexts, compared with C_η, higher social appreciation on η suggests that η contains beautiful scenery.

On the basis of these two related image sets, the final social appreciation based score consists of two parts. The first $I_k^s(o)$ based part is computed by the mean of ratios χ between an image η's number of "Favs" (i.e., $vote_\eta$) and view count (i.e., $view_\eta$):

$$S_1(spot_k, o) = \mu_\chi(I_k^s(o)); \ \chi(\eta) = \frac{vote_\eta}{view_\eta}. \tag{4.3}$$

The second C_η based part is computed by z-scores:

$$S_2(spot_k, o) = \frac{1}{|I_k(o)|} \sum_{\eta \in I_k(o)} \frac{\chi(\eta) - \mu_\chi(C_\eta)}{\sigma_\chi(C_\eta)}. \tag{4.4}$$

Function $\mu_\chi(C_\eta)$ and $\sigma_\chi(C_\eta)$ are used to respectively calculate the average χ and χ's standard deviation on the image set C_η. Thus, positive and negative z-scores represent observation images above and below the mean, respectively. To combine S_1 and S_2, a weight factor ω_c is used to adjust discrimination.

$$SAS(spot_k, o) = S_1 + \omega_c \cdot S_2. \tag{4.5}$$

By proposing the *SAS*, we have met the second criteria of answering where to take beautiful pictures. By adapting this system to be a photography suggestion guide,

the value of ω_c is somewhat a trade-off of recommending either attractive images or beautiful scenery. However, nobody can indeed know the reasons a viewer "Faves" an image seen when browsing the Web. Sometimes, we "Fave" images just because our friends have posted them. Thus, in addition to the image viewers, as another stakeholder, we try to evaluate the scenery quality from a photographer's point of view.

4.5.3 Criteria 3: Photographer Attention based Analysis

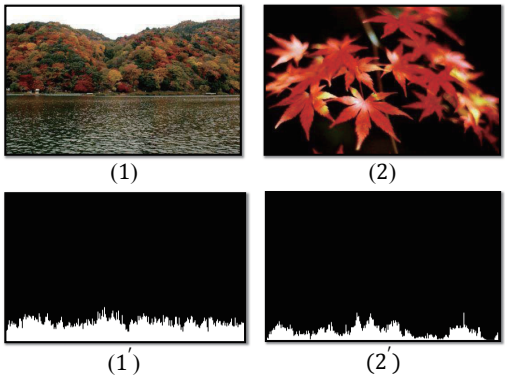
For a scenery object o, it needs not only to be beautiful from a distance but also to have a delicate beauty up close. Figure 4.4(a.1) shows beautiful maple trees from far away, while in Figure 4.4(a.2) it shows a close-up of the beautiful leaves. The photographers' choices of taking distance view or close-up view images psychologically indicate their attention to a scenery object o.

Accordingly, we first propose a new method to divide scenery images into "distance view" and "close-up view", which exactly correspond to the two kinds of beauty above. Then, the regular patterns of photographers' attention when visiting the known sightseeing spots with beautiful scenery object o are recognized. At last, we calculate the photographer attention based score, which indicates the degree of agreement with an observed regular pattern.

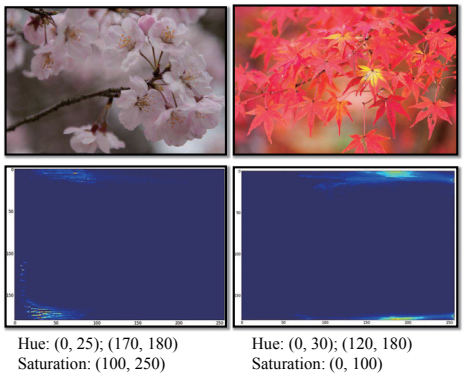
Image Classification

In the classification learning method, we use an image's edge histogram and HSV (hue, saturation, and value) shown in Figures 4.4a and 4.4b as features to generate the training model. For the edge histogram, we find that a close-up view image has more clearer edges in the part where the photographer's camera focused. In other words, the background of a close-up view image is more blurred. In contrast, a distance view image has a gentle contrast throughout the whole image. For the HSV, we assume that the hue and saturation values of close-up view images of particular scenery are always located in a relatively fixed range (see Figure 4.4b). In contrast, there is no such rule for distance view images. This is due to that, for close-up view images of particular scenery, most content of the images is generally made up of the target scenery objects.

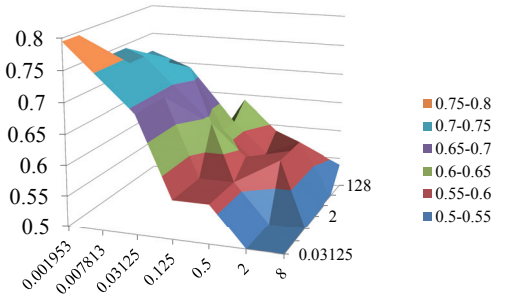
To evaluate the image classification performance, two original datasets, which repectively include 100 images, are labeled manually with either "distance view" or "close-up view". Then, 3-fold cross validation is utilized. In Figure 4.4, (c) $\sim$ (f) illustrate the effectiveness of image classifiers on both the $Cherry_{set}$ and the $Maple_{set}$ by a grid



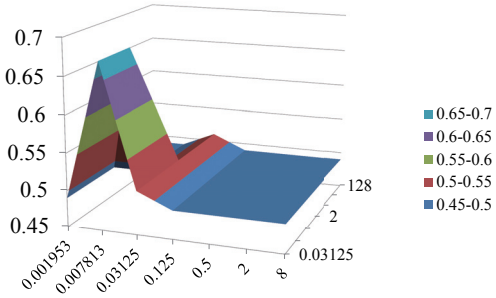
(a) Image edge histogram (EH)



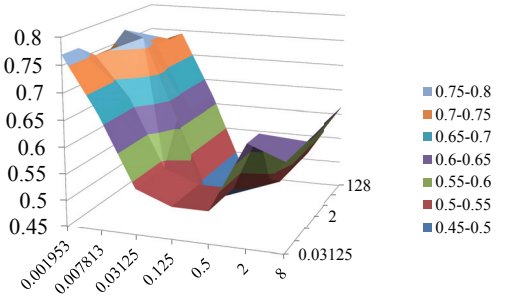
(b) Image HSV



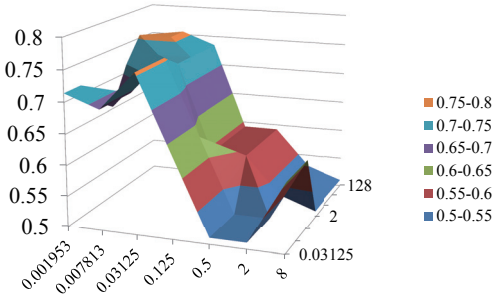
(c) EH on $Cherry_{set}$



(d) HSV on $Maple_{set}$



(e) EH&HSV on $Cherry_{set}$



(f) EH&HSV on $Maple_{set}$

Figure 4.4. Image classification. In (a), an image is vertically divided into 256 bins, and for each, the points of detected edges are counted to set up the histogram (i.e., EH). (b) presents the saturation (X-axis) and hue (Y-axis) distributions of close-up view images.

search method proposed by Hsu et al. [54]. The X-axis ($\gamma = 2^{-9}, 2^{-7}, ..., 2^{3}$) represents RBF kernel's parameter, while the Y-axis ($C = 2^{-5}, 2^{-3}, ..., 2^{7}$) stands for the penalty parameter in SVMs.

As shown in Figure 4.4c and Figure 4.4d, the performance varies when using one of the features alone. For the edge histogram, the best accuracy could reach greater than 80%. Conversely, for the HSV, the accuracy on maple image set is as low as about 65%. One reason is because this dataset contains some images about green maples. In this case, although these images are close-up view images, the classification model divided them incorrectly as the hue and saturation value of these images do not lie in the expected ranges.

In accordance with our experiments described in Figure 4.4e and Figure 4.4f, when both of the features are used, our classification model reaches a stable accuracy that is close to 80%. Torralba et al. [77] proposed a method to estimate the absolute depth of an image on the basis of the whole scene structure. In a range of fault-tolerant, for man-made scenes, the performance of depth estimations reaches 76%, while the result is relatively lower on natural image sets (about 70% on average). When applied in our task of classifying scenery images into two categories: (1) "distance view"; or (2) "close-up view", our methods outperformed those of Torralba et al. [77].

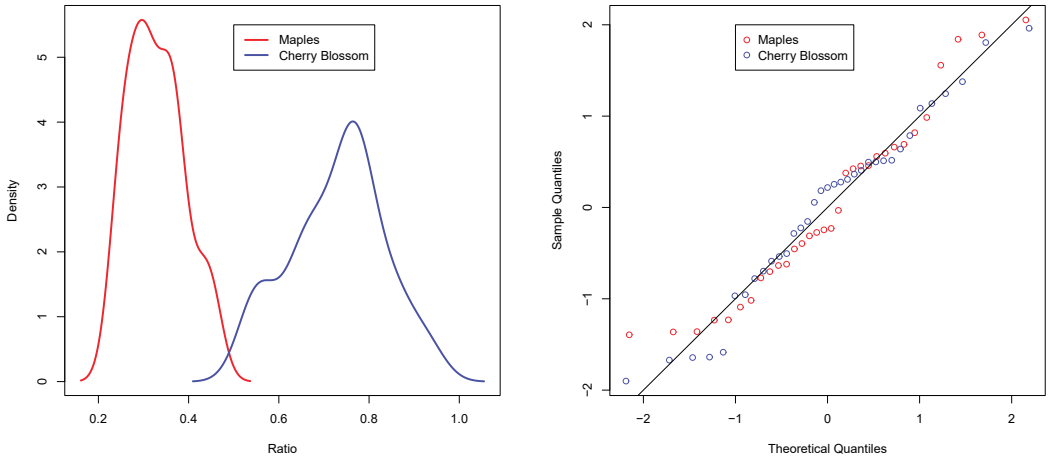
Photographer Attention Based Scenery Score

In accordance with the image classification, without loss of generality, we use the ratio of distance view images taken at a spot as r_o, which represents photographers' attention to the scenery object o's beauty from a distance.

Figure 4.5a shows the results of two investigations on the known spots having beautiful maples and cherry blossom. For maples, we find that photographers paid more attention to the up-close beauty. Conversely, cherry blossom's beauty from a distance attracted more attention compared with its up-close beauty.

Furthermore, by seeing the normal Q-Q plots in Figure 4.5b, we hypothesize that the two sample data sets come from the populations with normal distribution, since the points fall approximately along the 45-degree reference line. We performed the Lilliefors test, a kind of normality tests, to verify our hypothesis. By obtaining the $p - value = 0.26$ and 0.28, which are higher than the significance value (0.05 or 0.1), our hypothesis is statistically true.

As a result, based on the attention patterns recognized from spots having beautiful scenery o, a $spot_k$'s photographer attention based score is computed using a probability



(a) Density of r_o

(b) Fitting verification of r_o

Figure 4.5. Distribution fitting. Red: 22,421 geo-tagged maple images taken at 32 spots with beautiful maples; Blue: 10,748 geo-tagged cherry blossom images taken at 35 spots with beautiful cherry blossom.

density function. For instance, since maples and cherry blossom both have the same pattern, we use the normal distribution functions to get their scores:

$$PAS(spot_k, o) = \frac{1}{\sigma_o\sqrt{2\pi}} e^{-\frac{(r_o(k)-\mu_o)^2}{2\sigma_o^2}}. \tag{4.6}$$

μ_o and σ_o are obtained by distribution fitting of investigation results on the target scenery object o. $r_o(k)$ is the ratio of distance view images taken at $spot_k$. Despite different patterns would be recognized for different o, corresponding probability density functions are fitted. Intuitively, the *PAS* indicates how well a $spot_k$ can attract visitors' attention.

4.6 Experimental Evaluation

We first show an outline of the retrieved data sets and parameter selections in our experiments. Then, the user distribution based location popularity estimation will be evaluated by applying them to discover obscure spots in Kyoto. Finally, the scenery quality based ranking results will be evaluated by field visits.

Table 4.1. Details of the target two image datasets that are labeled by the queries $q_1 =$ "$Kyoto, Maples$" and $q_2 =$ "$Kyoto, CherryBlossom$".

	q_1	q_2
Images number	6,950	4,576
Photographers number	177	319
Images/photographers average number	39.3	14.3
Images from top 10% photographers average number	274	92.7
Percentage of images from the top 10% photographers	60.9%	62.9%

4.6.1 Data Preparation

From Flickr, we collected a total number of 929,403 images. All the *similar content images* and *context images* are also included in the data set, which would be used for the ranking tasks implemented in section 4.5.2.

Using queries $q_1 =$ "$Kyoto, Maples$" and $q_2 =$ "$Kyoto, CherryBlossom$", two target sub-datasets are labeled by our system. Tables 4.1 summarizes the maple and cherry blossom image data sets. In total, 6,950 geo-tagged maple images are uploaded by 177 unique photographers, while the other 319 photographers contribute 4,576 geo-tagged cherry blossom images.

4.6.2 Parameter Selection:

We list all the assignments of the parameters used in our solution.

The tree-based graph H in Definition 4: For spot clustering, both density based clustering algorithm (e.g., [78]) and agglomerative methods like hierarchical clustering can be used for building the graph H. Since the density based clustering would filter out a few sparsely distributed points that may be obscure, we utilized a centroid linkage based hierarchical clustering algorithm[§]. We set the smallest cluster size to $2km$ (i.e., leaves of H), which enables us to find some obscure sightseeing spots that exist in famous sightseeing areas (see Figure 4.9d).

The parameters in Algorithm 2: We applied our biased authority propagation algorithm in section 3.3.1 by setting the decay factor $\alpha_B = 0.85$, and the maximum number of iterations $M_B = 1000$.

[§]Open source software: scipy.cluster.hierarchy

Table 4.2. Details of the cherry blossom dataset and the maple tree dataset.

	Cherry blossom dataset	**Maple tree dataset**
Number of geo-taged images	4,576	6,950
Number of Flickr users	319	177
Number of clustered spots in H	368	582

The ws in Eq. 3.2, 4.1 and 4.5: Since both w_i in Eq. 3.2 and w_y in Eq. 4.1 have the same meaning of characterizing information staleness, on the basis of the actual effectiveness, we set them to the same exponential decay: $0.8^{(\lambda - y)}$. We set the w_c in Eq. 4.5 to 1.0, which means that the *SAS* emphasizes more on the beauty of scenery rather than the attractiveness of images.

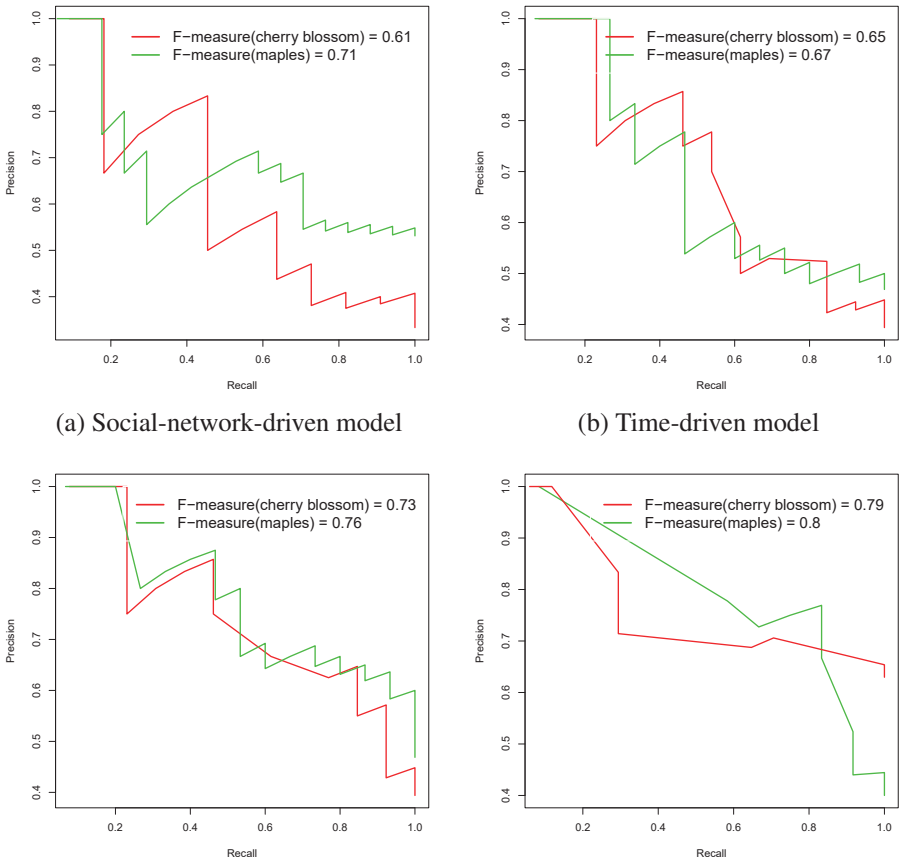
4.6.3 Experiment on Location Popularity Estimation

In this subsection, we try to evaluate the performance of estimating location popularity, i.e., Eq. 4.2. As have been mentioned above, it also can be regarded as the indirect test of the proposed user-profiling models in Chapter 3. In this sense, this indirect test is conducted to verify whether the user-profiling models would help to discover both popular and obscure locations in a city.

User Queries. In accordance with the two user queries $q_1 =$ "$Kyoto, Maples$" and $q_2 =$ "$Kyoto, CherryBlossom$", 929,403 geo-tagged images were crawled from Flickr. By using Flickr's keyword-based search and our labeling component, 4,576 geo-tagged images of cherry blossoms in Kyoto and 6,950 geo-tagged images of maple trees in Kyoto were labeled.

