

Personalized POI Recommendation on Location-Based Social Networks

by

Huiji Gao

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

Approved November 2014 by the
Graduate Supervisory Committee:

Huan Liu, Chair
James Caverlee
Guoliang Xue
Jieping Ye

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

The rapid urban expansion has greatly extended the physical boundary of our living area, along with a large number of POIs (points of interest) being developed. A POI is a specific location (e.g., hotel, restaurant, theater, mall) that a user may find useful or interesting. When exploring the city and neighborhood, the increasing number of POIs could enrich people’s daily life, providing them with more choices of life experience than before, while at the same time also brings the problem of “curse of choices”, resulting in the difficulty for a user to make a satisfied decision on “where to go” in an efficient way. Personalized POI recommendation is a task proposed on purpose of helping users filter out uninteresting POIs and reduce time in decision making, which could also benefit virtual marketing.

Developing POI recommender systems requires observation of human mobility w.r.t. real-world POIs, which is infeasible with traditional mobile data. However, the recent development of location-based social networks (LBSNs) provides such observation. Typical location-based social networking sites allow users to “check in” at POIs with smartphones, leave tips and share that experience with their online friends. The increasing number of LBSN users has generated large amounts of LBSN data, providing an unprecedented opportunity to study human mobility for personalized POI recommendation in spatial, temporal, social, and content aspects.

Different from recommender systems in other categories, e.g., movie recommendation in NetFlix, friend recommendation in dating websites, item recommendation in online shopping sites, personalized POI recommendation on LBSNs has its unique challenges due to the stochastic property of human mobility and the mobile behavior indications provided by LBSN information layout. The strong correlations between geographical POI information and other LBSN information result in three major human mobile properties, i.e., geo-social correlations, geo-temporal patterns, and

geo-content indications, which are neither observed in other recommender systems, nor exploited in current POI recommendation. In this dissertation, we investigate these properties on LBSNs, and propose personalized POI recommendation models accordingly. The performance evaluated on real-world LBSN datasets validates the power of these properties in capturing user mobility, and demonstrates the ability of our models for personalized POI recommendation.

DEDICATION

I dedicate my dissertation work to my loving parents, Haodu Gao and Lixing Zhang, for making me be who I am!

I also dedicate this dissertation to my precious wife, Ye Wu, for supporting me all the way! Without her help and encouragement it simply never would have been.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Huan Liu, for his consistent support during my Ph.D. study. Dr. Liu helped to cultivate my research strength, provide invaluable advices in the last five years. Working with him, I learned to write my first paper and addressed many challenging problems. More importantly, I learned from him the work attitude and disciplines that I will benefit all my life. I could not have imagined having a better advisor for my Ph.D study. Dr. Liu is more of a mentor and life long friend, than a professor.

I would like to thank my thesis committee, Dr. James Caverlee, Dr. Guoliang Xue, and Dr. Jieping Ye, for the assistance, encouragement, and insightful comments that they have provided at all levels of my research.

I would like to thank Dr. Jalal U. Mahmud (IBM Research) and Dr. Kun Liu (LinkedIn), for providing me the summer internship opportunities in their groups and mentoring me on diverse exciting research projects.

I would like to thank my colleagues at the Data Mining and Machine Learning Lab for their helpful suggestions and support. It is a great pleasure working with them, particularly, Jiliang Tang, Xia Hu, Lei Tang, Zheng Zhao, Geoffery Barbier, Shamanth Kumar, Fred Morstatter, Ali Abbasi, Reza Zafarani, Pritam Gundecha, Isaac Jones, Zhuo Feng, Suhas Ranganath, and Salem Alelyani.

I would like to thank the Office of Naval Research (ONR) and Arizona State University for their continuous support through the grant (No. N000140810477, No. N000141010091, and No. N000141110527) and Graduate Education Dissertation Fellowship during my entire Ph.D. study.

Last but not the least, I would like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my wife Ye Wu, without whose love and encouragement, I would not have finished this dissertation.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER	
1 INTRODUCTION	1
1.1 Background	1
1.1.1 Geographical Properties of Social Connections	4
1.1.2 Temporal Patterns of Geographical Check-ins	5
1.1.3 Semantic Indications of Check-in Content	6
1.2 Problem Statement	7
1.3 Contributions	7
1.4 Organization	9
2 LITERATURE REVIEW	10
2.1 General Recommender Systems	10
2.2 Personalized POI Recommendation	13
2.2.1 Personalized POI recommendation with GPS data	13
2.2.2 Personalized POI recommendation with LBSN data	16
3 PERSONALIZED GEO-SOCIAL POI RECOMMENDATION	19
3.1 Defining Geo-Social Correlations	19
3.2 gSCorr: Location Recommendation with Geo-Social Correlations ...	22
3.2.1 Modeling Geo-Social Correlation Strengths	22
3.2.2 Modeling Geo-Social Correlation Probabilities	25
3.2.3 Parameter Inference	26
3.3 Evaluating gSCorr	28
3.3.1 Data Collection	28

CHAPTER	Page
3.3.2	Experiment Setup..... 29
3.3.3	Geo-Social Correlation Measure Selection 31
3.3.4	Performance of gSCorr 34
3.3.5	Effect of Geo-Social Correlation Strengths and Measures 37
4	PERSONALIZED GEO-TEMPORAL POI RECOMMENDATION 40
4.1	Temporal Cyclic Patterns 40
4.1.1	Temporal Non-uniformness and Consecutiveness..... 41
4.1.2	POI Recommendation with Geo-Temporal Patterns 43
4.1.3	LRT: Location Recommendation with Temporal Effects..... 46
4.1.4	Experiments 50
4.1.5	Dataset and Experiment Setup..... 50
4.2	Temporal Chronological Patterns 59
4.2.1	Modeling Power-Law Distribution and Short-Term Effect ... 60
4.2.2	HM: Historical Model 65
4.2.3	SHM: Social-Historical Model..... 66
4.2.4	Experiments 68
4.2.5	Dataset and Experiment Setup..... 68
4.3	Complementary Effect of Temporal Cyclic and Chronological Patterns 76
5	PERSONALIZED GEO-CONTENT POI RECOMMENDATION 78
5.1	A POI Recommendation Model with Geo-Content Indications 79
5.1.1	Modeling User Sentiment Indications 80
5.1.2	Modeling User-Interest Content and POI-Property Content . 82
5.1.3	CAPRF: Content-Aware POI Recommendation Framework . 83
5.1.4	Parameter Estimation 84

CHAPTER	Page
5.1.5	Algorithm Analysis 88
5.2	Experiments..... 90
5.2.1	Foursquare Dataset 91
5.2.2	Experimental Setup 93
5.2.3	Performance Evaluation 95
5.2.4	Evaluation of Different Types of Content Information..... 99
5.2.5	Parameter Analysis 100
6	CONCLUSION AND FUTURE WORK..... 106
	REFERENCES 109
	BIOGRAPHICAL SKETCH 117

LIST OF TABLES

Table	Page
1.1 Facets of Check-in Actions w.r.t. Content Information	7
3.1 Geo-Social Correlations	22
3.2 Check-in and Social Features	24
3.3 Statistical Information of the Dataset	32
3.4 Statistical Information of the July Data	32
3.5 Location Recommendation for Measure Selection on $S_{F\bar{D}}$	34
3.6 Location Recommendation for Measure Selection on S_{FD}	34
3.7 Location Recommendation for Measure Selection on $S_{\bar{F}\bar{D}}$	35
3.8 Performance Comparison for Location Recommendation	37
3.9 Evaluation Metrics	38
3.10 POI Recommendation with Different Geo-Social Correlation Strengths and Measures	39
4.1 Statistical Information of the Dataset	52
4.2 Performance of Random Recommendation	56
4.3 Comparison of Aggregation Strategies (Precision)	57
4.4 Comparison of Aggregation Strategies (Recall)	57
4.5 Comparison of Temporal Patterns	58
4.6 Corresponding Features between Language and LBSN Modeling	61
4.7 Average Number of Check-ins between Two Users	66
4.8 Statistical Information of the Dataset	69
4.9 Number of Unique Check-ins at Each Time Point	76
5.1 Mathematical Notation	85
5.2 Statistical Information of the Dataset	92
5.3 Performance Comparison (CA)	97

Table	Page
5.4 Performance Comparison (NY)	97
5.5 Recommendation Effect of Different Types of Content Information	101
5.6 Recommendation Effect of Different Types of Content Information	101

LIST OF FIGURES

Figure	Page
1.1 The Information Layout of Location-Based Social Networks	3
1.2 Illustration of Personalized POI Recommendation on LBSNs	8
3.1 Empirical Cumulative Distribution (CDF) of Geographic Distance between Users and between Friends([67])	20
3.2 Probability of Friendship between Two Users w.r.t. Their Geographic Distance ([67])	21
3.3 The Geo-Social Correlations of New Check-in Behavior	21
3.4 Observed Social Correlations on New Check-ins	23
3.5 The User Distribution over the World	29
3.6 The User Distribution over the USA	30
4.1 Geo-Temporal Patterns of Check-in Behavior	40
4.2 Daily Check-in Activities on LBSN	41
4.3 POI Recommendation Framework with Geo-Temporal Patterns	46
4.4 Recommendation Performance (Precision)	55
4.5 Recommendation Performance (Recall)	56
4.6 Power-law Distribution of Check-ins from All the Users	59
4.7 Power-law Distribution of Individual Check-ins	60
4.8 The Generating Process of Check-in Sequence	64
4.9 Performance Comparison of Recommendation Models	72
4.10 The Performance of Social-historical Model w.r.t. η (T3)	75
4.11 The Performance of Social-historical Model w.r.t. η (T6)	75
4.12 The Performance of Social-historical Model w.r.t. η (T9)	76
5.1 Content Information on LBSNs	79
5.2 Content-Aware POI Recommendation Framework	84

Figure	Page
5.3 Check-in Distribution over the California State.....	92
5.4 Check-in Distribution over the New York State.....	93
5.5 Sentiment Indications- η	103
5.6 User-Interest Content- λ_1	104
5.7 POI-Property Content- λ_2	104
5.8 Semantic Overlapping- δ	105

Chapter 1

INTRODUCTION

The rapid growth of cities has developed an increasing number of points of interest (POIs), e.g., restaurants, theaters, stores, hotels, to enrich people’s life and entertainment, providing us with more choices of life experience than before. People are willing to explore the city and neighborhood in their daily life and decide “where to go” according to their personal interest and the various choices of POIs. At the same time, making a satisfying decision efficiently among the large number of POI choices becomes a tough problem for a user. To facilitate a user’s exploration and decision making, POI recommendation has been introduced by location-based services such as Yelp¹ and Foursquare². However, such recommendation models are commonly based on majority users’ preference on POIs, which ignore a user’s personal preference. Comparing to visiting places that best fit a user’s interest, visiting places against a user’s taste may give him very terrible experience, especially in a situation when the user travels to a new place. Therefore, personalized POI recommendation is proposed to help users filter out uninteresting venues according to their own taste and save their time in decision making.

1.1 Background

Before the Web 2.0 era, analyzing user’s mobility for personalized POI recommendation is infeasible even the mobile devices are widely adapted with large amount of cellphone-based GPS data available, as there is no indication of POI information from

¹<http://www.yelp.com>

²<http://foursquare.com>

the GPS data other than longitude and latitude records. For example, we could observe a set of locations in terms of longitude and latitude pairs that a user has been to, while there is no easy way to figure out whether a specific pair of longitude and latitude is corresponding to a restaurant, or a hotel, or just a point on highway, since all these information are passively recorded by mobile devices.

With the developing of Web 2.0 technology, a number of location-based social networking services, e.g., Foursquare, Yelp, and Facebook Places³, have emerged in recent years, making the study of personalized POI recommendation possible. Typical location-based social networking services maintain a large POI database and allow a user to “check-in” at a POI with his smartphone regarding to his current physical location. The user can also leave tips and share the “check-in” experience with his online friends, along with creating the opportunity to make new friends. According to a recent survey from the Pew Internet and American Life Project, over the past year 18% of smartphone owners use geosocial services to “check in” at certain locations and share them with their friends, while this percentage has risen from 12% in 2011 [102]. Such rapid growth has led to the availability of a large amount of user mobility data, promoting a new concept of online social media, namely location-based social networks (LBSNs).

Location-based social networks not only refer to the social connections among users, but also consist of the “location-based” context including geographical check-in POIs, check-in time stamps, and check-in related content (e.g., tips, comments, POI descriptions, etc.), as shown in Figure 1.1. Compared with other online social networks that consist of user activities interacting with the virtual world, LBSNs reflect a user’s geographical action in the real world, residing where the online world and real world intersect, therefore bridging the gap between the real world and the vir-

³<http://www.facebook.com/about/location>

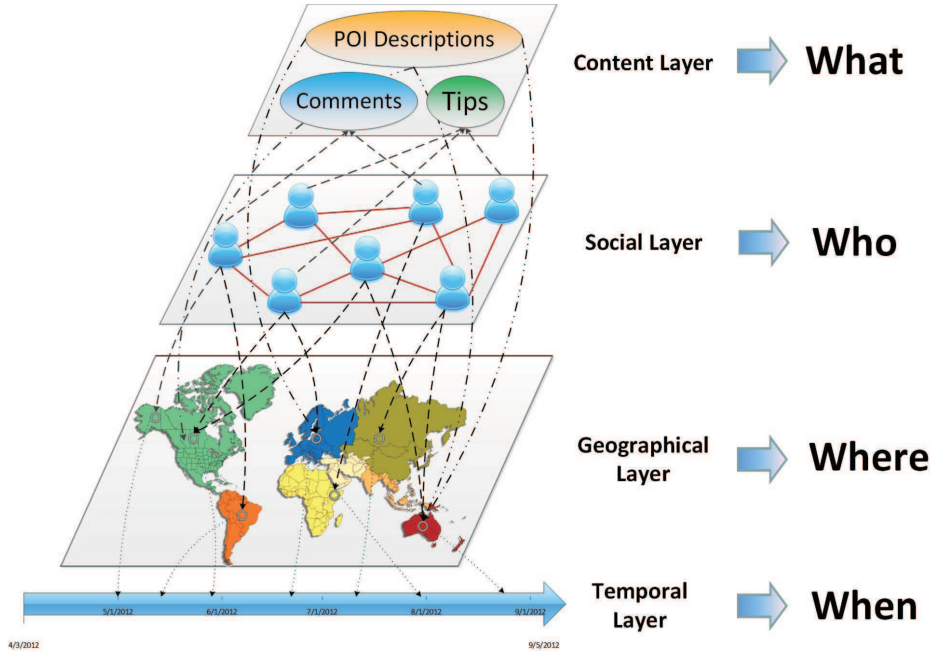


Figure 1.1: The Information Layout of Location-Based Social Networks

tual world, providing both opportunities and challenges for researchers to investigate users' check-in behavior for personalized POI recommendation in spatial (“where”), temporal (“when”), social (“who”) and content (“what”) aspects.