Table 4.2 summarizes the two target datasets. In total, 4,576 cherry blossom images were uploaded by 319 Flickr users, while the other 177 users contributed 6,950 images of maple trees. By generating the leaves of the tree-based graph H (see Definition 4), we obtained 368 cherry blossom locations and 582 maple locations in Kyoto.

Ground Truth. Before analyzing the effect of our application, we first invited three residents who had lived in Kyoto for more than 20 years to label the discovered locations using a five-point scale ranging from "0" (not famous) to "4" (very famous). To reduce their workload, we selected 33 cherry-blossom candidate locations and 32 maple candidate locations, for which 17 and 18 obscure locations were returned, respectively, by regarding the average scores of the three respondents with a relevance



(a) Social-network-driven model

(b) Time-driven model

(c) Ensemble 1: SVM

(d) Ensemble 2: DWE

Figure 4.6. Precision-recall curves for discovering obscure cherry blossoms and maple trees in Kyoto. Each query's F-measure is also computed and shown in this figure.

score of no more than 2.0.

Experimental Results. Because the location-driven model cannot be applied alone, Figure 4.6 shows the precision-recall curves from Eq. 4.2's experimental results obtained by using the other four user classification methods. The red curve shows the experimental results from the query "Kyoto, Cherry Blossom." The green curve displays the corresponding results from the query "Kyoto, Maples." Precision represents the fraction of discovered spots that are relevant, whereas recall represents the fraction of relevant spots that were discovered. Given these results, we can make the following two conclusions.

Location authority is useful for discovering obscure locations. We noticed that conventional authority-based methods such as "rank-by-visits" and "rank-by-users" can-

not efficiently detect obscure locations. By taking the cases in our application as example, the obscure botanical garden in Figure 4.9b may be wrongly identified as a popular location because of the high accumulation of images taken by locals or professional photographers. Without introducing users' authority information, the obscure spots located in popular areas cannot be found (see the blue tags in Figure 4.9d). Having obtained a high F-measure (0.72 on average), our methods effectively reveal that the "obscurity" is caused by the information asymmetry among different user groups. Thus, our authority-based solution is capable of identifying such groups.

The temporal information used in both our time-driven model and Fischer's method [1] is insufficient. Figure 4.6b shows that the time-driven model (Eq. 3.2) did not outperform the other methods in a real scenario. The reason for this is that only approximately 20% of the Flickr users uploaded and shared several images on the basis of the statistics in Section 4.6.1.

DWE based location popularity estimation obtained the best performance compared with the other user classification models. Insofar as the different models are compatible with differently available information, the ensembles yielded better results, especially the dynamic weighted ensemble shown in Figure 4.6d. Although the time-driven model performed better in the experiments in Section 3.4.3, the other two complementary models are needed for robustness. To draw a conclusion, since the ensembles (i.e., SVM and DWE) are able to deal with various available information, the popularity estimation based on them obtained the better performance.

4.6.4 Experiment on Scenery Quality Evaluation

At last, we discuss the performance of location's scenery quality evaluation. By field visits in Kyoto, we first collect ground truth for the evaluation. Then, by comparing with the baselines of ranking POIs proposed in related work, we present the experimental results using NDCG metric [79].

Ground Truth. We obtained the ground truth through field visits conducted respectively in November 2013 and April 2014. In the field visits, for each of the selected 32 maple spots and 33 cherry blossom spots, we invited at least three subjects to perform field evaluation. Despite a total of the 27 subjects are not experts on parks or tourism, using Google Nexus 7 we asked them to record their visiting experiences on Twitter as well as take geo-tagged photos at each spot. Verified by these logs, we can say that the human evaluation is credible.

Table 4.3. Two criteria of calculating scenery ranking scores.

Criteria	Explanations	Ratings
v	Is the spot worth visiting or not.	0, 1, 2, 3
r	Is the spot worth recommending or not.	0, 1, 2, 3

After that, they answered a questionnaire about their opinions of the spots they had visited. On the basis of the feedback, we defined two kinds of graded relevance to denote how well a discovered obscure spot meets tourists' requirements shown in Table 4.3. These two relevance criteria are denoted as "v" and "r", respectively.

To the best of our knowledge, this is the first effort to rank sightseeing spots on the basis of a scenery object's quality. Although the goals of the methods are not the same, we compare other 4 baseline methods with the *SAS* (i.e., Eq. 4.5) and *PAS* (i.e., Eq. 4.6):

- Image Attractiveness (IA): the image attractiveness ranking model proposed in [73];
- HITS based Authority Analysis (HAA): the user(hub)-location(authority) graph based analysis in [32];
- Rank-by-Visits (RV): the accumulative counts of visits to the spot;
- Rank-by-Users (RU): the accumulative counts of unique users who visited the spot.

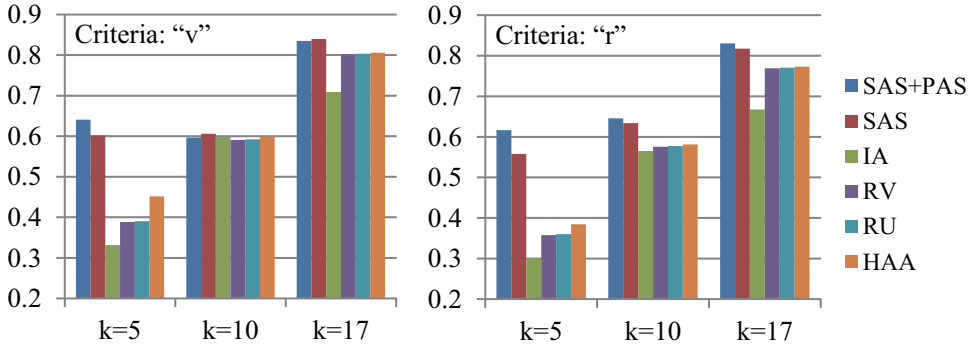
By implementing these methods, we rank the 17+18 obscure spots mentioned above.

NDCG based Evaluation. Figure 4.7a and 4.7b, respectively, depict the NDCG [79] of different measures on the two queries. From the data shown in these figures, we can obtain the following results.

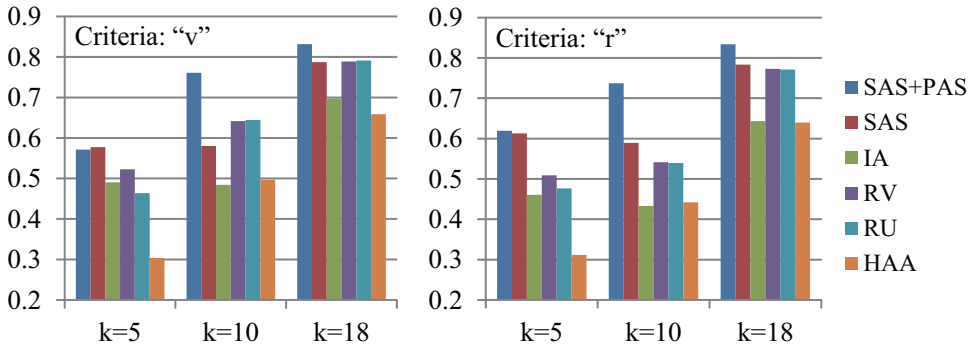
First, in contrast to the baseline methods, our proposed methods, $SAS + PAS$ and *SAS*, performed more effectively in the scenery quality based obscure spot ranking.

Second, the *PAS* brings a significant improvement to the ranking only based on *SAS*. Since ranking obscure spots of having few images is very challenging, our experiments verified that the photographer attention based analysis could provide a stable effectiveness improvement.

Third, by comparing Figure 4.7a and 4.7b, the authority based analysis, *HAA*, fluctuated considerably. This series of conventional methods heavily depend on how much



(a) NDCG on "Kyoto, Maples"



(b) NDCG on "Kyoto, Cherry Blossom"

Figure 4.7. NDCG values of the top-*k* ranking results evaluated by criteria "v" and "r".

the corresponding raw data obtained. In this sense, we have proved that the conventional authority analysis based methods could only be applied to evaluate hot or famous spots.

Finally, the results related to *IA* have verified our hypothesis that an attractive image doesn't mean beautiful scenery included in this image. As naive methods, *RV* and *RU* both performed better than expected, because our system has pre-filtered out the noises (like crowded spots of having no target scenery) in the first step of ranking.

4.7 Application Scenario: Discovering Obscure POIs

In this section, we describe a real application for discovering relatively unknown POIs by utilizing the proposed location profiling models. Unlike conventional studies that discover famous or popular POIs using a voting approach [44, 45, 20, 3], because of the lack of related raw data on the Internet, our application discovers POIs that are not

well known by analyzing information asymmetries among different user groups. This application well demonstrates the usefulness and novelty of the proposed models. In addition, because of the limited face-to-face access among social-media users scattered around the globe, we could not obtain a complete ground truth for the above evaluation in Section 4.6. Therefore, we conducted further visualization to present whether the proposed location profiling methods would help our application to discover obscure locations in Kyoto.

As mentioned in [9], obscure POIs are potential sightseeing resources. By dividing sightseeing locations into four quadrants on the basis of their "popularity" and "sightseeing quality" (see Figure 4.1), this application aims to discover obscure POIs—i.e., those located in the quadrant with high sightseeing quality and low popularity. Obscure sightseeing locations are pertinent to in-depth travel, not only for enjoying the beautiful scenery but also for experiencing local culture, especially with repeat tourists who have already visited the more popular locations.

In accordance with these two dimensions, "popularity" and "sightseeing quality", two main functions are implemented in our application: discovering obscure locations and determining whether they are worth visiting. In its current form, the application collects UGC from Flickr and supports user queries consisting of a target city c and scenery object o (e.g., "Kyoto, Maples"). The details for this application are as follows.

4.7.1 System Architecture

Figure 4.8 shows the architecture of our system, which is comprised of the following three components: Data Construction, Location Profiling, and Recommendation. For each component, a brief introduction is as follows.

Data Construction. Stored in the *GeoImage* and *PProfile* datasets, our application first collects geo-tagged images and user profiles from Flickr using its abundant APIs. Then, the Data Construction component extracts the necessary information for further analysis. Namely, from the raw collected data, three data structures are extracted, i.e., social network $N(V,E)$ (see Definition 1), visiting frequency matrices M_i (see Definition 2) and tree-based location graph H (see Definition 4).

Location Profiling. In the second component, we apply our proposed methods to do location profiling. Specifically, the Flickr users are first classified into different user groups on the basis of their location authority to the target city. Then, by calculating the information asymmetry between these two user groups, the system estimates the

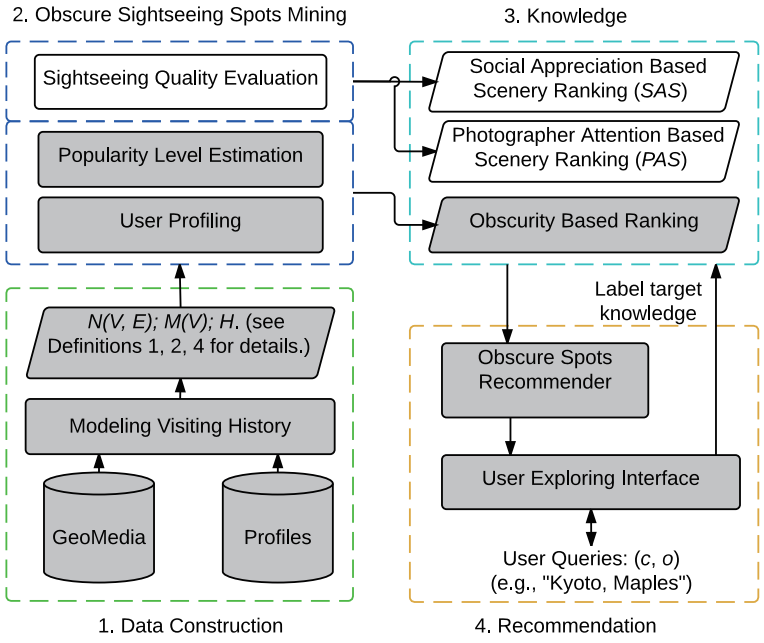


Figure 4.8. The architecture of our system. It is comprised of three components, i.e., data construction, obscure sightseeing spots mining and recommendation.

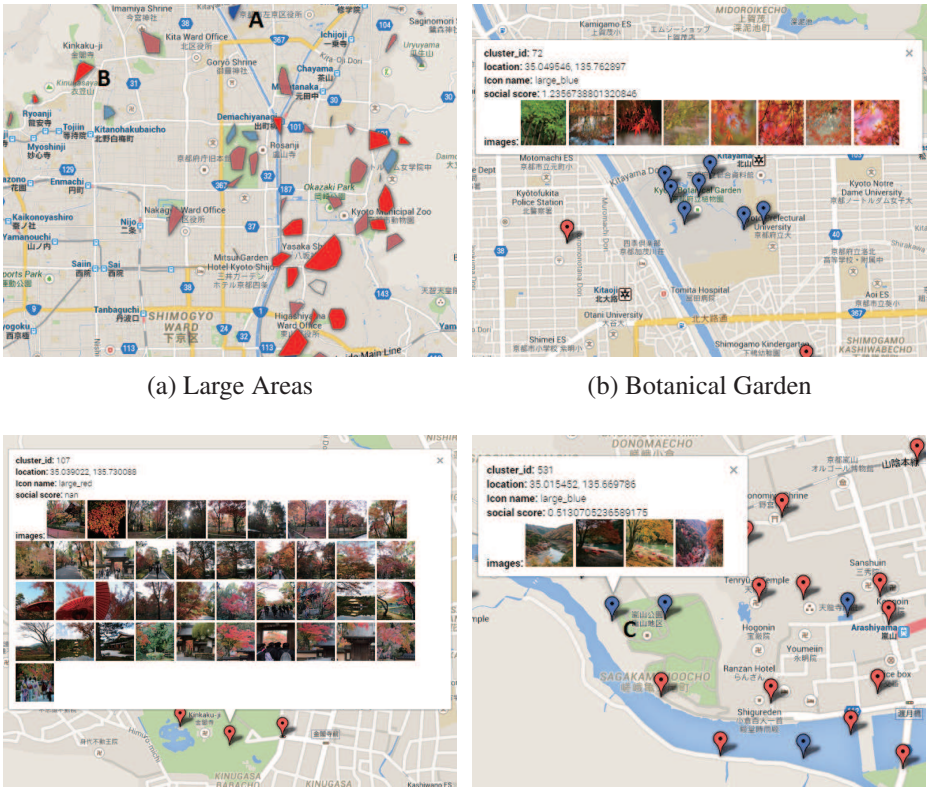
popularity of all the locations where the users have been to. At last, given the scenery object o, the system evaluates the scenery quality of each location. A ranking list will be obtained.

Recommendation. By utilizing the Google Map APIs, the system also has an interface for interactions with end users. The interface can accept user queries specifying a city and a scenery object, like "Kyoto, Maples". Once a query is accepted, the system will present all the candidate locations on the Google Map. Next, we will visualize results of a query.

4.7.2 Exploring POIs on Map

Given the query "Kyoto, Maples," the application first automatically labels all related geo-tagged images of maple trees. A list of locations with maple trees in Kyoto is then displayed on a web-based map. By changing the zoom level of the map, users can explore popular and obscure locations at different geographical granularities.

Figure 4.9 shows the interfaces with multi-level representations of maple locations. Figure 4.9a shows the initial results. From deep blue (very obscure) to deep red (very



(a) Large Areas (b) Botanical Garden

(c) Kinkaku-ji (d) Arashiyama

Figure 4.9. Interfaces for the proposed application, in which Google Maps' APIs are utilized. From deep blue to deep red, the color of each location represents how popular (or obscure) it is for sightseeing.

popular), different colors indicate the obscurity levels of areas with maple trees. By zooming in on the map, Figures 4.9b and 4.9c show the Botanical Garden (i.e., an obscure spot labeled "A" in Figure 4.9a) and the Kinkaku-ji (i.e., a very popular spot, labeled "B" in Figure 4.9a), respectively. By clicking any tag on the map, users can view the corresponding images taken there.

In Figure 4.9d, some small obscure spots exist, even in a famous sightseeing area such as Arashiyama. The obscure maples in Arashiyama are located around parking lots and on mountains that are almost inaccessible to most foreign tourists. In this sense, we can conclude that more detailed user profiling is needed in order to discover fine-grained POIs. As show in Figure 4.9, the application also gives each location a scenery quality score, which jointly considers the social-appreciation-based ranking and the photographer-attention-based ranking proposed in Section 4.5.

The study presented here is among the first of its kind and aims to inspire more interests in how to find diverse sightseeing resources on the basis of traveling experiences shared on the Web. Our work shows that the knowledge mined from location profiling can be used to discover obscure sightseeing locations in a city. For future work, the multifaceted nature of the problem of selecting sightseeing locations would be considered. Not only the scenery itself, but also a location's facilities, traffic, or culture values could be further considered. In addition, by considering continuous geo-tagged photo trips, constructing obscure travel routes is a potential work.

CHAPTER 5

LOCATION RELATION PROFILING

In the previous Chapter 4, our work focuses on the knowledge mining about a single location. By analyzing the user distribution based location popularity and scenery quality, an application is devised to discover both popular and obscure POIs in a city. In this chapter, we will go further to analyze the relations among different locations. By analyzing the global positioning system (GPS) trajectories, we will proposed a multi-view learning framework for presenting the construction of a new urban movement knowledge graph. In particular, by viewing GPS trajectory data from temporal, spatial, and spatiotemporal points of view, we construct a knowledge graph of which nodes and edges are their locations and relations, respectively. The utility of this knowledge graph will be discussed in detail in Chapter 6.