In the last decade, recommender systems have been widely studied among various categories, e.g., movie recommendation on NetFlix, dating recommendation on Zoosk, item recommendation on Amazon. However, it is not sufficient to directly apply these technologies as personalized POI recommendation on LBSNs presents unique challenges due to the heterogeneous information layout and the specificity of human mobility. Designing efficient POI recommendation approaches on LBSNs inevitably needs to consider the following properties.

1.1.1 Geographical Properties of Social Connections

Traditional social network analysis mainly studies network structure and properties without the consideration of geographical distance between nodes. Although the idea of “Death of Distance” proposed in 2011 claims that geographical distance plays a less important role due to the communication revolution and the rapid development of the Internet, which could make of our world a “global village” [7], studies on spatial structure of networks demonstrated that there is a strong correlation between geographical attributes and network properties, indicating the significance of considering the spatial properties of networks for future applications [25]. Researchers have further studied the distinctions between online and offline social networks [14], and discovered that geographical property does play important roles when constructing the social connection between two users especially in explaining their mobility in the physical world[67, 13].

As two special factors on LBSNs, Geographical property and social connections are coherent and affect each other in human behavior. For example, a user is more likely to be friends with other users who are geographically close to him, e.g, co-workers, colleagues. Likewise, a user may check-in at a location due to the influence from his friends, such as following friends’ suggestions to visit a restaurant, going out with friends for shopping, etc. Such coherence results in a new property, commonly referred to as socio-spatial properties.

In recommender systems, user similarity evaluates how similar two users’ preferences are, which is a significant measurement for recommendation especially social recommendation with collaborative filtering approaches. However, as discussed above, unlike regular social recommender systems, social connections on LBSNs exhibit unique geographical properties, providing a new dimension for computing user

similarity. Therefore, considering the social information together with the geographical property enables us to capture the user preferences more precisely in POI recommendation on LBSNs.

1.1.2 Temporal Patterns of Geographical Check-ins

As suggested in [83, 12, 51], human geographical movement exhibits strong temporal patterns and is highly relevant to the location property. For example, a user regularly goes to a restaurant for lunch around 12:00 pm, watches movie on Friday night, and shops during weekends. This is generally referred to as temporal cyclic patterns. Such temporal patterns are not widely observed in other recommender systems. For instance, it is not common to observe a user regularly watching a specific movie (e.g., Batman, Avatar) or purchasing a specific item (e.g., camera, cellphone) at specific hour of the day, or day of the week. (Although birthdays or holidays like Thanksgiving may affect human behavior a bit, they are not commonly considered).

On the other hand, the temporal information of check-in actions on LBSNs is also considered as an order indicator to connect check-ins chronologically for generating location trajectories [98, 48, 81]. This is commonly referred to as temporal chronological patterns. For example, a user may want to sip a cup of coffee at Starbucks before he goes to office; or watch a movie after dinner at a restaurant, and then relax at a bar.

In addition, temporal cyclic patterns and temporal chronological patterns are correlated to each other. Considering them together provides us a perspective to understand human mobility in terms of where a user would like to go at a specific time after his recent visits on other POIs. Thus, investigating the features embedded in temporal patterns enables us to better capture human check-in behavior, providing a potential opportunity to design more advanced POI recommender systems on LBSNs.

1.1.3 Semantic Indications of Check-in Content

Content information on LBSNs could be related to a user’s check-in action, providing a unique opportunity for POI recommendation. When checking-in at a POI, a user may leave tips or comments to express his attitude towards the POI. Such content indicates abundant information w.r.t. the user’s interested topics and personal preferences against various facets of the POI. For example, by observing a user’s comment on a Mexican restaurant discussing its spicy food, we observe the *User Interests* in spicy food. If the comment is actually a compliment, e.g., “Best spicy food ever!”, we could infer both the user’s *Sentiment Indications* and her interests.

On the other hand, a POI is commonly associated with descriptive tags. Through studying these tags, one can not only infer the POI’s property but also the interests of users who have checked-in at this POI. For example, by observing a POI’s description as “vegetarian restaurant”, we infer that the restaurant serves “vegetarian food” and users who check-in at this POI might be interested in the vegetarian diet. This is an example of *POI Properties*.

These three types of content information, i.e., *POI properties*, *User Interests*, and *Sentiment Indications*, are all related to a user’s check-in actions and provide conceptual interpretations to three facets of his check-in actions, as listed in Table 1.1. In recommender systems, user interests and target properties are the two essential elements in capturing a user’s action (e.g., check-in) on a target (e.g., POI) for recommendation [36], while user assessment has also been recognized as an important factor to gauge the check-in action for future recommendation [70]. Investigating them together makes it possible to infer how a user’s interests match a POI’s property and whether the user prefers to visit that POI. Thus, content information on LBSNs provides a conceptual perspective to investigate users’ check-in behavior, which in

Table 1.1: Facets of Check-in Actions w.r.t. Content Information

Content Information	Facets of Check-in Actions
POI Properties	What is this POI about?
User Interests	Am I interested?
Sentiment Indications	How good is this POI?

turn constitutes the key factors of recommender systems, suggesting its potential for improving POI recommendation.

1.2 Problem Statement

Let $\mathbf{u} = \{u_1, u_2, \dots, u_m\}$ be the set of users and $\mathbf{l} = \{l_1, l_2, \dots, l_n\}$ be the set of POIs where m and n are the numbers of users and POIs, respectively. The problem of personalized POI recommendation on LBSNs is defined as:

Given a user $u \in \mathbf{u}$, a set of POIs (locations) $\mathbf{l}_u \in \mathbf{l}$ that u has checked-in, recommend him some POIs for his future visits based on the LBSN context (e.g., social connections, content information of check-ins, time stamps of check-ins) related to him, as illustrated in Figure 1.2.

For ease of presentation, we use POI, venue, and location as interchangeable terms in this dissertation. The recommendation algorithms discussed in this work are designed for individuals. However, they can be easily extended for group recommendation with aggregation strategies [91].