5.1 Motivation

With the development of mobile devices, the GPS, and Web 2.0 technologies, the explosion of social networking services (SNSs) has led to a wealth of research based on the use of social media content and various social graphs. For example, major SNS companies, such as Facebook, Twitter, and Yahoo Flickr, now incorporate geo-tagged user generated content (UGC) in their geo-social recommendation results.

One problem of significant interest is that of predicting user attention to locations in a city. As one instance of this problem, point-of-interest (POI) recommendation sys-

tems have emerged recently [80]. Unlike conventional recommendation systems that generally require users to express their preferences by explicitly providing ratings for items (e.g., books or movies), POI recommendation systems always observe implicit user-location feedback matrices. In addition, the sparsity of a user-location matrix is much higher than that of a traditional user-item rating matrix. For example, the sparsity of the Netflix dataset [36] is around 99%, whereas the sparsity of the Gowalla dataset [4] is about 99.98%.

In related studies, these problems were addressed, in approximate terms, by considering three key relations: (1) *user-location relation*; (2) *user-user relation*; and (3) *location-location relation*. In terms of the user-location relation, users' travel histories (e.g., check-in behavior) have been well investigated to predict their preferences [81] [10]. For the user-user relation, the physical distances between users or friendship relations (e.g., online social networks) are considered, e.g., [82] [14]. However, few studies have comprehensively taken the location-location relation into consideration.

The relations among locations can be explained in different ways. In [82], the authors first propose a POI recommendation method by constructing a location similarity network. The state-of-the-art recommendation method [2] incorporated the physical distances between locations to significantly improve performance. However, considering only the similarity and physical distance is insufficient. For example, the typical *home* $\leftrightarrow$ *work* relation [4] would be filtered out because of a relatively long distance. Even worse, a user living in residential location A would rarely pay attention to a similar residential location B. As a result, in addition to learning user preferences [81] [10], a well-structured location-location graph is necessary.

In this work, a matrix-factorization-based framework is proposed to construct a comprehensive urban movement knowledge graph for embedding the location-location relations. Our method analyzes the observed GPS points from temporal, spatial, and temporal-spatio views to jointly learn the latent representations of locations (nodes on the graph) and relations (edges on the graph). Furthermore, to extract meaningful relations (lifestyles), two new unsupervised relation extraction methods are proposed. The utility of our knowledge graph is verified by introducing an application to predict user attention to locations on a real-world dataset. An encouraging result is obtained, thereby indicating our graph encodes deep insight into the location-location relation.

Overall, the main contributions of this work are summarized as follows:

1. From an application standpoint, we construct a comprehensive location-location knowledge graph for urban computing. In Chapter 6, we will verify its utility

by applying it to predict user attention to locations in a city. Encouraging results are obtained in our experiments.

2. From a theoretical standpoint, two new unsupervised relation extraction methods are devised in Section 5.4.3. For label-free learning tasks, we proposed a new way for encoding multi-view prior knowledge into loss functions.

The remainder of this chapter are as follows. Since our work of constructing location-location knowledge graph is among the first of its kind, Section 5.2 briefly talks about some related relation extraction work in the area of Nature Language Processing. Section 5.3 defines the problem about location-location knowledge graph construction. Section 5.4 introduces our learning framework of constructing the knowledge graph. By viewing GPS trajectory data from three different views (i.e., temporal view, spatial view, and spatiotemporal view), four totally different methods are devised. In Section 6.5.6, we will visualize some location relations obtained from the knowledge graph. Please note that, the verification of this knowledge graph will also be discussed in detail in Chapter 6.

5.2 Related Work

Although a lot of related work have been mentioned throughout the whole chapter, in this section, we in advance explains the major differences from the conventional relation extraction methods on existing knowledge graphs.

Relation extraction. A survey [83] revealed that most of the methods attempting to use machine learning (e.g., TranE [84], TranH [85], and TranR [86]) are trained on existing knowledge graphs (e.g., Freebase, Wikidata, and YAGO2) in a supervised manner and then used to predict new edges on those graphs. Similar to our methods, we list some latent feature models for comparison. In [87], a tensor factorization based model, RESCAL, is proposed for describing the pairwise interactions of latent features. Recently, several NN-based models were proposed, e.g., [88] [89], to reduce the number of parameters used in RESCAL. NNs can learn to position semantically similar nodes close to each other even if they are not explicitly trained to do this [90]. For our framework, which is an NN matrix factorization ensemble, we devised a novel NN matrix factorization method for unsupervised relational learning, rather than using supervised graph completion.

5.3 Preliminary

We first define the input (Trs) and output (G) of our problem, and then provide an overview of our work.

Definition 5. *(GPS trajectory): A trajectory Tr is a sequence of GPS points, i.e., $p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_n$. Each point p consists of {id, datetime stamp, longitude, and latitude}.*

Definition 6. *(Urban movement knowledge graph): The knowledge graph is denoted as $G(V,R)$, where V is the location set of v_i, and R is the set of relations $r(v_i,v_j)$ between v_i and v_j. After learning, each v and r are represented in a vector format.*

From the GIS standpoint, a location v is a polygon, which is a closed shape defined by a connected sequence of {longitude, latitude} coordinate pairs. Therefore, "location" could be a landmark, Point of Interest (POI), small spot, large area, etc.

Unlike the conventional supervised relation prediction on existing knowledge graphs (e.g., Freebase) [83], we face a new challenge of extracting location-location relations in an unsupervised manner. To solve this problem, in Sections 5.4.1 and 5.4.2, we first mine relations' context knowledge from temporal and spatial views, respectively. Then, in Section 5.4.3, by jointly considering the spatiotemporal context knowledge, we proposed two novel methods for unsupervised relation extraction.

5.4 Knowledge Graph Construction

5.4.1 Temporal View-based Construction

By mapping all the points p in Trs to locations V, we could easily obtain a time series matrix $S \in \mathbb{R}^{|V| \times h}$, where $|V|$ and h are the number of locations and hours, respectively. When $h = 24 \times 7$, for each row of S, it aggregates the number of visits to the target location (i.e., points in Trs) by hours of the week. When $h = 24 \times 2$, for each row of S, it aggregates the number of visits using workday hours and holiday hours. These are the two main time windows used in urban computing [91] [4].

Figure 5.1 shows three rows of the normalized S in our experiment. The data points were calculated using taxi trajectories. A residential location v_1 is easy to identify by rising workday morning and evening visits. The number of visits to a popular shopping location v_2 increases at 9am and lasts until midnight. The curve of a sightseeing

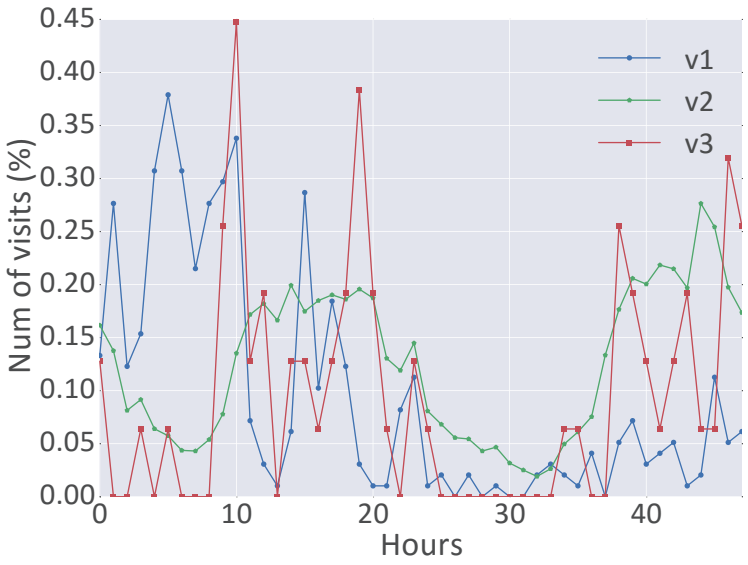


Figure 5.1. Visual representation of S. On the x-axis, $[0 \sim 23]$ and $[24 \sim 47]$ are workday and holiday hours, respectively.

location v_3 even reveals the hours during which it is open (between two daily peaks). This observation is consistent with earlier research suggesting that locations containing large dynamic populations are readily visible in comparatively coarse wireless-traffic analyses [92].

In accordance with the observation, to distinguish each location v, we represent all the locations using k_t common components $E \in \mathbb{R}^{k_t \times h}$ (also known as bases in Nonnegative Matrix Factorization). As a result, the locations' temporal view-based latent representations $V_t \in \mathbb{R}^{|V| \times k_t}$ are learned by the following Nonnegative Matrix Factorization:

$$\min_{E \geq 0, V_t \geq 0} \|S - V_t E\|^2 + \lambda_r \left[\|V_t\|^2 + \|E\|^2 \right], \tag{5.1}$$

where $\|\cdot\|^2$ denotes the Frobenius norm. The latter two entries of Eq. 5.1 are regularizers for penalizing the norms of V_t and E. After the first step, the knowledge graph $G(V,R)$ is represented as $G(V_t, R = \varnothing)$, where R is an empty set.

5.4.2 Spatial View-based Construction

We capture the spatial influence from Trs, in the second step, by first creating a transition matrix $T \in \mathbb{R}^{|V| \times |V|}$. Similar to the transition matrix defined in a Markov chain,

an entry $T_{i,j}$ denotes the probability of a transition (movement) from location v_i to v_j. Creating such a matrix for Trs is straightforward. For each $Tr = p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_n$, using the longitude and latitude information, we first map it to the locations: $Tr^{'} = v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_m$. Then, by aggregating all the $Tr^{'}$s, we obtain the initial transition matrix $T^{'}$, the entry $T^{'}_{i,j}$ of which records the number of movements $v_i \rightarrow v_j$. Lastly, T is equal to the normalized $T^{'}$.

Given the transition graph T, for each location v, we embed the structured information of T by a factorization of T:

$$T \approx V_s M V_s^T, \tag{5.2}$$

where $V_s \in \mathbb{R}^{|V| \times k_s}$ is the embedding matrix, $M \in \mathbb{R}^{k_s \times k_s}$ is the interaction matrix and k_s specifies the number of latent features. Similar to Eq. 5.1, the latent matrices V_s and M are computed by solving the following optimization function:

$$\min_{M \geq 0, V_s \geq 0} \left\| T - V_s M V_s^T \right\|^2 + \lambda_r \left[\|V_s\|^2 + \|M\|^2 \right]. \tag{5.3}$$

Eq. 5.2 can be regarded as a model for embedding the graph structure. Namely, the latent matrix V_s is an embedding of the locations in a latent component space, where the similarity of locations in this space reflects their similarity in the domain of graph T. M models the interactions of the latent components in this space.

After the second step, the knowledge graph $G(V,R)$ is represented as $G([V_t;V_s],T)$. The locations V are described by concatenating V_t and V_s. For relations R, T is assigned indicating the mutual dependencies among different locations.

5.4.3 Spatio-temporal View-based Construction

In the third step, we face the challenge of labeling edges on the transition graph T. Given the current knowledge graph $G([V_t;V_s],T)$, although we can observe which locations are tightly related, we do not know why this occurs. In the final step, we perform the relation extraction task in an unsupervised manner. In other words, we try to divide the graph T into several sub-graphs, $T_1, T_2, ..., T_r$, each of which represents one kind of relation. The relations can be regarded as different lifestyles (moving patterns) in a city.

Although relation extraction is a extensively investigated research topic [83], to the best of our knowledge, few work have attempted to perform the extraction in an unsupervised manner (i.e., without labeling any edges of T). Two new methods are proposed in our framework.

(1) Selectional Preference-based Method

In natural language processing, the term *selectional preference* denotes the tendency of a word to co-occur with words that belong to certain lexical sets. For example, the adjective *delicious* prefers to modify nouns that denote food. Accordingly, we assume that relations $\forall r \in (v_h, r, v_t)$ also have selectional preferences. Intuitively, a relation r allows only certain location types in its head-argument slot and tail-argument slot. As a result, a *selectional preference* bridges the observed time series matrix S and transition graph T.

In this method, for each location, we draw a latent vector $v \in \mathbb{R}^{1\times k_{ts}}$. For each relation, we draw a latent vector $r \in \mathbb{R}^{1\times arity(r)*k_{ts}}$, where function $arity(r)$ returns the number of argument slots of r. Since in our case r is a binary relation (there is a head-argument slot for head location v_h and a tail-argument slot for tail location v_t), $arity(r) = 2$. Hence, a relation can be represented in a concatenating way: $r =< d_h, d_t >$, where $d \in \mathbb{R}^{1\times k_{ts}}$ denotes an argument slot.

As a result, measuring the compatibility of a tuple (v_h, r, v_t) (i.e., the edge weight of T) amounts to summing up the compatibility between the representations of each location (v_h, v_t) and the corresponding argument slots (d_h, d_t). Then, we have the following objective function:

$$Obj_1 = \min_{V_{ts}, D\geq 0} \left\| T - (V_{ts}D_h^T + V_{ts}D_t^T) \right\|^2, \tag{5.4}$$

where $V_{ts} \in \mathbb{R}^{|V|\times k_{ts}}$ records all the locations v and $D_* \in \mathbb{R}^{|V|\times k_{ts}}$ record all the argument slots d_h, d_t, respectively.

By jointly considering the spatial and temporal influences, i.e., Eq. 5.4 and Eq. 5.1, the selectional preference-based relation extraction method minimizes the following function:

$$\begin{aligned} \min_{E, V_{ts}, D\geq 0} & Obj_1 + \lambda_s \left\| S - V_{ts}E \right\|^2 + \\ & \lambda_r \left[\|V_{ts}\|^2 + \|E\|^2 + \|D_h\|^2 + \|D_t\|^2 \right], \end{aligned} \tag{5.5}$$

where V_{ts} is shared in the first and second entries. After learning, a hierarchical clustering method is utilized for all candidate relations $< d_h, d_t >, d_h \in D_h; d_t \in D_t$. A cluster corresponds to one kind of relation.

However, in the context of unsupervised learning, we find that the current method is only suitable for 1-to-1 relations. In other words, for an N-to-1 relation r of which the head could be multiple instances, they most likely will be divided into different clusters. Next, an improved method is devised.

(2) Neural Network (NN)-based Method

The most important advantage of this method is that, in addition to the latent variables, the neural network (NN)-based method also learns the objective function rather than assuming it to be fixed. Compared with the fixed Obj_1 in Eq. 5.4, we define the NN-based objective function:

$$\begin{aligned} Obj_2 &= \min_{V_{ts},I,R_{ts}\geq 0} \sum_{\forall(v_h,r,v_t)\in T} (T_{ht} - f_\Phi(\vec{t}))^2 \\ f_\Phi(\vec{t}) &= W_2^T \, Sigmoid(W_1 \vec{t}^T) \\ \vec{t} &= [v_h; v_t; I_{(v_h,r,v_t)}(R_{ts}C)] \end{aligned} \tag{5.6}$$

where $V_{ts} \in \mathbb{R}^{|V|\times k_{ts}}$ records all the locations; $R_{ts} \in \mathbb{R}^{|R|\times k_r}$ records all the relations; I is a latent indexing matrix; $C \in \mathbb{R}^{k_r \times k_r}$ is a constant sparsity matrix [93]; $f_\Phi(\vec{t})$ denotes a feed-forward NN; $W_1 \in \mathbb{R}^{n\times(2k_{ts}+k_r)}$ and $W_2 \in \mathbb{R}^{n\times 1}$, respectively, represent the weights of its first and second layers; n is the number of hidden neurons; $Sigmoid()$ is an element-wise sigmoid function.

For each tuple (v_h, r, v_t), an indexing vector $I_{(v_h,r,v_t)} \in \mathbb{R}^{1\times|R|}$ is introduced to specify a relation in R. The sparsity of $I_{(v_h,r,v_t)}$ is increased as much as possible by introducing a constant sparsity matrix $C = (1-\theta)eye(k_r) + \frac{\theta}{k_r} ones(k_r)$ [93], where $eye(k_r)$ is an identity matrix of size $k_r \times k_r$, $ones(k_r)$ is a matrix of size $k_r \times k_r$ with all entries of 1s, and θ is the sparsity parameter. When we multiply C with R_{ts} in Eq. 5.6, to compensate for the loss of sparsity in R_{ts}, $I_{(v_h,r,v_t)}$ becomes sparse. An extreme example is that, there is only one entry in $I_{(v_h,r,v_t)}$ close to 1, whereas the other entries approximate 0. The introduction of I allows a relation to have multiple head or tail instances. As a result, this method can accommodate N-to-1, 1-to-N, and N-to-N relations.

Thus, the NN f_Φ has $2k_{ts} + k_r$ real-valued input units and one real-valued output unit. The first $2k_{ts}$ inputs are head- and tail-specific features, respectively, whereas the last k_r inputs are relation-specific features. As mentioned above, in addition to learning the latent variables V_{ts}, I, and R_{ts}, this method further learns the objective function, i.e., f_Φ.

Similar to Eq. 5.5, the NN-based method is defined as:

$$\begin{aligned} \min_{E,V_{ts},I,R_{ts}\geq 0} & Obj_2 + \lambda_s \|S - V_{ts}E\|^2 + \\ & \lambda_r \left[\|V_{ts}\|^2 + \|E\|^2 + \|R_{ts}\|^2 + \|I\|^2 \right]. \end{aligned} \tag{5.7}$$

According to the survey [83], feed-forward NNs have already been applied in state-of-the-art supervised relation extraction tasks (e.g., [89]) and the KV project [94]. By

observing the context information of head and tail locations (i.e., S), our work goes further by applying an NN in an unsupervised relation extraction task. Because of the NN matrix factorization ensemble, we refer to Eq. 5.7 as NN matrix factorization. Our method is consistent with the proposal in [95]. That is, for a label-free learning task, we are leveraging prior knowledge (S, T, and selectional preference) to restrict the space of possible functions specifying the hypothesis class (the NNs f_Φ with different weights in our case). In this context, our NN matrix factorization provides a new way for encoding prior knowledge into loss functions.

Intuitively, by proposing these two relation extraction methods, two constraints are satisfied: (1) *syntactic constraint* (Eq. 5.4 and 5.6) for eliminating incoherent extractions and (2) *lexical constraint* (the second entries in Eq. 5.5 and 5.7) for constraining certain location types in a relation's head- and tail-argument slots. The final knowledge graph can be represented as $G(V_{ts}(1), R_{ts}(1))$ and $G(V_{ts}(2), R_{ts}(2))$, which are learned by Eq. 5.5 and Eq. 5.7, respectively.

To draw a conclusion about this chapter, given GPS trajectories, we first proposed a novel framework for constructing an urban movement knowledge graph that embeds temporal and spatial information. Two new unsupervised relation extraction methods were then devised to learn the relations among nodes on this graph.

However, there is no quantitative analysis in this chapter to support the utility of our knowledge graph. Accordingly, in the next chapter, an experiment on a real-world dataset will be conducted. Compared with state-of-the-art baselines, our location relation knowledge graph shows significant improvements. Furthermore, knowledge graph visualization will also be presented in the next chapter.

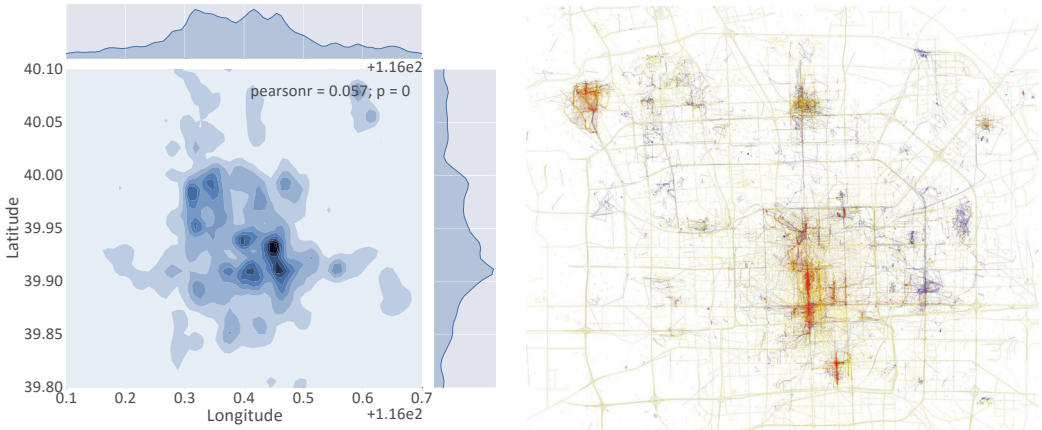
CHAPTER 6

USER-TO-LOCATION ATTENTION PREDICTION

The introduction of this chapter is three-fold: first, to claim that predicting user attention to locations is a key problem in social informatics research area; second, to verify the utility of the learned knowledge graph *G*s in Chapter 5; and third, to provide a quantitative analysis on *G*s. Therefore, in this chapter, we first present the motivation by introducing some observations from the real world. Then, two new solutions are proposed to predict the extent (high or low) of user attention to locations, one of which utilized the knowledge graph obtained in Chapter 5. At last, a comprehensive discussion will be presented on the basis of experimental results.

6.1 Motivation

Benefiting from the explosion of social network services (SNSs), a wealth of studies have been conducted using online social media content and various social graphs. Furthermore, heterogeneous offline data generated by a diversity of sources in urban spaces has led to a wider research area into solving real-world social issues. For example, a series of urban computing applications have been enabled [8]. Moreover, in order to meet the diverse requirements of users, significant effort has been dedicated to developing new applications.



(a) Check-in behavior (b) Photography behavior

Figure 6.1. User attention dynamics. (a) shows the density estimation of our check-in dataset in Beijing. (b) shows [1]'s statistical results of the mobility behavior of different photographer groups in Beijing. Red: tourists; blue: locals; yellow: either.

In this context, one problem of significant interest is that of *user attention estimation*, i.e., predicting user attention, high or low, of each {user, location} pair in a city. In our work, by analyzing the observed collective user mobility behaviors that are the results of user attention, unobserved ones can be predicted.

Figure 6.1 shows a motivating example. The density estimation in Figure 6.1a indicates that the user attention dynamics absolutely exists in a city. That is, some of the regions are attracting more users while some of the others are not. For these obscure regions, they may indicate that (1) users do not like the regions; or (2) with the rapid development of the city, users do not know about them; or (3) as a realistic problem, we have not collected a complete body of raw data. Figure 6.1b further shows that users from different groups pay attention to different regions of a city. It is clear that some regions appeal to more users from the tourist group (in red), while others attract many users from the local group (in blue). As a result, there exist a challenge to effectively model the user attention dynamics in a city.

In this chapter, we proposed two solutions to predict user attention to different locations in a city. The first one, TMPMF, directly utilizes the transition matrix T (please see Section 5.4.2 for details) for prediction, whereas the second one, KGPMF, utilizes the knowledge graph Gs obtianed in Chapter 5.

TMPMF. In this solution, we propose a flexible algorithm, called Transition Matrix based Probabilistic Matrix Factorization (TMPMF), which can deal with different

kinds of dependencies among locations. In this algorithm, a transition matrix (i.e., T) that corresponds to a particular dependency is constructed to encode the relationships among locations. For example, if a nearest neighbor graph is constructed, our algorithm is instantiated to the model proposed in [42] which considers the "similarity" relationships among items. To the end of predicting user attention to locations, we introduce the collective transition matrix T to encode the "complementary" relationships among locations. Intuitively, we define the "complementariness" as the latent purpose of transition. It is a natural assumption that for a user, given a transition from location A to location B (e.g., by bus, by taxi, etc.), there is an inherent property that describes his/her purpose which can be fulfilled not in location A but B. Accordingly, by introducing the collective transition matrix, our algorithm learns different latent transition purposes. We also develop an optimization scheme to solve the new matrix factorization objective function that incorporates the graph structure into it.

KGPMF. In the second solution, a Knowledge Graph based Probabilistic Matrix Factorization (KGPMF) method is proposed. Unlike TMPMF model that directly utilizes the transition matrix T, KGPMF tries to enrich the latent location features by embedding supplementary information from knowledge graph Gs. By comparison between the models, i.e., TMPMF and KGPMF, we are going to explain why the knowledge graph is necessary for such a social problem.

Both of the models can be used to predict the user-attention of the unobserved $\{$user, location$\}$ pairs. Also, in the experiments, we find that these models can mine the collective moving patterns (lifestyles) in a city. Unlike the related works [17] [18], without using any semantic information (e.g., categories of POIs), our models further provide a complementary way to learn the functional compositions of urban spaces.

In summary, the work in this chapter makes the following major contributions:

- While standard MF techniques rarely consider mutual dependencies among items, we devise flexible algorithms that incorporate them as additional latent regularization terms. Thus, by constructing different kinds of dependency graph, our algorithm is applicable in different problem domains. The proposed method is a general one that can leverage the power of MF.

- By utilizing our algorithms, we propose a new application to predict the user attention to different locations in a city. By jointly considering the locations' user-attention information and a city's traffic information (e.g., by bus or taxi), both collective moving patterns and locations' functionality can be learned. It

can further help in different application scenarios, including geo-social recommendation, user profiling, urban planning, personal digital assistant (PDA), and so on.

- We evaluate the proposed algorithms with experiments in a real world problem. Encouraging results are obtained.

The remainder of this chapter is as follows. Section 6.2 shows the related work that aims to solve similar social problems by enriching latent location or user features. In addition, from a technical point of view, we also list some related work about MF. Section 6.3 and Section 6.4 present the implementations of TMPMF and KGPMF, respectively. In Section 6.5, the experimental results are presented.

6.2 Related Work

Enrichment of latent user features. A number of conventional location recommendation models relying on users' visiting history, geographical influence, temporal influence, etc., have been proposed to infer users' location preferences. For instance, HMM [21], periodic GMM [4], and BPP [96] have been proposed to model individual mobility behavior for preferences inferring. A user preference prediction method investigating the context of the location was proposed [81]. They assumed a user's visit to a location is not only influenced by the set of locations they previously visited, but also by the locations they subsequently visited. Recently, several studies [10] [97] focused on predicting the places users are expected to visit next, based on their previously generated content. Nevertheless, these related studies mainly focus on enriching the latent user features, i.e., stage 1 in our Algorithm 6.

Enrichment of latent location features. Few studies have considered the enrichment of latent location features. One reason may be that, in most of the aforementioned POI recommendation models based on the enrichment of latent user features, they assumed that POI categories (i.e., V_t in our case) are observed. However, it is insufficient. Some initially believed [82] that location similarity can also influence recommendation. They constructed a similarity network according to their geo-distance and categories. Then, the GeoMF model [2] defined a POI-influence-area matrix based on the geo-distance. They assumed the influence areas of a POI to have a normal distribution centered at this POI. To the best of our knowledge, the enrichment of latent location features has not been well investigated yet. Our framework is among the first

of its kind to systematically construct a knowledge graph G for location oriented enrichment. Not only the categories belonging to the location (i.e., nodes on G) but also the relations among them (i.e., edges on G) are embedded.

Matrix Factorization. From a theoretical viewpoint, very few attempts have been made to take items' mutual dependencies into consideration. To predict the user attention of each {user, location} pair in a city, the matrix factorization (MF) techniques are potential solutions. Existing methods for MF can roughly be categrized into two classes: (1) collaborative filtering (CF) based methods without using user or item content information [98] [99] and (2) hybrid CF based methods by combining content analysis [37] [38]. In [37], the authors apply Latent Dirichlet allocation (LDA) in their CF model to learn the topics of scientific articles for recommendation. In [38], the authors generalize advances in deep learning. Their model jointly performs deep representation learning for content information and collaborative filtering for the user-item ratings matrix. In this sense, by involving the mutual dependencies of items, our algorithms consummates the MF techniques.

Similar to our algorithms, in [42], the authors propose a graph regularized nonnegative matrix factorization method by assuming that if two items x_j, x_l are close (i.e., high weight on the graph), then z_j and z_l, the latent representations of these two items with respect to the new basis, are similar to each other. However, we find that their algorithm is not flexible enough to solve the problem that we encounter. In our problem, two items (i.e., locations) are highly dependent because of "complementariness" rather than "similarity". For example, a working region and a residential region are functionally complementary but not similar. On the contrary, two educational regions are similar but geographically independent. Therefore, a content similarity regularized MF method is not suitable to solve our problem. In other words, the key point of solving our problem is to automatically learn the "complementariness" among locations in a city rather than to use a fixed assumption in advance.

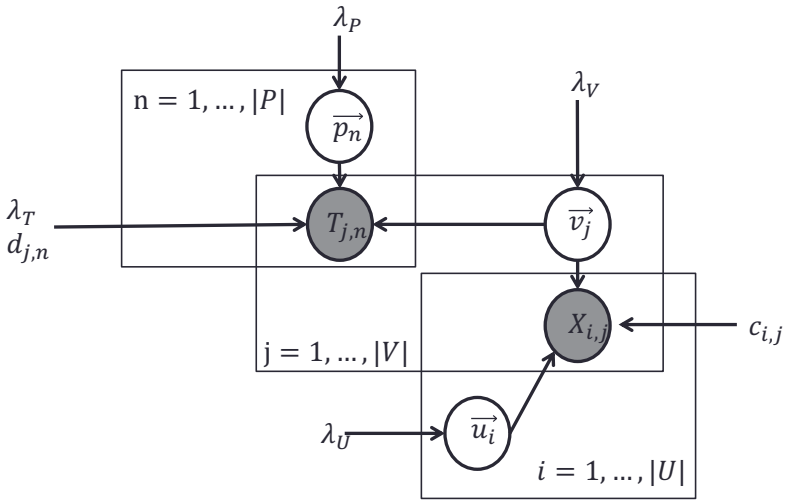


Figure 6.2. The graphical illustration of our Transition Matrix based Probabilistic Matrix Factorization (TMPMF). The gray nodes, X and T, are observed data, while the others are latent variables.

6.3 Transition Matrix based Probabilistic Matrix Factorization

6.3.1 Problem Definition

Definition 7. *Given a user-location attention matrix $X \in \mathbb{R}^{M \times N}$ and a location-location transition matrix $T \in \mathbb{R}^{N \times N}$, TMPMF aims to jointly learn the user, location, and locations' dependency latent spaces (i.e., U, V and P shown in Figure 6.2) whose products can well approximate the original matrices X and T.*

From the viewpoint of our application, the problem can be described as: by observing a part of the user-location attention matrix X^o and locations' mutual dependencies T, we expect to well predict the unobserved attention matrix $X^{\neg o}$.

In our application, we partition a city into $50 \times 50 = 2,500$ grids. Hereinafter, a grid denotes a location. By collecting geo-tagged check-ins, we first construct the user-grid attention matrix X. Then, using the collective transition information (e.g., by bus or by taxi) among different grids, we construct the mutual dependency graph T. In the remainder of this section, we will introduce our algorithm when X and T are given. Detailed information about X and T will be presented in Section 6.5.

6.3.2 Modeling

As shown in Figure 6.2, our algorithm is a generalized hierarchical Bayesian method which jointly learns the user, item, and items' dependency latent spaces. It can be seen that our algorithm to a considerable extent fuses two MF to obtain compact feature representation. Compared with the method proposed in [42], by introducing the additional MF on mutual transition matrix T, a latent space (i.e., P) is introduced to decode different kinds of dependency flexibly.

Although our algorithm is a general one that can be applied in different problem domains, we explain it in the context of our problem for consistency. Table 6.1 shows the notations used for this algorithm. To generate the user-grid attention matrix X, Algorithm 5 uses a generative process. By introducing Bayes' theorem, our algorithm models user attention based on the following observations.

1. **Observation 1:** For a user or user group in a city, we find that they tend to go to a few grids, and that their visits may follow a power-law distribution. This means that users tend to visit only a few grids multiple times (i.e., strong demands) and many grids only a few times (i.e., weak demands). For working people, an intuitive example is the periodic state transition between "home" and "work" [4] [22]. Therefore, by assuming the number of different users' demands is k, we introduce a latent k-dimensional vector $\vec{u}_i \in U$ to represent a particular user or user group's demands.

2. **Observation 2:** Different grids can meet different users' demands. As shown in Figure 6.1b, if we classify users into two groups, some regions appeal to more users from the red tourist-like group, while others attract many users from the blue local-like group. Accordingly, we introduce a latent k-dimensional vector $\vec{v}_j \in V$ to indicate what kinds of demands a grid can meet.

3. **Observation 3:** User demands can be learned from the transition information, i.e., the observed mutual transition matrix T. That is, given a user's transition from the source grid to the destination grid, there is an inherent property that describes his/her demand. It is the result of the functional asymmetry between the two grids. Accordingly, we introduce a latent variable $P \in \mathbb{R}^{N \times K}$, the k columns of which can be regarded as different moving patterns. Each pattern indicates one or multiple user demands.

Table 6.1. Notations used in the algorithm.

Notations	Size	Description
X	$M \times N$	A user-grid attention matrix.
T	$N \times N$	A grid-grid transition matrix.
U	$M \times k$	A user latent space, $\vec{u}_i \in U$.
V	$N \times k$	A grid latent space, $\vec{v}_j \in V$.
P	$N \times k$	A mobility latent space, $\vec{p}_n \in P$.
I_k	$k \times k$	A k-dimensional identity matrix.
k	1	The number of latent user demands.
$c_{i,j}$	1	A confidence parameter for X.
$d_{j,n}$	1	A confidence parameter for T.
$\mathbf{1}_R(x)$		The indicator function $\mathbf{1}_R(x) \rightarrow \{1,0\}$.
$l(x)$		A logistic function $l(x) = \frac{1}{1+exp^{-x}}$.
$\mathcal{N}(\mu, \Sigma)$		Gaussian distribution.