1.3 Contributions

The properties discussed above, i.e., geographical properties of social connections, temporal patterns of geographical check-ins, and semantic Indications of check-in

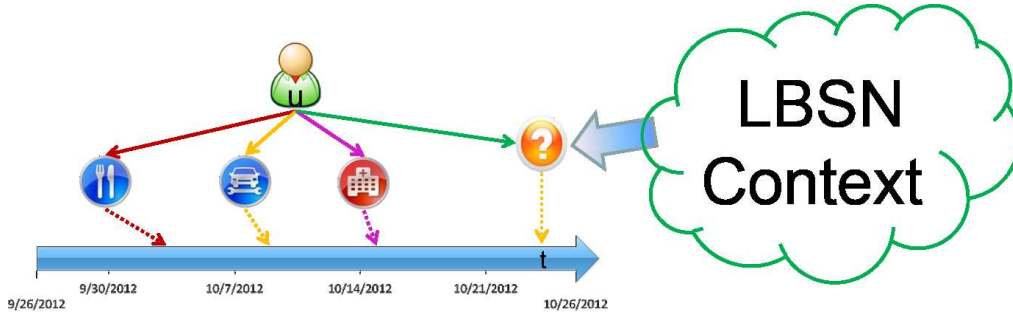


Figure 1.2: Illustration of Personalized POI Recommendation on LBSNs

Content, reveal the unique relationships of human behavior between geographical information and temporal, social, content information respectively, which are not commonly observed in other recommendation problems. In this dissertation, we study each property, and propose personalized POI recommender systems correspondingly, i.e., personalized geo-temporal recommendation, personalized geo-social recommendation, and personalized geo-content recommendation. To the best of our knowledge, this is the first work investigating these properties for POI recommendation on LBSNs. The contributions of our research are:

- Study the relationship between geographical check-ins and temporal information, model the temporal cyclic patterns and chronological patterns of a user’s check-in behavior, and propose geo-temporal POI recommender systems regarding to these patterns with their complementary effect.
- Investigate the geo-social correlations of user check-in behavior to solve the “cold-start” POI recommendation problem and propose personalized geo-social POI recommender systems.
- Identify the challenges of analyzing semantic indications of content information on LBSNs, propose models to leverage such information for personalized geo-

content POI recommendation.

1.4 Organization

The remainder of this dissertation is organized as follows. We first give a brief literature review in Chapter 2. From Chapter 3 to Chapter 5, we investigate the three LBSN properties to design personalized POI recommender systems. In Chapter 3, we introduce the personalized POI recommender system with geo-social correlations. We study the relationships between geographical distance and social friendships, and investigate them as a component w.r.t various facets. In chapter 4, we propose personalized geo-temporal POI recommender system. We study both temporal cyclic and temporal chronological patterns and their combinational effect. In Chapter 5, we analyze the user-generated content and POI-associated content, and leverage three types of content information including sentiment indications, user interests, and POI properties for personalized geo-content POI recommendation. We conclude the dissertation and point out promising research directions in Chapter 6.

Chapter 2

LITERATURE REVIEW

In the last decade, recommender systems have been widely studied among various categories, e.g., movie recommendation on NetFlix, job recommendation on LinkedIn, item recommendation on Amazon, news recommendation on Yahoo. POI recommendation, also referred to as location recommendation, has been recognized as an essential task on recommender systems for enriching human life experience and facilitating decision making, which belongs to a sub-category of recommender systems. Thus, technologies of general recommender systems are also practically applicable to location recommendation, although the performance may be limited due to the specific properties of human mobility on LBSNs. In the following sections, we first give a literature review on general recommender systems, and then review techniques of location-based recommender systems for personalized POI recommendation.

2.1 General Recommender Systems

Recommender systems refer to technologies that help users find items of interest among a large amount of items by generating personalized recommendations [1]. The techniques of recommender systems can be generally classified into three categories: collaborative filtering, content-based, and hybrid models. Among them, collaborative filtering (CF) is one of the most successful approaches, which has been proven effective in practise [65, 71]. It requires a user-item rating matrix (i.e., user-location check-in frequency matrix) as an input. The fundamental assumption of CF is that if two users have similar behavior on the similar items (e.g., watching similar movies, buying similar products, visiting similar restaurants, etc.), they will most likely have similar

behavior on other items in the future. In general, collaborative filtering approaches can be further classified into memory-based CF and model-based CF. The memory-based CF approach leverages the entire user-item rating matrix for recommendation, which has been adopted in many commercial systems. According to whose similarity it relies on to perform the recommendation, approaches contain user-based [29] and item-based [65]. The idea of user-based CF is to capture a user u 's preference on unvisited locations based on the preferences from K users most similar to him on locations. As an example of POI recommendation, it generally contains three steps:

1. Select K most similar users to u as his neighborhood \mathcal{N}_u .
2. Aggregate the preferences of users from $\mathcal{N}(u)$ on the locations not visited by u , deem them as u 's preferences.
3. Rank u 's preferences on those unvisited locations and select the top N locations for recommendation.

Analogously, item-based CF firstly finds K most similar locations and then calculates a weighted average of their check-in frequency.

Memory-based collaborative filtering approaches are efficient and easy to adopt. However, there are two shortfalls when it is applied to large-scale sparse data.

- **Sparsity**

In many real-world applications, the user-item matrix is usually very sparse with a density of 10^{-4} to 10^{-5} . Under sparse data, the similarity measured from ratings (or check-in frequency) may not be reliable due to the insufficient information observed [60]. In an extreme case of “cold-start” problem, a new user who has no rating/check-in history would have the similarity value of 0 to any other users.

- **Scalability**

Memory-based CF makes use of the whole user-item matrix to perform recommendation, which requires a large storage space. In addition, the computing of nearest neighbor is also computationally inefficient when the number of users/items is large.

Model based collaborative filtering approaches are proposed to address the above issues. They leverage data mining and machine learning technologies to learn a model from training data, and applies the model on test data to predict user interests on different items. Various models are investigated in this category, including latent factor models, classification/regression models, etc. Among all these models, latent factor models such as the matrix factorization model have been widely used [10].

The basic idea of the matrix factorization approach is to assume that there are certain latent factors related to both users' interests and locations' properties. As an example of restaurant in POI recommendation, latent factors could be taste, quality, environment, price, etc. These latent factors may dominate the occurrences of major check-in actions, with each check-in happening as a result of the combinational effect from a user's interests and a locations' property on these factors. For example, a user who likes seafood and is concerned with dining environment may be interested in a restaurant that serves fresh seafood with beautiful ocean view.

Classification/Regression-based recommender systems first generate training data consisting of (user, item) pairs. Then, features for users and items are extracted to construct the feature space. The observed (user, item) pair, i.e., a user has selected that item, is assigned with a positive label, while the unobserved (user, item) pair is assigned with a negative label. A classification/regression model is learned based on the training data through certain learning models, e.g., logistic regression, Decision Tree. To perform recommendation, the learned model is applied to a target pair of

(user, item) and outputs the likelihood of this user’s interest in that item [54].

Hybrid CF makes use of information other than the user-item matrix, e.g., content information, and combines CF with the content-based recommender systems. The results from each recommender system are weighted and combined into a final result for recommendation.

2.2 Personalized POI Recommendation

Existing work in POI recommendation contain generic approaches and personalized approaches. In generic approaches, POIs are commonly recommended based on their popularity [78, 8, 100], which is similar to some approaches in news recommendation [2]. Thus, generic POI recommendation recommend the same POIs for every user. In personalized approaches, recommendation is made according to a user’s personal preference; and different user receives recommendation of different POIs. Since our dissertation focuses on the personalized approach, in the following, we review several main methods of personalized POI recommendation.