Algorithm 5 User-Attention-Matrix Generative Process.

for each user group $\vec{u}_i \in U$ **do** ▷ **Observation 1**
    draw user group latent vector $\vec{u}_i \sim \mathcal{N}(0, \lambda_U^{-1} I_k)$
end for
for each grid $\vec{v}_j \in V$ **do** ▷ **Observation 2**
    draw grid latent vector $\vec{v}_j \sim \mathcal{N}(0, \lambda_V^{-1} I_k)$
end for
for each observed user-grid pair $(\vec{u}_i, \vec{v}_j) \in X^o$ **do**
    draw user-attention rating $X_{i,j} \sim \mathcal{N}(l(\vec{u}_i{}^T \vec{v}_j), c_{i,j}^{-1})$
end for
for each mobility specific $\vec{p}_n \in P$ **do** ▷ **Observation 3**
    draw mobility latent vector $\vec{p}_n \sim \mathcal{N}(0, \lambda_P^{-1} I_k)$
end for
for each observed transition $(\vec{v}_j, \vec{p}_n) \in T$ **do**
    draw transition graph $T_{j,n} \sim \mathcal{N}(l(\vec{v}_j{}^T \vec{p}_n), \lambda_T^{-1} d_{j,n}^{-1})$
end for

In accordance with Algorithm 5, we define the conditional distribution over the observed user-grid attention matrix X as

$$p(X|U,V,C) = \prod_{i=1}^{M}\prod_{j=1}^{N}[\mathcal{N}(X_{i,j}|l(\vec{u}_i{}^T\vec{v}_j), c_{i,j}^{-1})]^{\mathbf{1}_R(i,j)}, \tag{6.1}$$

where $\mathcal{N}(x|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and $\mathbf{1}_R(i,j)$ is the indicator function that is equal to 1 if user $\vec{u}_i$ has checked in grid $\vec{v}_j$ and is equal to 0 otherwise. $l(x)$ is the logistic function $l(x) = \frac{1}{1+exp^{-x}}$, which bounds the range of $\vec{u}_i{}^T\vec{v}_j$ within $[0,1]$. $c_{i,j}$ is a confidence parameter for rating $X_{i,j}$. If $c_{i,j}$ is large, we trust $X_{i,j}$ more. As we mentioned above, $X_{i,j} = 0$ can be interpreted into two ways–the user u_i is either not interested in grid v_j or is unaware of it. Similar to the strategy in [100], we introduce different confidence parameters $c_{i,j}$ for different ratings $X_{i,j}$.

$$c_{i,j} = 1 + \alpha X_{i,j}. \tag{6.2}$$

In our experiments, setting $\alpha = 1$ is found to obtain good results. We place zero-mean spherical Gaussian priors on user feature vector and grid feature vector:

$$p(U|\lambda_U) = \prod_{i=1}^{M}\mathcal{N}(\vec{u}_i|0,\lambda_U^{-1}I_k), \tag{6.3}$$

$$p(V|\lambda_V) = \prod_{j=1}^{N}\mathcal{N}(\vec{v}_j|0,\lambda_V^{-1}I_k). \tag{6.4}$$

On the basis of Observation 3, we then define the conditional distribution over the grid-grid transition matrix T. Similar to the user-grid matrix X, we have

$$p(T|V,P,\lambda_T,D) = \prod_{j=1}^{N}\prod_{n=1}^{N}\mathcal{N}(T_{j,n}|l(\vec{v}_j{}^T\vec{p}_n), \lambda_T^{-1}d_{j,n}^{-1}), \tag{6.5}$$

where P denotes latent moving patterns (user demands), which also have a zero-mean spherical Gaussian prior:

$$p(P|\lambda_P) = \prod_{n=1}^{N}\mathcal{N}(\vec{p}_n|0,\lambda_P^{-1}I_k). \tag{6.6}$$

Similar to $c_{i,j}$, $d_{j,n}$ is a confidence parameter for $T_{j,n}$. λ_T is introduced to control the influence of mutual transition matrix T. As an extreme example, if $\lambda_T = 0$, our algorithm collapses to the MF algorithm.

Hence, through Bayesian inference, we have the posterior distribution of our generative algorithm:

$$\begin{aligned} p(U,V,P|X,T,\lambda_U,\lambda_V,\lambda_P,\lambda_T,C,D) \propto \\ p(X|U,V,C)p(T|V,P,\lambda_T,D)\times \\ p(U|\lambda_U)p(V|\lambda_V)p(P|\lambda_P). \end{aligned} \tag{6.7}$$

The log of posterior distribution for the Eq. 6.7 can be found by substituting the above corresponding probability density functions, i.e., Eq. 6.1, Eq. 6.3, Eq. 6.4, Eq. 6.5 and Eq. 6.6.

6.3.3 Parameter Learning

For learning the parameters, we first have the following corollary:

Corollary 1. *When the prior distribution we are assuming is given, the maximum a posterior (MAP) estimate is equivalent to the complete maximum likelihood (ML) estimate on both observed data and model parameters.*

Since it is an obvious corollary, we put the detailed derivation in Appendix A for reference. On the basis of Corollary 1, maximization of the posterior Eq. 6.7 is equivalent to maximizing the complete log-likelihood of U, V, P, X and T, given the priors λ_U, λ_V, λ_P, λ_T, C and D. As a result, we are going to maximize the following log-likelihood function:

$$\begin{aligned} \ell = &-\frac{\lambda_U}{2}\sum_i \vec{u}_i{}^T\vec{u}_i - \frac{\lambda_V}{2}\sum_j \vec{v}_j{}^T\vec{v}_j \\ &-\frac{\lambda_P}{2}\sum_n \vec{p}_n{}^T\vec{p}_n - \sum_{i,j}\frac{c_{i,j}}{2}(X_{i,j}-\vec{u}_i{}^T\vec{v}_j)^2 \\ &-\frac{\lambda_T}{2}\sum_{j,n} d_{j,n}(T_{j,n}-\vec{v}_j{}^T\vec{p}_n)^2, \end{aligned} \tag{6.8}$$

where we have omitted the constants. We maximize this function with a gradient ascent approach by iteratively optimizing the collaborative filtering variables $\vec{u}_i$, $\vec{v}_j$ and $\vec{p}_n$. Accordingly, we first take the gradient of ℓ with respect to $\vec{u}_i$, $\vec{v}_j$ and $\vec{p}_n$, respectively.

$$\frac{\partial \ell}{\partial \vec{u}_i} = \sum_{i,j}\vec{v}_j c_{i,j}(X_{i,j}-\vec{v}_j{}^T\vec{u}_i) - \lambda_U\sum_i \vec{u}_i{}^T; \tag{6.9}$$

$$\begin{aligned}\frac{\partial \ell}{\partial \vec{v}_j} =& \sum_{i,j} \vec{u}_i c_{i,j}(X_{i,j} - \vec{u}_i{}^T \vec{v}_j) \\ &+ \sum_{j,n} \lambda_T \vec{p}_n d_{j,n}(T_{j,n} - \vec{p}_n{}^T \vec{v}_j) - \lambda_V \sum_j \vec{v}_j{}^T;\end{aligned} \tag{6.10}$$

$$\frac{\partial \ell}{\partial \vec{p}_n} = \sum_{j,n} \lambda_T \vec{v}_j d_{j,n}(T_{j,n} - \vec{v}_j{}^T \vec{p}_n) - \lambda_P \sum_n \vec{p}_n{}^T. \tag{6.11}$$

Then, setting them to zeros leads to the updating equations:

$$\vec{u}_i \leftarrow (VC_iV^T + \lambda_U I_k)^{-1}(VC_iX_i); \tag{6.12}$$

$$\begin{aligned}\vec{v}_j \leftarrow (UC_jU^T + \lambda_T PD_jP^T + \lambda_V I_k)^{-1} \\ (UC_jX_j + \lambda_T PD_jT_j);\end{aligned} \tag{6.13}$$

$$\vec{p}_n \leftarrow (\lambda_T VD_nV^T + \lambda_P I_k)^{-1}(\lambda_T VD_nT_n). \tag{6.14}$$

where C_x and D_x are diagonal matrices with $c_{x,y}$, $d_{x,y}$; $y = 1...Y$ are their diagonal elements. $X_i = (X_{i,j})_{j=1}^J$ for user $\vec{u}_i$ and $T_j = (T_{j,n})_{n=1}^N$ for grid $\vec{v}_j$. Eq. 6.13 shows how mutual transition graph T affects the grid latent vector $\vec{v}_j$, where λ_T balances this effect. In Appendix B, we present a detailed derivation of the updating rules.

6.3.4 Predicting

After the optimal parameters $\vec{u}_i{}^* \in U^*$, $\vec{v}_j{}^* \in V^*$ and P^* are learned, our proposed algorithm can be used for prediction or recommendation tasks. In our case, because the mutual transition graph T is observed, both in-matrix and out-matrix predictions [37] can be estimated by approximating their expectations:

$$\mathbb{E}[X_{i,j}|X^o, T] \approx \mathbb{E}[\vec{u}_i|X^o]^T \mathbb{E}[\vec{v}_j|X^o, T] \tag{6.15}$$

$$X^*_{i,j} \approx (\vec{u}_i{}^*)^T \vec{v}_j{}^* \tag{6.16}$$

For in-matrix prediction, i.e., a grid has been rated by at least one user, the expectation of $\vec{v}_j$ is constrained by both X^o and T. For out-matrix prediction, i.e., none of the users have rated a grid, the expectation of $\vec{v}_j$ is constrained only by T. In our experiments in Section 6.5, in order to verify our Observations 1, 2 and 3, we mainly conduct out-matrix predictions to do the evaluation.

6.4 Knowledge Graph based Probabilistic Matrix Factorization

6.4.1 Problem Definition

Similar to the problem definition in the solution above, KGPMF based solution is defined as follows.

Definition 8. *Given a user-location attention matrix $X \in \mathbb{R}^{M \times N}$ and a knowledge graph G, KGPMF aims to jointly learn the user and location latent spaces (i.e., U and V) whose products can well approximate the original matrix X.*

From the viewpoint of our implementation, the problem can be described as: by observing a part of the user-location attention matrix X^o and location knowledge graph Gs, we expect to well predict the unobserved attention matrix $X^{\neg o}$. Regarded as a knowledge base, the graph G plays a role of embedding auxiliary information into the location latent space.

6.4.2 Modeling

Similar to the first solution shown in Algorithm 5 and related work on collaborative filtering, e.g., [36] [37], as shown in Algorithm 6, by placing Gaussian priors $\mathcal{N}(\vec{\mu}, \sigma^2)$ on latent feature vectors, we solve this prediction problem using a generative process.

Algorithm 6 Generative process for attention prediction.

1: **for** each user $\vec{u}_i \in U$ **do** ▷ Stage 1
2: draw latent user vector $\vec{u}_i \sim \mathcal{N}(0, \sigma_U^2 \mathbf{I})$
3: **end for**
4: **for** each location $\vec{v}_j \in V$ **do** ▷ **Stage 2**
5: draw latent location offset vector $\vec{o}_j \sim \mathcal{N}(0, \sigma_V^2 \mathbf{I})$
6: draw latent location vector $\vec{v}_j = \vec{o}_j + \vec{v}_j^*$
7: **end for**
8: **for** each user-location pair $(\vec{u}_i, \vec{v}_j) \in X$ **do** ▷ Stage 3
9: draw the rating $X_{i,j} \sim \mathcal{N}(\vec{u}_i^T \vec{v}_j, \sigma_X^2)$
10: **end for**

In stage 2, we enrich the latent feature vectors of locations by embedding the knowledge graph. Namely, $V^*, \forall \vec{v}_j^* \in V^*$ could be V_t, $[V_t; V_s]$, $V_{ts}(1)$, or $V_{ts}(2)$ (please see

Section 5.4 for details). In the experiments, by comparison, we discuss in detail why our knowledge graph G can significantly improve such an application.

6.4.3 Parameter Learning

Since there is nothing new about the training processes in matrix factorization methods in Chapter 5, we briefly describe their training strategies used in the experiments.

We minimize the objective functions Eq. 5.1, Eq. 5.3, and Eq. 5.5 with a gradient decent approach by iteratively optimizing the latent variables V_t, E, V_s, M, V_{ts}, D_h, and D_t. For Eq. 5.7, we alternate between optimizing the NN weights (i.e., W_1 and W_2), while fixing the latent features, and optimizing the latent features, while fixing the network weights. Optimization is also carried out by gradient descent on the observed datasets (i.e., time series matrix S and transition matrix T).

For Algorithm 6, since the optimization is very similar to the process that we have described in Section 6.3.3, we omit the detail process here. Someone could also refer to the following related work for further understanding. The optimization could be carried out by maximizing the log-likelihood [37], minimizing a sum-of-squared errors function [36], or using Gibbs sampling. In our experiment, we utilized a previously proposed method [36] for consistency. Namely, all the learning tasks are conducted in a matrix factorization manner.

6.4.4 Predicting

After the optimal parameters $\vec{u}_i^* \in U^*$, $\vec{v}_j^* \in V^*$ are learned, our proposed algorithm can do the prediction as follows:

$$\mathbb{E}[X_{i,j}|X^o, G] \approx \mathbb{E}[\vec{u}_i|X^o]^T \, \mathbb{E}[\vec{v}_j|X^o, G] \tag{6.17}$$

$$X^*_{i,j} \approx (\vec{u}_i^*)^T \vec{v}_j^* \tag{6.18}$$

Similar to the first solution discussed in Section 6.3.4, because the knowledge graph G is observed, both in-matrix and out-matrix predictions [37] can be estimated by approximating their expectations.

6.5 Experimental Evaluation

In the experiment, our goal is to well predict user-attention dynamics in a city. Extensive experiments are conducted on a real-world dataset collected from Beijing to

demonstrate the effectiveness of our proposed algorithms both quantitatively and qualitatively.

Sequentially, we first present the outline of all the datasets collected from the city Beijing. Second, we define an unify evaluation schema, which will be used in performance comparison. Third, the performance of our algorithms, i.e., TMPMF and KGPMF, will be discussed in details. At last, we will also visualize some moving patterns (lifestyles) mined from these two methods.

6.5.1 Data Preparation

Constructing time series matrix S. We used the public Beijing taxi trajectory dataset [101], which contains trajectories of 10,357 taxis, to generate the S. The total number of GPS points p (see Definition 5) is approximately 15 million. Figure 5.1 shows some examples of S. Please see Section 5.4.1 for the detailed process of constructing S.

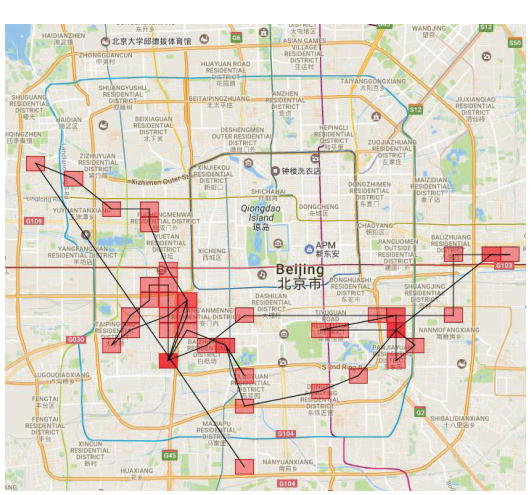
Constructing transition matrix T. The grids' mutual transition graph T is constructed using two kinds of transition data, i.e., by bus and taxi. Figures 6.3c and 6.3d show the heat maps of the 2,500 grids' out-degrees in the transition matrices constructed using bus and taxi trajectory datasets, respectively. Because the bus transition is route-fixed, there are few grids that have high out-degrees (a maximum of 200 shown in Figure 6.3c). By contrast, as shown in Figure 6.3d, most of the grids have very high out-degrees using the taxi transition dataset. For each entry $T_{j,n}$, it records the number of transitions from grid j to grid n. After normalization, $T_{j,n}$ can be regarded as the probability of the transition from grid j to grid n. In the next experiments, we will compare the performance of using the bus and taxi transition information.

Figure 6.3a shows a taxi's $Tr^{'}$ (see Section 5.4.2) in one day on Google Maps. All the $Tr^{'}$ are aggregated to obtain $T \in \mathbb{R}^{2500\times2500}$. Although the other edge weights are close to zero, 35,259 main edges are detected in T. The same process is done on the bus trajectory dataset. Accordingly, we use T_{taxi} and T_{bus} to denote the taxi and bus transtion matrix, respectively.

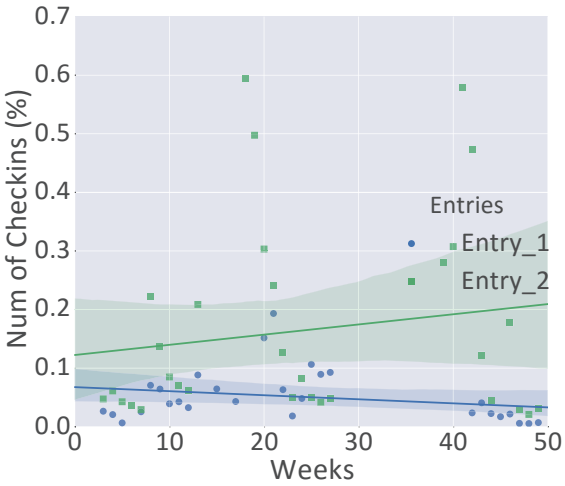
Constructing user-location attention matrix X. The user-grid attention matrix X is constructed using the geo-tagged check-ins obtained from Sina Weibo APIs *. Figure 6.1a shows the {longitude, latitude} distribution of the data. Totally, we obtain a raw dataset of 527,809 users and 1,729,998 check-ins.

Then, similar to the CCCF model proposed in [102], we first conduct a prelimi-

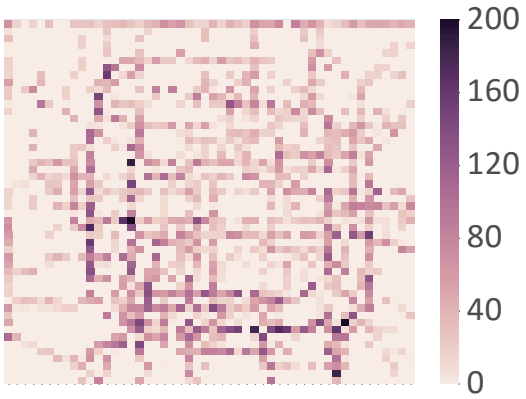
*http://open.weibo.com/wiki/API



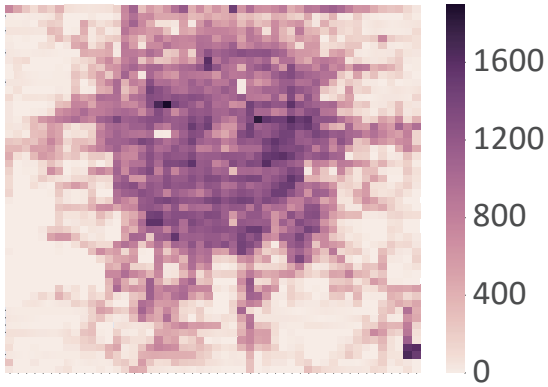
(a) Taxi route in one day



(b) Two observed user-grid entries



(c) Bus transition dataset



(d) Taxi transition dataset

Figure 6.3. Dataset preparation. (c) and (d) present the heat maps of the 50×50 grids' out-degrees in transition matrix T.

nary task of user-region co-clustering to discover like-minded user groups, each of whom has similar interests (i.e., visiting similar regions in a city). In addition to the advantages specified in [102], the simulation of collective attention to grids is more meaningful than that of individual attention to grids. For example, by analyzing the tourist-like user group's attention (i.e., the red group in Figure 6.1b), we can discover the sightseeing spots in Beijing.

After clustering, we calculate each observed entry $X_{i,j}$ based on the percentage of numbers of check-ins that were uploaded by user group u_i in grid v_j. Figure 6.3b shows the percentages of two $X_{i,j}$s over all the weeks in 2015. Unsurprisingly, because

of anomalous events [103] [104] [105], the percentages fluctuate over time. The value of each observed $X_{i,j}$ is the average percentage of all the recorded weeks. Finally, we obtain the X of 100 classified user groups and 2,500 grids with 43,069 observed user-grid entries. This matrix has a sparsity of 82.8%.

6.5.2 Evaluation Schema

Similar to [106] [107], we randomly divide all the grids associated with each user group into 2 chunks. One round of the validation performs training on one chunk (i.e., the training set) and validates the trained model on the other chunk (i.e., the testing set). To evaluate and compare the algorithms under both sparse and dense settings, $x = 10\%, 30\%$ of the grids are selected as the training set, respectively. For each value of x, we repeat the validation five times with different randomly selected training sets. Finally, the average performance is reported.

Similar to [37] [38], we use recall as the performance measure because the rating information in X is in the form of implicit feedback [100] [98]. As discussed above, a zero entry may be due to the fact that the user group does not like the grid or does not know about it. This makes it difficult to accurately compute precision. Because recall only considers the nonzero entries, for each user group we define the $recall@M$ as:

$$recall@M = \frac{TPs@M}{TPs@M + FNs@M}, \tag{6.19}$$

where $TPs@M$ (True Positives) is the number of grids the user group pays attention to in top M, and $TPs@M + FNs@M$ (False Negatives) is the total number of grids the user group pays attention to. The reported recall for the entire system is the average recall from all user groups.

6.5.3 Our Methods and Baselines

To make our experiments more reproducible, we list all methods included in our comparison as well as the values of their hyperparameters. The parameter-tuning process will also be discussed later in Section 6.5.4. Our methods and all the baseline are listed as follows.

- **TMPMF:** Transition Matrix based Probabilistic Matrix Factorization is our first proposed algorithm in Section 6.3. By introducing the latent space P, it can deal with different kinds of mutual dependencies among locations. We apply our

algorithm by setting the hyperparameters $\lambda_U = 0.1$, $\lambda_V = 1$, $\lambda_P = 10$, $\lambda_T = 10$ and $\alpha = 1$ (see Eq. 6.2). We set the number of latent moving patterns (user demands) $k = 15$. To support our choice, we visualize some learned moving patterns in Section 6.5.6.

- **KGPMF-*none*, KGPMF-V_t, KGPMF-V_t&V_s, KGPMF-$V_{ts}(1)$, KGPMF-$V_{ts}(2)$.** These five variants of the Knowledge Grpah based Probabilistic Matrix Factorization are obtained when *none*, V_t, $[V_t; V_s]$, and V_{ts}s are assigned to V^* in Algorithm 6, respectively. When learning the location relation knowledge graph G, the values of the hyperparameters are: $k_t = 20$; $k_s = 20$; $k_{ts} = 10$; $k_r = 5$; $n = 20$; $\lambda_r = 0.01$; $\lambda_s = 1$; $\theta = 0.8$.

- **GeoMF:** GeoMF [2] is the state-of-the-art method for Point-of-Interests (POIs) recommendation. By regarding the "grids" in our application as "POIs", our methods and GeoMF are solving a similar problem. Compared with the conventional MF (see Eq. 2.1), GeoMF (see Eq. 2.6) introduces two additional matrices, i.e., a user-activity-areas matrix X and a POI-influence-areas matrix Y, to incorporate the geographical information. Hence, GeoMF also takes the location-location spatial information into consideration. By comparison, a detailed discussion will be presented later in this section.

- **CMF:** Although the Collective Matrix Factorization is a high-level framework, to verify our method's advantages, we implemented an instance of this model using squared euclidean distance as the Bregman divergence (see Eq. 2.4). The reason of using squared euclidean distance is that, as has been mentioned in [41], the link function $f(x) = ax + b$ (i.e., the derivative of squared euclidean distance) is corresponding to Gaussian noise, which is the same as the assumption in our model.

- **GNMF:** Graph Regularized Nonnegative Matrix Factorization is the algorithm proposed in [42]. Our TMPMF to a considerable extent is an extension of this one. Thus, we implement their algorithm. By comparison, we expect to verify that our algorithm can deal with different mutual dependencies that GNMF cannot be applied to. Through parameter tuning , we set $\lambda = 0.1$ in Eq. 2.5.

- **MF:** As we discussed above, if $\lambda_T = 0$, the TMPMF algorithm collapses to the conventional Matrix Factorization. By keeping the values of other parameters in TMPMF the same, we use MF to denote this baseline method.

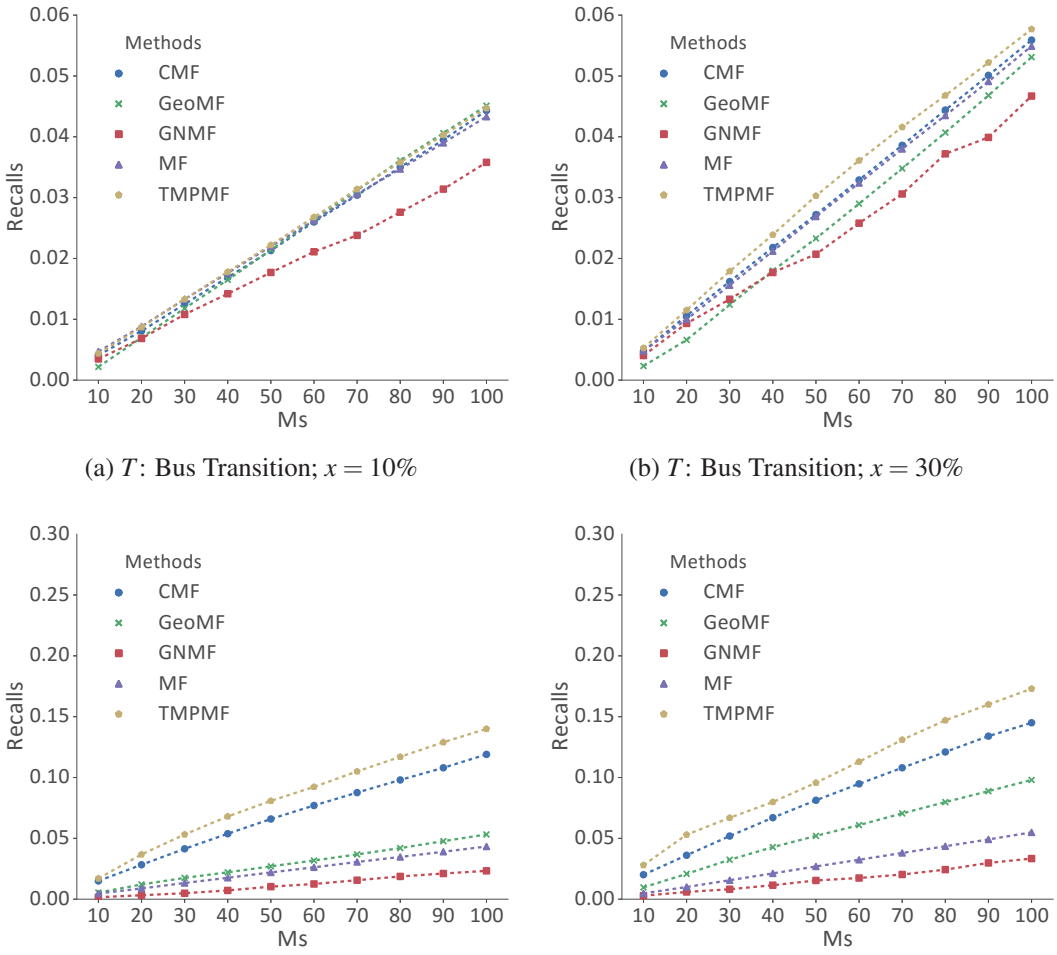
6.5.4 Evaluation on TMPMF

The experiment results are reported in Figure 6.4. About the transition matrix T, Figures 6.4a and 6.4b show the results when the bus transition information is used, while Figures 6.4c and 6.4d show the results when the taxi transition information is used. Overall, the results show that the proposed algorithm, TMPMF, outperforms the baselines significantly. The details are as follows.

Our algorithm, TMPMF, obtains encouraging results in solving real-world problems. By comparing the experimental results shown in Figures 6.4c and 6.4d, we can see that the proposed TMPMF significantly outperformed the CMF by an average margin of 1.5% in the sparse setting (i.e., $x = 10\%$ in Figure 6.4c) and 1% in the dense setting (i.e., $x = 30\%$ in Figure 6.4d). On the basis of Eq. 6.19, when M is small (e.g., recall@$M = 10, 15, ...100$), a margin of 1% is a significant improvement. Furthermore, a larger margin in the sparse setting means that our algorithm can well learn the user demands from the taxi transition graph T, by which the user-attention dynamics are well predicted.

The bus transition information is inadequate for learning user demands. As shown in Figures 6.4a and 6.4b, the bus transition information has not helped our algorithm more efficiently predict users' attention to grids. One reason is that the bus transition graph T_{bus} is too sparse. As shown in Figure 6.3c, because the bus stops are fixed, there are very few grids involved in constructing the T_{bus}. What's worse, since the bus transition is route-fixed, its moving patterns cannot be regarded as user demands. That is, the latent moving patterns are learned from the pre-designed bus system rather than users' preferences. In this sense, our algorithm to a considerable extent is robust enough to tolerate bad inputs that would mislead the final results. As a counter example, it has further verified our observation that the complementary relationships among grids can be learned from users' mobility but not from buses' fixed movements.

CMF is not so practical. As has been mentioned in [41] [40], when applying CMF, we should inject our domain knowledge about what sort of noise there is in observed X and T. Also, we should identify what parameter matrices U, V and P are a priori most likely. What's worse, when solving a particular problem, someone has to control many of the model and optimization parameters across the experiments, which makes CMF not so practical. As a result, although the same assumption (i.e., X and T are consistent with the normal distribution) is used in both our TMPMF and CMF in the experiment, TMPMF obtained better performance.



(a) T: Bus Transition; $x = 10\%$

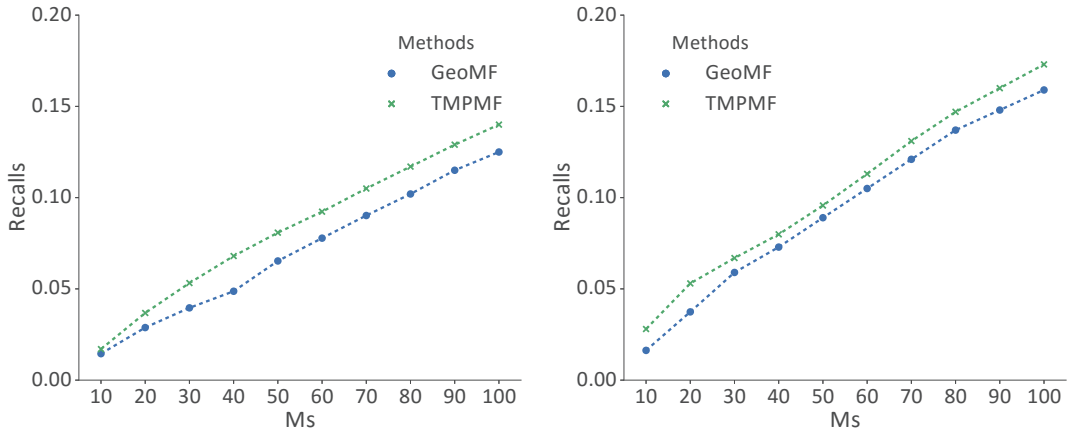
(b) T: Bus Transition; $x = 30\%$

(c) T: Taxi Transition; $x = 10\%$

(d) T: Taxi Transition; $x = 30\%$

Figure 6.4. Performance comparison of TMPMF, GeoMF, CMF, GNMF and MF based on recall@M ($M = 10, 15, ..., 100$) (see Eq. 6.19). For the mutual dependency graph T, bus and taxi transition information is used, respectively. For the sparse-dense setting, the performance is calculated and presented when $x = 10\%, 30\%$. Please note that the value ranges of the y-axis are different.

GeoMF provides a solid baseline for the user-attention prediction task, but cannot mine the user mobility. By seeing Figures 6.4c and 6.4d, GeoMF's low performance indicates that, when we assigned taxi transition T_{taxi} to Y in Eq. 2.6, GeoMF cannot learn the moving patterns well for the user-attention prediction task. However, if we assigned the POI-influence-areas matrix (see the Definition 2 in [2] for details) to Y in Eq. 2.6, GeoMF obtained good performance shown in Figure 6.5. The differences



(a) $x = 10\%$ (b) $x = 30\%$

Figure 6.5. Performance comparison of TMPMF and GeoMF based on recall@M ($M = 10, 15, ..., 100$) (see Eq. 6.19). In this figure, we present TMPMF's performance when the taxi transition T_{taxi} is used. For GeoMF, we use the exactly same POI-influence-areas matrix Y in [2] to obtain the best performance.

between the transition matrix T and the POI-influence-areas matrix Y are that: $T_{j,n}$ defines the interaction from grid $\vec{v}_j$ to grid $\vec{v}_n$ as the probability of users' movements from $\vec{v}_j$ to $\vec{v}_n$ (regarded as the social distance); $Y_{j,n}$ is fixed in advance, which is calculated based on the physical distance between $\vec{v}_j$ and $\vec{v}_n$. In this sense, by comparing the performance in Figure 6.5, we can conclude that the social distance based user-attention prediction outperforms the physical distance based prediction. What's worse, by comparing the performance in Figure 6.4, GeoMF cannot deal with the social distance based prediction.

GNMF is not flexible enough to deal with different kinds of T. Although both our algorithm (Eq. 6.7) and GNMF (Eq. 2.5) have a similar way of constraining the grid latent space V using a graph, we obtain totally different results when solving our problem. Because GNMF cannot deal with the dependencies in our problem, it has been misled by the "bad" input T. As a result, it obtains the lowest recalls in comparison with the other methods. This result strongly suggests that our algorithm is much more robust than GNMF. On the other hand, since our algorithm can also deal with the "similarity" dependency (i.e., the definition of T is the same as W in Eq. 2.5), it is an extension of GNMF. That is, by inputting a proper graph T, GNMF can be regarded as an instance of the proposed TMPMF.

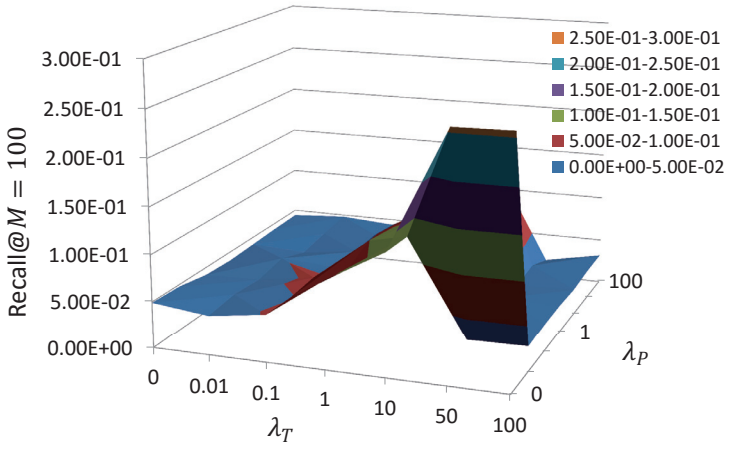
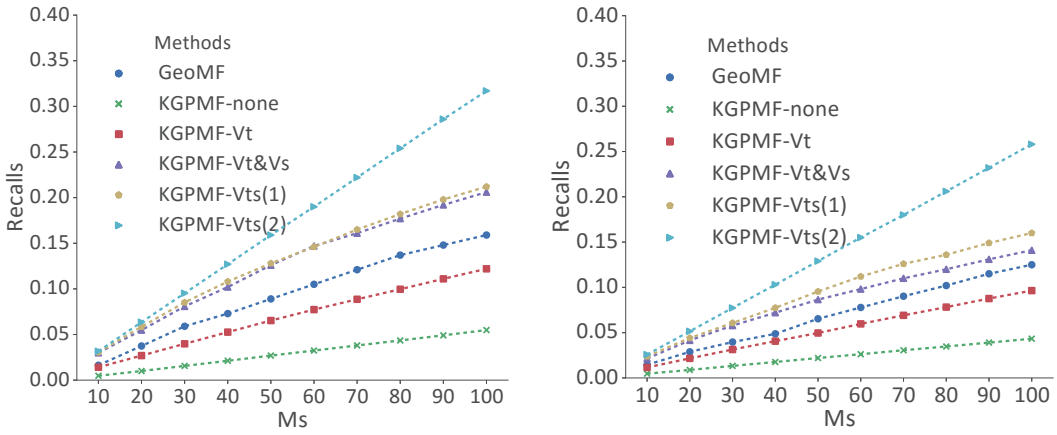


Figure 6.6. The plot of prediction recall@$M = 100$ for the proposed TMPMF algorithm by varying moving pattern parameter λ_P and dependency parameter λ_T.