2.2.1 *Personalized POI recommendation with GPS data*

The task of personalized POI recommendation is highly related to human mobility. It was traditionally studied on mobile data, i.e., cellphone-based GPS data. In the mobile era, cellphones have been widely used to facilitate humans’ life and communication. A user generally have his cell phone with him among most of the time. Thus, cellphones can be considered as mobile sensors of human beings, while data collected through these sensors could provide abundant information regarding human mobility. Typical cellphone-based GPS data contains a set of time-stamped GPS points that a user has been to, along with the mobile activities such as listening to music, generating bluetooth connections, browsing web pages, watching videos. Various existing

work has done on such data to study the human mobility, which promote a set of location-based applications including POI recommendation [72, 30, 5, 68, 61].

Due to the lack of mapping information between geo coordinates and specific real-world POIs, a POI is usually determined by the stay points extracted from hundreds of users' GPS trajectory logs [100, 95]. A GPS trajectory is a sequence of time-stamped latitude/longitude pairs, which are collected repeatedly at intervals of a short period (e.g., a few seconds). The stay points are geographical regions at which a user spent sufficient long time, and thus are considered as POIs (locations).

Since content and social information is usually not available on such datasets, spatial and temporal patterns are commonly adopted with collaborative filtering methods to perform POI recommendation. Leung et al. [39] studied different user classes and temporal preferences for collaborative location recommendation with a dynamic clustering algorithm. Zheng et al. [97] proposed a HITS-based inference model which takes into account both the location interests and users' travel experience with a tree-based hierarchical graph. Their recommendation effect was evaluated on a real-world GPS data over one-year period. Ye et al. [87] investigate individual life patterns from GPS data, which can be used for location prediction and recommendation. Ge et al. [27] use collaborative filtering based approach to recommend locations and travel packages with GPS trajectory data. Zheng et al. [99] proposed a recommendation framework for location recommendation and friend recommendation with the consideration of sequence property, region popularity, and hierarchical property of geographical spaces.

Content information could be obtained in certain types of the GPS data. Zheng et al. [94, 96, 95] proposed a user-centered collaborative location and activity filtering approach to find like-minded users and similar activities at different locations with tensor decomposition. The dataset is collected through voluntary users while user

comments regarding to geographical visits are used. In tour recommendation [26, 47] and tourist POI recommendation [35], content related to travel package or tourist POIs, such as package description or POI attributes, are used to analyze users' interested topics for POI recommendation.

Since GPS data is obtained from users' cell phones through telecommunication services, user privacy is a big concern which limits the data availability. Majority users do not feel comfortable to share their mobile data even for research purposes. Thus, GPS data usually contains a limited number of users over long period [17]. Gao et al. [19] summarized the characterises and limitations of leveraging GPS data for POI recommendation, as listed below.

- **Small-Scale Mobility Data**

Due to the user privacy concerns, cellphone-based GPS data generally contain a small number of users, which usually cannot be public available. The observations on such data may be biased due to certain factors such as region, demography, gender, age, education. For generating statistically significant conclusions especially in big data era, more data are encouraged when analyzing human mobility.

- **Absence of Semantic Indications**

GPS data store location information in terms of geographical coordinates, i.e., latitude and longitude. It is not straightforward to associate such coordinates with real-world points of interests, e.g., restaurants, hotels, theaters, malls. Generally, semantic information of locations is not available on GPS data. Although one can use third-party library to map coordinates into POIs, it does not work well on places with dense POIs, as it is difficult to distinguish POIs close to each other based on geographical coordinates. Furthermore, even via

observing that a user has stopped at a geographical point, it is not easy to determine whether he was visiting the corresponding POI or just passing by.

- **Insufficient Social Information**

Social connections are not easily obtained from GPS data. Generally, social connections can be inferred through the history of one’s phone calls, messages, or bluetooth connections. However, it is difficult to collect this kind of data due to the privacy concerns. There are work collecting social information on GPS data through communication network or bluetooth network with a number of participated users who grant permissions [17, 40, 79]. However, social information obtained in this way maybe in low quality. For example, bluetooth may not be commonly used thus connections inferred through this way maybe biased; users who have phone communications do not necessarily indicate their friendships, not to mention that they share common interests of locations.

2.2.2 *Personalized POI recommendation with LBSN data*

With the rapid development of location-based social networking services, users are able to check-in at real-world POIs and share such check-ins with their friends through mobile devices, resulting in more abundant spatial, temporal, social, and content information to improve personalized POI recommendation. Ye et al. [85] introduced POI recommendation into LBSNs. Due to the strong correlations between geographical distance and social connections discovered in previous work [12, 13, 67, 92, 22], current work on POI recommendation on LBSNs mainly focuses on leveraging the geographical and social properties to improve recommendation effectiveness.

Techniques of personalized POI recommendation with geographical influence and social connections mainly study these two elements separately, and then combine their output together with a fused model. The social influence is usually modeled through

friend-based collaborative filtering with either memory-based approaches [101] or model-based approaches [84, 80, 49]. Ye et al. [86] investigated the geographical influence with a power-law distribution. The hypothesis is that users tend to visit places in short distance. Cho et al. [13] and Cheng [10] investigated the geographical influence through a multi-center Gaussian model. Zhang et al. [93] proposed a Kernel density estimation method to model the geographical influence without knowing a specific type of distribution. All these work further combine the geographical influence with social influence through a fused model based on the sum rule or the product rule for POI recommendation.

There are also work using joint model to study geographical influence and social connections for personalized POI recommendation. Ying et al. [89] proposed a set of features related to social factor, individual preference, and location popularity, and utilized a regression-tree model to recommend POIs. Gao et al. [23] studied the two factors as a component, named as geo-social correlations, to solve the POI recommendation problem on LBSNs.

Among the current work on LBSNs, temporal information has also attracted much attention from researchers. Ye et al. [83] introduced temporal dimension of daily and weekly check-ins to identify the types of unknown geographic target on LBSNs. Chang et al. [9] proposed a logistic regression model with observed temporal patterns as one type of feature. POI recommendation with temporal effects mainly leverages temporal cyclic patterns and temporal chronological patterns on LBSNs. Cheng et al. [11] introduced the task of successive personalized POI recommendation in LBSNs by embedding the temporal chronological patterns and localized regions into a matrix factorization method. Gao et al. [22] used a Hierarchical Pitman-Yor language model to capture the temporal chronological patterns with the consideration of power-law distribution and short-term effect. Ye et al [82] studies the chronological patterns with

a Hidden Markov model. For temporal cyclic patterns, Gao et al. [21] investigated their properties in terms of temporal non-uniformness and temporal consecutiveness with a matrix factorization model. Yuan et al. [90] incorporated both temporal cyclic information and geographical information through a unified framework for time-aware POI recommendation. Gao et al. [21] discovered the distribution of cyclic patterns and proposed a Gaussian Mixture model for personalized POI recommendation.

Most recently, researchers have started exploring the content information on LBSNs for POI recommendation. Current work of content-aware POI recommendation focuses on one of the three types of content information, i.e., POI-property content, user-interest content, and user sentiment indications. Yang et al. [16] introduced sentiment information into POI recommendation and reported its better performance over state-of-the-art approaches. Hu et al. [31] investigated the user-interest content from Twitter and Yelp, and proposed a topic model for POI recommendation considering both the spatial aspect and textual aspect of user posts. Liu et al. [43, 44] studied the effect of POI-associated tags for POI recommendation with an aggregated LDA model and matrix factorization method. Hu et al. [32] incorporated content information into social correlations and proposed a topic model for POI recommendation. Yuan et al [90] studied content information with its spatio-temporal patterns. Yin et al. [88] investigated both personal interest and local preference in terms of item-based content on LBSNs and EBSNs. All of this work focuses on one type of the content information without considering the other two and their correlations.