Impact of Parameters λ_T

To gain better insight into our algorithm, it is worthwhile to explore how the parameters, especially λ_T, can balance or influence our algorithm. By utilizing the grid search method proposed in [54], Figure 6.6 shows a parameter-tuning process for obtaining the best prediction performance. λ_T (the x-axis) is the precision parameter that controls the impact of the transition graph T, and λ_P (the y-axis) is the prior on the moving pattern distributions P.

From this figure, we observe that the value of λ_T impacts the prediction results significantly, which demonstrates that fusing items' mutual dependency information can increase prediction recall considerably (about 10% in our case). When the other parameters are kept fixed, the contour plot also indicates that for small values of λ_T, the improvement in prediction recall is small, and it increases with further increase in λ_T. However, when λ_T increases beyond a certain threshold, prediction recall decreases. Intuitively, for large values of λ_T, our algorithm gives more preference to items' dependency information but less preference to collaborative information between users and items. In our application, TMPMF achieves best performance when $\lambda_T \approx 10$ and $\lambda_P \approx 10$.



(a) Dense setting, i.e., $x = 30\%$

(b) Sparse setting, i.e., $x = 10\%$

Figure 6.7. Performance comparison of our KGPMF based methods and baselines based on recall@M ($M = 10, 20, ..., 100$).

6.5.5 Evaluation on KGPMF

On the basis of the experimental results shown in Section 6.5.4, our TMPMF algorithm obtained the highest performance when the taxi transition information (i.e., T_{taxi}) is used. Therefore, in this section, we will only compare the different variants of KGPMF with TMPMF in the context of using taxi transition matrix T_{taxi}.

The experimental results are reported in Figure 6.7. Figure 6.7a shows the results under a dense setting (i.e., $x = 30\%$), whereas Figure 6.7b shows the results under a sparse setting (i.e., $x = 10\%$). Overall, the results show that our knowledge graph (i.e., KGPMF) greatly improves the prediction performance. The details are as follows.

User attention prediction is a challenging problem. The curve of KGPMF-*none* indicates, without introducing any auxiliary information, that this method performs poorly, especially under a sparse setting.

The spatial information V_s is essential for the prediction. A comparison of the curves of KGPMF-V_t and KGPMF-V_t&V_s shows that KGPMF-V_t&V_s significantly outperformed KGPMF-V_t by an average margin of 3.4% in the sparse setting and 5.8% in the dense setting. On the basis of Eq. 6.19, when M is small (e.g., $M = 10, ..., 100$), a margin of 1% is a significant improvement. In conventional collaborative filtering-based recommendation (e.g., for movies), if a user prefers sci-fi movies, given a new sci-fi movie, the model would predict a high score for this user. However, this assumption is not applicable in our case. That is, a user living in residential area A (high

attention) rarely pays attention to another faraway residential area B. Hence, in addition to the content features V_t of the locations, the mutual dependencies among them (i.e., V_s) are essential for this type of prediction.

The TMPMF algorithm is insufficient. Since the transition matrix is directly used in TMPMF, as a result, important spatial information is lost. For example, a semantic *work* $\leftrightarrow$ *home* relation [4] cannot be embedded when learning. By comparing TMPMF with the other methods shown in Figure 6.7, it can be further improved.

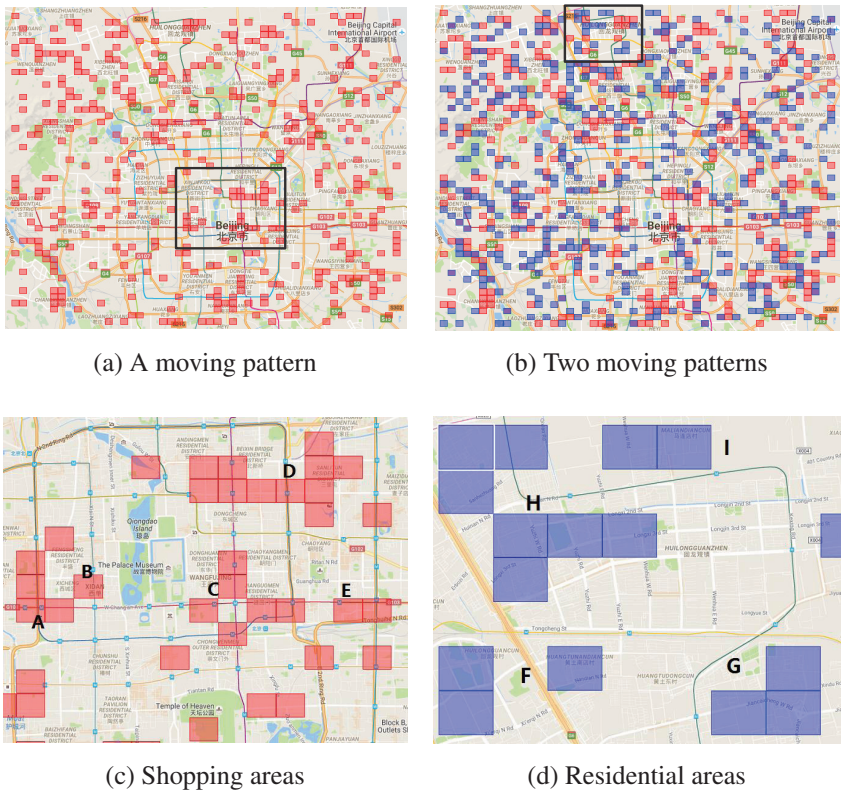
KGPMF-$V_{ts}(1)$ and KGPMF-$V_{ts}(2)$ obtained the highest performance. Compared with KGPMF-V_t&V_s that separately takes temporal and spatial information into consideration, KGPMF-$V_{ts}(1)$ and KGPMF-$V_{ts}(2)$ jointly consider both of them. Several latent semantic relations are detected among the locations. We can regard these relations as lifestyles in a city. For example, office workers may pay more attention to the locations in the *work* $\leftrightarrow$ *home* relations. Students may pay more attention to the locations in the *school* $\leftrightarrow$ *shopping* relations. Consequently, when a small training set is used (i.e., the sparse setting), KGPMF-$V_{ts}(1)$ significantly outperforms KGPMF-V_t&V_s, suggesting that KGPMF-$V_{ts}(1)$ and KGPMF-$V_{ts}(2)$ provide two ways to leverage GPS trajectories into a latent semantic space.

6.5.6 Visualization

Visualizing moving patterns learned by TMPMF

Since one of our goals is to mine users' moving patterns from transition information, in this section, we visualize several moving patterns (i.e., P in Figure 6.2) learned from the taxi transition information to show the effect of the proposed TMPMF algorithm. As shown in Figure 6.3d, when the taxi transition information is used to construct the graph T, the 50×50=2,500 grids are almost fully connected with different weights. As a result, it is a big challenge to mine the meaningful moving patterns. By visualization, we try to present and discuss the moving patterns we have obtained.

By utilizing the Google Map's APIs, Figure 6.8 shows two moving patterns learned by our algorithm. Labeled by red, Figure 6.8a shows the grids involved in one of them. The representative grids are the destinations of the target moving pattern, which can be regarded as the user demands. In Figure 6.8b, both of them are merged on the map. Because it is non-intuitive to explain them as a whole, by zooming in on the map, we discuss some particular examples (i.e., the grids in the two black boxes) shown in Figures 6.8c and 6.8d



(a) A moving pattern

(b) Two moving patterns

(c) Shopping areas

(d) Residential areas

Figure 6.8. Moving pattern visualization. Labeled by red and blue, in (a) and (b), two moving patterns are presented on Google map. By zooming in on the map, (c) and (d) show some representative grids involved in these two moving patterns.

As seen in Figure 6.8c, the red moving pattern is more related with the shopping demands. Labeled by "A", "B", "C", "D" and "E", the shopping areas are discovered in this pattern. More specifically, "A" is a financial street having some shopping malls; "B", "C", "D" and "E" are all popular shopping areas, which are called Xidan, Dongdan, Sanlitun and SOHO, respectively. The moving pattern shown in Figure 6.8d is more related with the residential demands. The areas labeled by "F", "G", "H" and "I" are all residential regions around Huilongguan town. In this sense, we can say that the representative grids in each moving pattern accurately reveal the underlying meaning of the corresponding user demands.

A number of related studies have been conducted for modeling the individual mobility behavior, such as HMM [21], CRF [11], and periodic GMM [4]. Recently, several studies [22] [10] [108] have also focused on predicting the places where users will go next, based on their previously generated content. Nevertheless, to the best

of our knowledge, estimating users' attention to locations (i.e., the collective mobility behavior) by jointly analyzing both online and offline generated social media has not yet been thoroughly investigated. Potential application scenarios include Points-of-Interests recommendation, user profiling, personal digital assistant (PDA), urban planning, and so on.

On the other hand, our application provides another way to learn the functional compositions of urban spaces. A number of algorithms have been proposed recently. For instance, [17] have proposed a supervised multi-output regression method to estimate the functional composition of urban spaces. In [18], the authors proposed a topic-based inference model, which regards a region as a document, a function as a topic, to discover different functionalities. In these works, the semantic information (e.g., the categories of POIs) is utilized. Unlike the related works, our application discovers a grid's functionalities by learning user demands. Without using any semantic information, our application provides a complementary way for solving such a problem.

Visualizing moving patterns learned by KGPMF

Since the final location relation knowledge graph $G(V_{ts}(2), R_{ts}(2))$ (i.e., KGPMF-$V_{ts}(2)$) obtained the best performance in our experiments described in Section 6.5.5, we visualize some relations in $G(V_{ts}(2), R_{ts}(2))$. The Google Maps APIs are utilized on $G(V_{ts}(2), R_{ts}(2))$ to learn some relation instances, as shown in Figure 6.9.

Figure 6.9a shows a shopping like relation between red grids (head) and blue grids (tail). By zooming in on the map, we find that almost all the blue grids are shopping areas. More specifically, Figure 6.9b shows that there is a shopping mall in A, a plaza in B, a famous shopping street in C and the Times Square in D. Although E is not as well known, we find it has many bank ATMs suggesting it is also a shopping area.

Figure 6.9c and 6.9d present two other interesting relation instances. In Figure 6.9c, our method detected an airport expressway, because of the high similarity among head grids. Figure 6.9d shows an instance our method learned in the Zhongguancun district. Although the grids in Zhongguancun very likely play multi-roles (e.g., for working, education, shopping, and residency), a residency-to-education like relation is detected. For example, H contains a university and some research institutes. However, we find that several large universities are missing in this relation. Labeled by F, Tsinghua University is not included. One possible explanation is that, since most of the students and teachers reside on campus (e.g., G, which has many apartments and dormitories), they walk to their learning places. The taxi trajectories cannot characterize such cases.

(a) Top 50 tuples in a relation

(b) Zoom-in on the map of (a)

(c) An airport expressway

(d) A relation in Zhongguancun

Figure 6.9. Relation visualization. Red: heads; blue: tails.

Although only GPS points are used in our framework, the case studies show that our methods proposed in Chapter 5 performed well in terms of learning semantic relations among locations.

In this chapter, we propose two algorithms, TMPMF and KGPMF, that exploit the mutual dependencies among locations. In terms of learning, we also derive an optimization scheme for the Bayesian treatment of TMPMF. In our experiments, both TMPMF and KGPMF sufficiently mine collective moving patterns in a city, by which an encouraging result is obtained for predicting users' attention to locations.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

How to make full use of the large heterogeneous UGC is a big challenge in the Location-Based Social Networks (LBSNs) research area. The work presented in this thesis provided a solution for it.

In this work, we proposed a very broad framework of mining various knowledge from LBSNs. To this end, a new framework of jointly considering user profiling knowledge and location profiling knowledge is proposed (please see Figure 1.5 for reference). In this framework, by collecting User Generated Content (UGC) from LBSNs, we first devised some methodologies to mine the useful knowledge of users and locations, respectively. Then, by mapping the different knowledge into a unified latent space with a same metric, we tried to analyze the interaction between users and locations. For instance, two new generative algorithms were devised to embed the knowledge of both users and locations when predicting the extents of user attention to locations in a city.

To verify the utility of our technical contributions, we also applied our framework to solve different social problems. For instance, our framework could be used to discover potential sightseeing locations, learn user mobility patterns, predict user attention to locations, and so on.

To draw a conclusion, a comprehensive structure, (1) user profiling; (2) location

profiling; (3) location relation profiling; and (4) user-location profiling, was defined in this framework. Although our current research achievements cannot take into account all aspects of this big problem, we have attacked the following particular sub-problems.

User Profiling. We proposed a new concept, *location authority*, for mining information asymmetries among different online user groups. By respectively considering a user's social, temporal, and spatial information, we devised three models for estimating users' location authority: a social-network-driven model, time-driven model, and location-driven model. We then introduced two ensembles of these models, which demonstrated robust performance across different classification tasks.

Location Profiling. We are the first who proposed to use two dimensions, i.e., user group distribution based popularity and scenery quality, to describe a location. To quantify these two dimensions, by analyzing different available information, several methods were proposed from various points of view.

Location Relation Profiling. Our work is among the first of its kind to construct a location relation knowledge graph that embeds temporal and spatial information. Two new unsupervised relation extraction methods were devised to learn the relations among nodes (locations) on this graph.

User-to-Location Profiling. In addition to the relation analysis among locations, we also did a work on analyzing the relations between users and locations. By utilizing the knowledge discovered in the work above, two new algorithms were proposed to predict the user attention to locations. The essential patterns learned can be used in a wide range of problem domains, such as city planning, geo-social recommendation, traffic forecasting, and so on.

The utilities of all the proposed novel concepts and models have been verified in real scenes. For instance, the user and location profiling knowledge, i.e., quantifying the two dimensions, can be applied to detect potential obscure sightseeing locations. The location relation knowledge graph can mine both individual and collective user mobility, the essential knowledge it contains can be used in a wide range of problem domains, like city planning, geo-social recommendation, advertisement targeting, traffic forecasting, and so on. Therefore, we want to emphasize that our proposed research framework did construct a benign cycle among: (1) devising fundamental technologies; (2) developing applications and systems; and (3) promoting more offline activities. By analyzing the users' content generated from offline activities, our current work mainly focus on the stage of devising fundamental technologies.

7.2 Future Work

The proposed research framework in this dissertation is intended to inspire more interests in social informatics research. As another goal, we are aiming to apply the advances in computer science into the real world. Accordingly, we have the following future plan.

Application Standpoint. We are interested in how to shorten the gap between technologies and our daily life. My current work mainly focuses on urban computing that aims to solve particular problems occurred around us. For example, as mentioned in this dissertation, we have already developed a demo to recommend various sightseeing locations to tourists. In the future, we hope to improve the current demo. The multi-faceted nature of the problem of selecting sightseeing locations would be considered. Not only the scenery itself, but also a location's facilities, traffic, or culture values could be further considered. In addition, by considering continuous geo-tagged UGC, real-time travel route planning is a potential work.

Theoretical Standpoint. We also have the ambition to make breakthroughs in the theoretical research area. Since we have already constructed a knowledge graph that can be used in different problem domains, one of the key problems is about how to use such a huge high-dimensional graph as input for different models or algorithms. Therefore, in the near future, we will mainly focus on this topic. For example, similar to the Convolution Neural Networks devised for image processing, one problem of significant interest for me is to devise an Anisotropy Convolution Neural Network for knowledge graph.

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Professor Masatoshi Yoshikawa and Associate Professor Qiang Ma, for their supports in overcoming numerous obstacles I have been facing through my research. In addition, I would like to thank my other five academic advisors, Professor Katsumi Tanaka, Doctor Xing Xie, Doctor Ruihua Song, Associate Professor Xuefeng Liang and Professor Toru Ishida, for their valuable suggestions and comments on my research. Thanks to Professor Michihiko Minoh for being a member of the thesis committee. Expertise on different research areas, they broadened my research horizons.

I would like to thank all our Lab faculties for their help in every possible way. During the Lab seminars, Associate Professor Yasuhito Asano, Associate Professor Adam Jatowt, Assistant Professor Toshiyuki Shimizu, Assistant Professor Makakoto P. Kato and Assistant Professor Takehiro Yamamoto made valuable comments on my research, which inspired me a lot of new ideas. Also, the Lab secretaries, Ms. Yukiko Nakajima, Ms. Yoko Nakahara and Ms. Rika Ikebe helped me much on daily business. For example, all the necessary processes of attending a conference were prepared by them.

I would like to thank all the Lab mates for sharing souvenirs from all over the world. Thanks to Ms. Yizhu Shen, Mr. Min Ge and Mr. Junjie Sun in our sightseeing group. Thanks to my fellow mates, Doctor Xinpeng Zhang, Doctor Jiyi Li, Doctor Yang Cao, Ms. Yiling Dai, Ms. Purnomo Husnul Khotimah, Ms. Sora Lim and Mr. Hideaki Ohashi. We helped, cooperated, and encouraged with each other.

Last but not the least, I would like to thank my parents, for supporting me throughout the doctor program. Specially, I would like to thank my better half, who is studying in the same Department of Social Informatics. We make progress together.

Chenyi ZHUANG, June 2017

REFERENCES

[1] https://www.flickr.com/photos/walkingsf/sets/72157624209158632/.

[2] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *ACM SIGKDD*, pages 831–840, 2014.

[3] Jiajun Liu, Zi Huang, Lei Chen, Heng Tao Shen, and Zhixian Yan. Discovering areas of interest with geo-tagged images and check-ins. In *ACM Multimedia*, pages 589–598, 2012.

[4] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *SIGKDD*, pages 1082–1090, 2011.

[5] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006.

[6] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2008.

[7] Ray M Chang, Robert J Kauffman, and YoungOk Kwon. Understanding the paradigm shift to computational social science in the presence of big data. *Decision Support Systems*, 63:67–80, 2014.

[8] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 5(3):38:1–38:55, 2014.

[9] Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua. Research and applications on georeferenced multimedia: a survey. *Multimedia Tools Appl.*, 51(1):77–98, 2011.

[10] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *ACM SIGKDD*, pages 1275–1284, 2015.

[11] Nicholas Jing Yuan, Yingzi Wang, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. Reconstructing individual mobility from smart card transactions: A space alignment approach. In *ICDM*, pages 877–886, 2013.

[12] Claudio Cioffi-Revilla. Computational social science. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):259–271, 2010.

[13] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. Graph theory. *Springer, New York*, 2008.

[14] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[15] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW*, pages 61–70, 2010.

[16] Dan Xu, Peng Cui, Wenwu Zhu, and Shiqiang Yang. Find you from your friends: Graph-based residence location prediction for users in social media. In *ICME*, pages 1–6, 2014.

[17] Zack Zhu, Jing Yang, Chen Zhong, Julia Seiter, and Gerhard Tröster. Learning functional compositions of urban spaces with crowd-augmented travel survey data. In *ACM SIGSPATIAL*, pages 22:1–22:10, 2015.

[18] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *ACM SIGKDD*, pages 186–194, 2012.

[19] Jiebo Luo, Dhiraj Joshi, Jie Yu, and Andrew Gallagher. Geotagging in multimedia and computer vision - a survey. *Multimedia Tools Appl.*, 51(1):187–211, 2011.

[20] Rongrong Ji, Xing Xie, Hongxun Yao, and Wei-Ying Ma. Mining city landmarks from blogs by graph modeling. In *ACM Multimedia*, pages 105–114, 2009.

[21] Yong-Joong Kim and Sung-Bae Cho. A hmm-based location prediction framework with location recognizer combining k-nearest neighbor and multiple decision trees. In *HAIS*, pages 618–628, 2013.

[22] Huiji Gao, Jiliang Tang, and Huan Liu. Exploring social-historical ties on location-based social networks. In *ICWSM*, pages 114–121, 2012.

[23] Susan Hanson and Perry Hanson. Gender and urban activity patterns in uppsala, sweden. *Geographical Review*, pages 291–299, 1980.

[24] Shan Jiang, Joseph Ferreira Jr, and Marta C Gonzalez. Discovering urban spatial-temporal structure from human activity patterns. In *ACM SIGKDD workshop on urban computing*, pages 95–102, 2012.

[25] Jameson L Toole, Michael Ulm, Marta C González, and Dietmar Bauer. Inferring land use from mobile phone activity. In *ACM SIGKDD workshop on urban computing*, pages 1–8, 2012.

[26] Srinivas Devarakonda, Parveen Sevusu, Hongzhang Liu, Ruilin Liu, Liviu Iftode, and Badri Nath. Real-time air quality monitoring through mobile sensing in metropolitan areas. In *ACM SIGKDD workshop on urban computing*, page 15, 2013.

[27] Yu Zheng, Xuxu Chen, Qiwei Jin, Yubiao Chen, Xiangyun Qu, Xin Liu, Eric Chang, Wei-Ying Ma, Yong Rui, and Weiwei Sun. A cloud-based knowledge discovery system for monitoring fine-grained air quality. *MSR-TR-2014–40, Tech. Rep.*, 2014.

[28] Silvia Santini, Benedikt Ostermaier, and Andrea Vitaletti. First experiences using wireless sensor networks for noise pollution monitoring. In *Workshop on Real-world Wireless Sensor Networks*, pages 61–65, 2008.

[29] Rajib Rana, Chun Tung Chou, Nirupama Bulusu, Salil Kanhere, and Wen Hu. Ear-phone: A context-aware noise mapping using smart phones. *Pervasive and Mobile Computing*, 17:1–22, 2015.

[30] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. Diagnosing new york city's noises with ubiquitous data. In *UbiComp*, pages 715–725, 2014.

[31] Chih-Chieh Hung, Chih-Wen Chang, and Wen-Chih Peng. Mining trajectory profiles for discovering user communities. In *Workshop on Location Based Social Networks*, pages 1–8, 2009.

[32] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.

[33] Jia-Dong Zhang and Chi-Yin Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *SIGSPATIAL*, pages 334–343, 2013.

[34] Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. pages 58–65, 2012.

[35] Shoko Wakamiya, Ryong Lee, and Kazutoshi Sumiya. Crowd-sourced urban life monitoring: urban area characterization based crowd behavioral patterns from twitter. In *ICUIMC*, pages 26–34, 2012.

[36] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, volume 1, pages 2–1, 2007.

[37] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *ACM SIGKDD*, pages 448–456, 2011.

[38] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *ACM SIGKDD*, pages 1235–1244, 2015.

[39] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful represen-

tations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[40] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *ACM SIGKDD*, pages 650–658, 2008.

[41] Geoffrey J Gordon. Generalizedˆ 2 linearˆ 2 models. In *NIPS*, pages 577–584, 2002.

[42] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[43] Tianran Hu, Ruihua Song, Yingzi Wang, Xing Xie, and Jiebo Luo. Mining shopping patterns for divergent urban regions by incorporating mobility data. In *CIKM*, pages 569–578, 2016.

[44] Wei-Chao Chen, Agathe Battestini, Natasha Gelfand, and Vidya Setlur. Visual summaries of popular landmarks from community photo collections. In *ACM Multimedia*, pages 789–792, 2009.

[45] Qiang Hao, Rui Cai, Changhu Wang, Rong Xiao, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. Equip tourists with knowledge mined from travelogues. In *WWW*, pages 401–410, 2010.

[46] Hau-Wen Chang, Dongwon Lee, Mohammed Eltaher, and Jeongkyu Lee. @ phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *ASONAM*, pages 111–118, 2012.

[47] Michael D Lieberman and Jimmy Lin. You are where you edit: Locating wikipedia contributors through edit histories. In *ICWSM*, pages 106–113, 2009.

[48] Adrian Popescu and Gregory Grefenstette. Mining user home location and gender from flickr tags. In *ICWSM*, pages 307–310, 2010.

[49] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. A latent variable model for geographic lexical variation. In *EMNLP*, pages 1277–1287, 2010.

[50] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *SMUC*, pages 37–44, 2010.

[51] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: user classification in twitter. In *ACM SIGKDD*, pages 430–438, 2011.

[52] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *VLDB*, pages 576–587, 2004.

[53] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[54] Chih wei Hsu, Chih chung Chang, and Chih jen Lin. A practical guide to support vector classification, 2010.

[55] Lior Rokach. *Pattern Classification Using Ensemble Methods*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2010.

[56] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996.

[57] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

[58] Daniel Jiménez. Dynamically weighted ensemble neural networks for classification. In *IJCNN*, pages 753–756, 1998.

[59] Luca Pretto. A theoretical analysis of googlefs pagerank. In *SPIRE*, pages 131–144, 2002.

[60] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[61] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR*, pages 737–744, 2011.

[62] Qiang Hao, Rui Cai, Jiang-Ming Yang, Rong Xiao, Like Liu, Shuo Wang, and Lei Zhang. Travelscope: standing on the shoulders of dedicated travelers. In *ACM Multimedia*, pages 1021–1022, 2009.

[63] Yunpeng Li, David J Crandall, and Daniel P Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, pages 1957–1964, 2009.

[64] Heng Liu, Tao Mei, Jiebo Luo, Houqiang Li, and Shipeng Li. Finding perfect rendezvous on the go: accurate mobile visual localization and its applications to routing. In *ACM Multimedia*, pages 9–18, 2012.

[65] Yu Zheng and Xing Xie. Learning travel recommendations from user-generated gps traces. *ACM TIST*, 2(1):2, 2011.

[66] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092, 2009.

[67] Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting flickr tags and groups for finding landmark photos. In *ECIR*, pages 654–661, 2009.

[68] Lyndon S Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *WWW*, pages 297–306, 2008.

[69] Yuheng Ren, Mo Yu, Xin-Jing Wang, Lei Zhang, and Wei-Ying Ma. Diversifying landmark image search results by learning interested views from community photos. In *WWW*, pages 1289–1292, 2010.

[70] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Brian Curless, Steven M Seitz, and Richard Szeliski. Reconstructing rome. *IEEE Computer*, 43(6):40–47, 2010.

[71] Keisuke Hasegawa, Qiang Ma, and Masatoshi Yoshikawa. Trip tweets search by considering spatio-temporal continuity of user behavior. In *DEXA (2)*, pages 141–155, 2012.

[72] Yan Ke, Xiaoou Tang, and Feng Jing. The design of high-level features for photo quality assessment. In *CVPR (1)*, pages 419–426, 2006.

[73] Jose San Pedro and Stefan Siersdorfer. Ranking and classifying attractiveness of photos in folksonomies. In *WWW*, pages 771–780, 2009.

[74] Andreas E Savakis, Stephen P Etz, and Alexander CP Loui. Evaluation of image appeal in consumer photography. In *Electronic Imaging*, pages 111–120. International Society for Optics and Photonics, 2000.

[75] Stefan Winkler. Visual fidelity and perceived quality: Toward comprehensive metrics. In *Photonics West 2001-Electronic Imaging*, pages 114–125. International Society for Optics and Photonics, 2001.

[76] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[77] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1226–1238, 2002.

[78] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD*, pages 226–231, 1996.

[79] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[80] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[81] Xin Liu, Yong Liu, and Xiaoli Li. Exploring the context of locations for personalized location recommendations. In *IJCAI*, pages 1188–1194, 2016.

[82] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *ACM HT*, pages 119–128, 2013.

[83] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[84] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[85] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, 2014.

[86] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[87] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *WWW*, pages 271–280, 2012.

[88] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934, 2013.

[89] Tim Van de Cruys. A neural network approach to selectional preference acquisition. In *EMNLP*, pages 26–35, 2014.

[90] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[91] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. *ICwSM*, 11:70–573, 2011.

[92] Jonathan Reades, Francesco Calabrese, Andres Sevtsuk, and Carlo Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3), 2007.

[93] Alberto Pascual-Montano, Jose Maria Carazo, Kieko Kochi, Dietrich Lehmann, and Roberto D Pascual-Marqui. Nonsmooth nonnegative matrix factorization (nsnmf). *IEEE transactions on pattern analysis and machine intelligence*, 28(3):403–415, 2006.

[94] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, pages 601–610, 2014.

[95] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. *arXiv preprint arXiv:1609.05566*, 2016.

[96] Quan Yuan, Gao Cong, and Aixin Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *CIKM*, pages 659–668, 2014.

[97] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[98] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[99] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887, 2008.

[100] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[101] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *ACM SIGKDD*, pages 316–324, 2011.

[102] Yao Wu, Xudong Liu, Min Xie, Martin Ester, and Qing Yang. Cccf: Improving collaborative filtering via scalable user-item co-clustering. In *WSDM*, pages 73–82, 2016.

[103] John Krumm and Eric Horvitz. Eyewitness: Identifying local events via space-time signals in twitter feeds. In *ACM SIGSPATIAL*, pages 20:1–20:10, 2015.

[104] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860, 2010.

[105] Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang, and Junjie Yao. A unified model for stable and temporal topic detection from social media data. In *ICDE*, pages 661–672, 2013.

[106] Hao Wang, Binyi Chen, and Wu-Jun Li. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, pages 2719–2725, 2013.

[107] Hao Wang and Wu-Jun Li. Relational collaborative topic regression for recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1343–1355, 2015.

[108] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *ACM SIGKDD*, pages 605–613, 2013.

SELECTED LIST OF PUBLICATIONS

- **Journals**

  [1] Chenyi Zhuang, Qiang Ma, Masatoshi Yoshikawa. SNS User Classification and its Application for Discovering Obscure POIs. *Multimedia Tools and Applications*, Volume 76, Issue 4, pp. 5461-5487, February 2017.

  [2] Yizhu Shen, Min Ge, Chenyi Zhuang, Qiang Ma. Sightseeing Value Estimation by Analyzing Geosocial Images. *International Journal of Big Data Intelligence*, 2016. (Accepted, to appear)

- **International Conferences**

  [3] Chenyi Zhuang, Nicholas Jing Yuan, Ruihua Song, Xing Xie, Qiang Ma. Understanding People Lifestyles: Construction of Urban Movement Knowledge Graph from GPS Trajectory. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. (Accepted, to appear)

  [4] Yizhu Shen, Min Ge, Chenyi Zhuang, Qiang Ma. Sightseeing Value Estimation by Analyzing Geosocial Images. *IEEE International Conference on Multimedia Big Data (BigMM)*, pp. 117-124, 2016.

  [5] Chenyi Zhuang, Qiang Ma, Masatoshi Yoshikawa. Location Familiarity Based Flickr Photographer Classification for POI Mining. *ACM International Conference on Advances in Geographic Information System (ACM SIGSPATIAL)*, Article No. 84, November 2015.

  [6] Chenyi Zhuang, Qiang Ma, Xuefeng Liang, Masatoshi Yoshikawa. Discovering Obscure Sightseeing Spots by Analysis of Geo-tagged Social Images. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 590-595, 2015.

[7] Chenyi Zhuang, Qiang Ma, Xuefeng Liang, Masatoshi Yoshikawa. Anaba: An Obscure Sightseeing Spots Discovering System. *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, 2014.

[8] Chenyi Zhuang, Jingyi Li, Guoping Rong, Dong Shao. APIS A Web-based System for PSP/TSP. *ENASE EAST*, pp. 68-74, 2011.

- **Domestic Conferences and Workshops**

[9] Hajime Takeda, Chenyi Zhuang, Qiang Ma. Feature Analysis of Traders on Social Trading Services. *The 9th Forum on Data Engineering and Information Management (DEIM)*, 2017.

[10] Chenyi Zhuang, Qiang Ma, Xuefeng Liang, Masatoshi Yoshikawa. Discover Anaba Sightseeing Spots Using Social Images. *IPSJ Database System (DBS)*, 2013.

APPENDIX

A Proof of Corollary 1

Let $\mathscr{D}$ be a set of observed data generated from some distributions parameterized by Θ. The hyperparameters are denoted as Ω.

Proof. In MAP, denoted as $\hat{\Theta}_{MAP}$, we maximize $P(\Theta|\mathscr{D})$ directly. According to Bayes Rule, we have

$$\begin{aligned}\hat{\Theta}_{MAP} &= \arg\max_{\Theta} \log p(\Theta|\mathscr{D}) \\ &= \arg\max_{\Theta} \log p(\mathscr{D}|\Theta)p(\Theta|\Omega) \\ &= \arg\max_{\Theta} \log \prod_{d_i \in \mathscr{D}} p(d_i|\Theta) \cdot p(\Theta|\Omega) \\ &= \arg\max_{\Theta} \sum_{d_i \in \mathscr{D}} \{\log p(d_i|\Theta)\} + \log p(\Theta|\Omega).\end{aligned} \tag{7.1}$$

We notice that, when the prior distribution parameterized by Ω is given, MAP estimate is equivalent to the complete ML estimate , i.e. the maximum likelihood of the observed data $\mathscr{D}$ (i.e., $\sum_{d_i \in \mathscr{D}}\{\log p(d_i|\Theta)\}$) and the model's parameters Θ (i.e., $\log p(\Theta|\Omega)$). □

B Derivation of the Update Equations for Eq. 6.8

Since the derivations for updating $\vec{u}_i$, $\vec{v}_j$ and $\vec{p}_n$ in Eq. 6.8 are very similar, we only list the detailed derivation for the latent moving pattern parameter $\vec{p}_n$.

We first simplify the gradient of ℓ with respect to $\vec{p}_n$, i.e., Eq. 6.11.

$$\begin{aligned}\frac{\partial \ell}{\partial \vec{p}_n} &= \sum_n [\sum_j \lambda_T \vec{v}_j d_{j,n} (T_{j,n} - \vec{v}_j{}^T \vec{p}_n) - \lambda_P I_k \vec{p}_n{}^T] \\ &= \sum_n [\sum_j \lambda_T \vec{v}_j d_{j,n} T_{j,n} - \sum_j \lambda_T \vec{v}_j d_{j,n} \vec{v}_j{}^T \vec{p}_n - \lambda_P I_k \vec{p}_n{}^T] \\ &= \sum_n [\lambda_T V D_n T_n - \vec{p}_n (\lambda_T V D_n V^T + \lambda_P I_k)].\end{aligned} \tag{7.2}$$

Then, by setting each entry in the sum to zero, we have the updating equation for a particular $\vec{p}_n$:

$$\vec{p}_n \leftarrow (\lambda_T V D_n V^T + \lambda_P I_k)^{-1} (\lambda_T V D_n T_n). \tag{7.3}$$