PERSONALIZED GEO-SOCIAL POI RECOMMENDATION

Geographical property and social connections are two special factors on LBSNs. The geographical property reflects human behavior in real world, which distinguishes location-based social networks from content-based social networks [66]. The explicit social network information, which is generated by users through the “add friend” action, distinguishes location-based social networks from cellphone data. On cellphone-based GPS data, social information is commonly collected by user study [40, 17] or inferred from communication networks through the calling/messaging actions or bluetooth connections [79].

3.1 Defining Geo-Social Correlations

Researchers have investigated how geographical distance influences social networks, and how social networks influence human movement on LBSNs. One study on three location-based social networking sites (Brightkite, Foursquare, and Gowalla) discovers strong heterogeneity across users at different geographic scales of interactions across social ties. The probability of a social tie between two users is roughly a function of the geographical distance between them [67]. The study on LBSN data and cell phone data reports that long-distance travel is more influenced by social friendship while short-range human movement is not influenced by social networks [13]. More recently, the investigation [37] on twitter social network concludes that offline geography still matters in online social networks, while one third of the users would like to have their social links in other countries, which is consistent with the previous findings in [42, 66]. Figure 3.1 and Figure 3.2 present the probability of a social tie

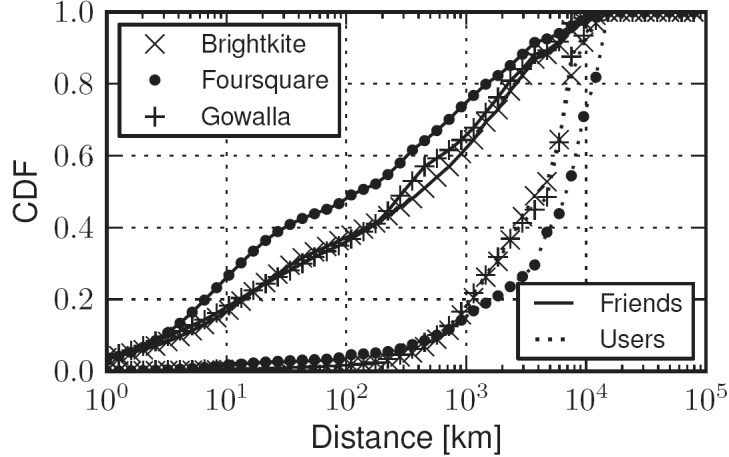


Figure 3.1: Empirical Cumulative Distribution (CDF) of Geographic Distance between Users and between Friends([67])

between two users w.r.t. their geographical distance on three typical LBSN datasets. It shows that users with short geographical distance are more likely to become friends than users with long geographical distance. On the other hand, friends usually have short geographical distance than non-friends.

The above influence between geographical distance and social networks is generally referred to as geo-social correlations. According to the different combinations of geographical distance and social strength between two users, we define four geo-social circles to represent four types of geo-social correlations, i.e., local friends $S_{F\bar{D}}$, distant friends S_{FD} , local non-friends $S_{\bar{F}\bar{D}}$, and distant non-friends $S_{\bar{F}D}$, as shown in Table 3.1. Figure 3.3 illustrates a user’s “new check-in” behavior with the corresponding geo-social correlations. User u goes to the airport at t_1 , and then the restaurant at t_2 followed by the hospital at t_3 . When u performs a “new check-in” at t_4 , i.e., the check-in POI does not belong to $\{L_1, L_2, L_3\}$, it may be correlated to those users that are from u ’s four geo-social circles $S_{F\bar{D}}$, S_{FD} , $S_{\bar{F}\bar{D}}$ and $S_{\bar{F}D}$, corresponding to the four types of geo-social correlations. Investigating these four circles

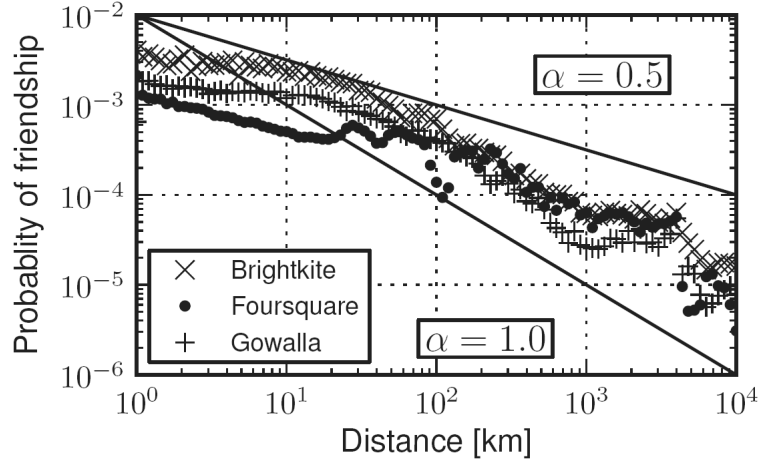


Figure 3.2: Probability of Friendship between Two Users w.r.t. Their Geographic Distance ([67])

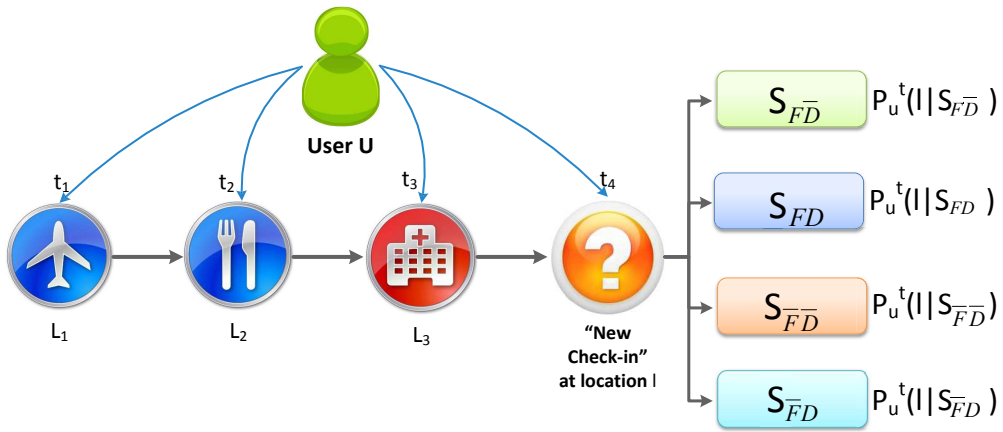


Figure 3.3: The Geo-Social Correlations of New Check-in Behavior

enables us to study a user’s check-in behavior in four corresponding aspects: local social correlation, distant social correlation, confounding, and unknown effect.

Table 3.1: Geo-Social Correlations

	F	\bar{F}
\bar{D}	$S_{F\bar{D}}$: Local Friends	$S_{\bar{F}\bar{D}}$: Local Non-friends
D	S_{FD} : Distant Friends	$S_{\bar{F}D}$: Distant Non-friends

3.2 gSCorr: Location Recommendation with Geo-Social Correlations

To model the geo-social correlations of “new check-in” behavior, we propose **gSCorr**, a recommendation model that generates the probability of a user u ’s check-in at a new POI l at time t , i.e., $P_u^t(l)$, as a combination of the four geo-social correlations, as defined below,

$$\begin{aligned}
 P_u^t(l) = & \Phi_1 P_u^t(l|S_{\bar{F}\bar{D}}) + \Phi_2 P_u^t(l|S_{F\bar{D}}) \\
 & + \Phi_3 P_u^t(l|S_{FD}) + \Phi_4 P_u^t(l|S_{\bar{F}D}).
 \end{aligned}
 \tag{3.1}$$

where Φ_1 , Φ_2 and Φ_3 and Φ_4 are four distributions that govern the strength of different geo-social correlations, $P_u^t(l|S_x)$ indicates the probability of user u checking-in at POI l that is correlated to u ’s geo-social circle S_x .

3.2.1 Modeling Geo-Social Correlation Strengths

The modeling of Φ_1 , Φ_2 and Φ_3 and Φ_4 is based on the observation of “new check-in” distribution in Figure 3.4, which indicates that Φ_1 is a real-valued and differentiable increasing function, and Φ_2 and Φ_3 are fairly constant. The percentage of new check-ins from $S_{\bar{F}D}$ is not presented, since it can be deduced from the other

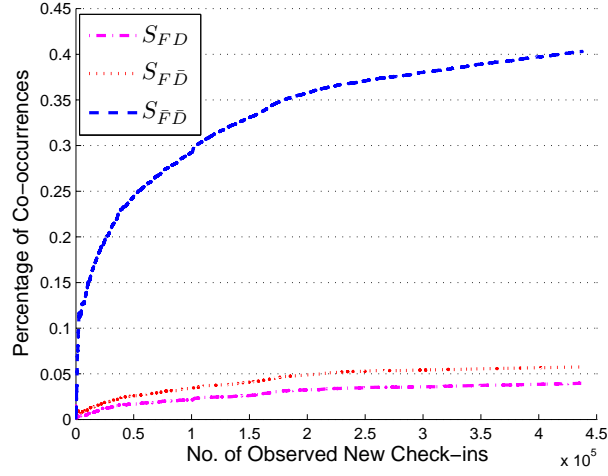


Figure 3.4: Observed Social Correlations on New Check-ins

three. Therefore,

$$\begin{aligned}
\Phi_1 &= f(w^T f_u^t + b), \quad 0 \leq \Phi_1 \leq 1 \\
\Phi_2 &= (1 - \Phi_1)\phi_1 \\
\Phi_3 &= (1 - \Phi_1)(1 - \phi_1)\phi_2 \\
\Phi_4 &= (1 - \Phi_1)(1 - \phi_1)(1 - \phi_2), \tag{3.2}
\end{aligned}$$

where f_u^t is a check-in feature vector of a single user u at time t , w is a vector of the weights of f_u^t , and b controls the bias. In this work, we define a user's check-in and social features f_u^t in Table 3.2.

Based on above definitions, we can rewrite the probability $P_u^t(l)$ in Eq. (3.1) as

Table 3.2: Check-in and Social Features

Features	Description
N^c	Number of check-ins in u 's history
N^{nc}	Number of new check-ins in u 's history
$N_{F\bar{D}}$	Number of friends in $S_{F\bar{D}}$
$N_{F\bar{D}}^c$	Number of check-ins from $S_{F\bar{D}}$
$N_{F\bar{D}}^{uc}$	Number of unique check-ins from $S_{F\bar{D}}$
$N_{F\bar{D}}^{vc}$	Number of visited check-ins from $S_{F\bar{D}}$
$N_{F\bar{D}}^{wvc}$	Number of visited unique check-ins from $S_{F\bar{D}}$
N_{FD}	Number of friends in S_{FD}
N_{FD}^c	Number of check-ins from S_{FD}
N_{FD}^{uc}	Number of unique check-ins from S_{FD}
N_{FD}^{vc}	Number of visited check-ins from S_{FD}
N_{FD}^{wvc}	Number of visited unique check-ins from S_{FD}
$N_{\bar{F}\bar{D}}$	Number of users in $S_{\bar{F}\bar{D}}$
$N_{\bar{F}\bar{D}}^c$	Number of check-ins from $S_{\bar{F}\bar{D}}$
$N_{\bar{F}\bar{D}}^{uc}$	Number of unique check-ins from $S_{\bar{F}\bar{D}}$
$N_{\bar{F}\bar{D}}^{vc}$	Number of visited check-ins from $S_{\bar{F}\bar{D}}$
$N_{\bar{F}\bar{D}}^{wvc}$	Number of visited unique check-ins from $S_{\bar{F}\bar{D}}$

shown below,

$$\begin{aligned}
P_u^t(l) &= f(\mathbf{w}^T \mathbf{f}_u^t + b) P_u^t(l|S_{\bar{F}\bar{D}}) \\
&+ (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) \phi_1 P_u^t(l|S_{F\bar{D}}) \\
&+ (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) (1 - \phi_1) \phi_2 P_u^t(l|S_{FD}) \\
&+ (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) (1 - \phi_1) (1 - \phi_2) P_u^t(l|S_{\bar{F}D}). \tag{3.3}
\end{aligned}$$

3.2.2 Modeling Geo-Social Correlation Probabilities

To capture the geo-social correlation probabilities $P_u^t(l|S_x)$, three geo-social correlation measures are proposed considering the factors of location frequency, user frequency and user similarity, as described below,

- **Sim-Location Frequency (S.Lf)**

$$P_u^t(l|S_x) = \frac{\sum_{v \in S_x} s(u, v) N_v^t(l)}{\sum_{v \in S_x} s(u, v) N_v^t}, \tag{3.4}$$

where $s(u, v)$ is the user similarity between user u and user v . $N_v^t(l)$ represents the number of check-ins at POI l by user v before time t , and N_v^t the total number of POIs visited by user v that user u has not visited before time t .

- **Sim-User Frequency (S.Uf)**

$$P_u^t(l|S_x) = \frac{\sum_{v \in S_x} \delta_v^t(l) s(u, v)}{\sum_{v \in S_x} s(u, v)}, \tag{3.5}$$

where $\delta_v^t(l)$ equals to 1 if user v has checked in at l before t , and 0 otherwise.

- **Sim-Location Frequency & User Frequency (S.Lf.Uf)**

$$P_u^t(l|S_x) = \frac{\sum_{v \in S_x} s(u, v) N_v^t(l)}{\sum_{v \in S_x} s(u, v) N_v^t} \frac{\sum_{v \in S_x} \delta_v^t(l)}{N_{S_x}}, \tag{3.6}$$

3.2.3 Parameter Inference

With the definitions described in the last section, we discuss the process of inferring the parameters defined in Eq. (3.3). We define (u, l, t) as a check-in action at location l performed by user u at time t , the likelihood of the observation over the whole data set is the product of the probability of each (u, l, t) action, defined as:

$$P(\mathcal{C}|\Theta) = \prod_{(u,l,t) \in \mathcal{C}} P_u^t(l), \quad (3.7)$$

where \mathcal{C} is the set of all the observed (u, l, t) actions, and Θ is the parameter set consisting of \mathbf{w} , b , ϕ_1 , ϕ_2 . We learn these parameters through maximum likelihood, which is equivalent to the following minimization problem:

$$\begin{aligned} \min \quad & \sum_{(u,l,t) \in \mathcal{C}} -\ln P(\mathcal{C}|\Theta) \\ & + \lambda(\|\mathbf{w}\|_2^2 + \|b\|_2^2 + \|\phi_1\|_2^2 + \|\phi_2\|_2^2) \end{aligned} \quad (3.8)$$

where parameter λ controls the quadratic regularized term to avoid overfitting. In this work, we set the value of λ as 0.05, and get the objective function below,

$$\begin{aligned} \min \quad & \sum_{(u,l,t) \in \mathcal{C}} -\ln \left(f(\mathbf{w}^T \mathbf{f}_u^t + b) P_u^t(l|S_{\bar{F}\bar{D}}) \right. \\ & + (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) \phi_1 P_u^t(l|S_{F\bar{D}}) \\ & + (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) (1 - \phi_1) \phi_2 P_u^t(l|S_{FD}) \\ & \left. + (1 - f(\mathbf{w}^T \mathbf{f}_u^t + b)) (1 - \phi_1) (1 - \phi_2) P_u^t(l|S_{\bar{F}D}) \right) \\ & + \lambda(\|\mathbf{w}\|_2^2 + \|b\|_2^2 + \|\phi_1\|_2^2 + \|\phi_2\|_2^2) \\ \text{s.t.} \quad & 0 \leq \phi_1 \leq 1, \quad 0 \leq \phi_2 \leq 1 \end{aligned} \quad (3.9)$$

We take the projected gradient method [6] to solve Eq. (3.9). The basic idea is to update each current parameter towards an optimal direction (determined by the first

derivative of the objective function) with an appropriate step size in each learning step. In each step, if the parameter value runs out of the constraints (e.g., $0 \leq \phi_1 \leq 1$, $0 \leq \phi_2 \leq 1$), we project it back to the corresponding range. The process will go iteratively to update the parameters until convergence. As shown below, the parameters are updated as,

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \gamma_{\mathbf{w}} \nabla_{\mathbf{w}} \\
b &\leftarrow b - \gamma_b \nabla_b \\
\phi_1 &\leftarrow \begin{cases} 0 & \phi_1 - \gamma_{\phi_1} \nabla_{\phi_1} < 0 \\ 1 & \phi_1 - \gamma_{\phi_1} \nabla_{\phi_1} > 1 \\ \phi_1 - \gamma_{\phi_1} \nabla_{\phi_1} & \text{else} \end{cases} \\
\phi_2 &\leftarrow \begin{cases} 0 & \phi_2 - \gamma_{\phi_2} \nabla_{\phi_2} < 0 \\ 1 & \phi_2 - \gamma_{\phi_2} \nabla_{\phi_2} > 1 \\ \phi_2 - \gamma_{\phi_2} \nabla_{\phi_2} & \text{else} \end{cases} \tag{3.10}
\end{aligned}$$

where $\gamma_{\mathbf{w}}$, γ_b , γ_{ϕ_1} and γ_{ϕ_2} are learning step sizes, which are chosen to satisfy Goldstein Conditions [53]. $\nabla_{\mathbf{w}}$, ∇_b , ∇_{ϕ_1} and ∇_{ϕ_2} are the partial derivatives of the objective function in Eq. (3.9) with respect to \mathbf{w} , b , ϕ_1 and ϕ_2 respectively,

$$\begin{aligned}
\nabla_{\mathbf{w}} &= 2\lambda\mathbf{w} - \sum_{(u,l,t) \in \mathcal{C}} \frac{B}{A} \frac{e_1}{(1+e_1)^2} \mathbf{f}_u^t \\
\nabla_b &= 2\lambda b - \sum_{(u,l,t) \in \mathcal{C}} \frac{B}{A} \frac{e_1}{(1+e_1)^2} \\
\nabla_{\phi_1} &= 2\lambda\phi_1 - \sum_{(u,l,t) \in \mathcal{C}} \frac{(1-\Phi_1)}{A} C \\
\nabla_{\phi_2} &= 2\lambda\phi_2 - \sum_{(u,l,t) \in \mathcal{C}} \frac{(1-\Phi_1)(1-\phi_1)}{A} D \tag{3.11}
\end{aligned}$$

where

$$\begin{aligned}
e_1 &= e^{-(\mathbf{w}^T \mathbf{f}_u^t + b)} \\
A &= \Phi_1 P_u^t(l|S_{\bar{F}\bar{D}}) + (1 - \Phi_1) \phi_1 P_u^t(l|S_{F\bar{D}}) \\
&+ (1 - \Phi_1)(1 - \phi_1) \phi_2 P_u^t(l|S_{FD}) \\
&+ (1 - \Phi_1)(1 - \phi_1)(1 - \phi_2) P_u^t(l|S_{\bar{F}D}), \\
B &= P_u^t(l|S_{\bar{F}\bar{D}}) - \phi_1 P_u^t(l|S_{F\bar{D}}) - (1 - \phi_1) \phi_2 P_u^t(l|S_{FD}) \\
&- (1 - \phi_1)(1 - \phi_2) P_u^t(l|S_{\bar{F}D}) \\
C &= P_u^t(l|S_{F\bar{D}}) - \phi_2 P_u^t(l|S_{FD}) - (1 - \phi_2) P_u^t(l|S_{\bar{F}D}) \\
D &= P_u^t(l|S_{FD}) - P_u^t(l|S_{\bar{F}D})
\end{aligned} \tag{3.12}$$

3.3 Evaluating gSCorr

In this work, we evaluate the performance of our proposed geo-social correlation model **gSCorr**. In particular, we evaluate the following: (1) how well the proposed geo-social correlation measures capture the geo-social correlation probabilities; (2) how the geo-social correlation strengths and measures affect the cold-start check-in behavior; and (3) whether social correlations help cold-start location recommendation. Before we delve into experiment details, we first discuss an LBSN dataset and experiment settings.

3.3.1 Data Collection

We use a Foursquare dataset to study the geo-social correlations of check-in behavior on location-based social networks. Foursquare is one of the most popular online LBSNs. It has more than 45 million members as of January, 2014¹ and keeps growing every month. The web site itself does not provide a public API to access

¹<https://foursquare.com/about>



Figure 3.5: The User Distribution over the World

users' check-in data, however, it provides an alternative way for users to link their twitter accounts with Foursquare, and then pop out the check-in messages as tweets to Twitter. Previous work [67, 22] uses this way to collect the data from Twitter for studying check-in behavior. Similarly, by getting access to the check-in tweets through the Twitter REST API, we collected public Foursquare check-in data from January 2011 to December 2011. We also collected the user friendships and hometown information through Foursquare. Note that the friendships on Foursquare are undirected. The statistics of the final dataset are shown in Table 3.3. The user distributions w.r.t. the world and the USA are given in Figure 3.5 and Figure 3.6.

3.3.2 Experiment Setup

We test our proposed model **gSCorr** on the data of each month from July to December respectively, with the corresponding training data from the previous 6

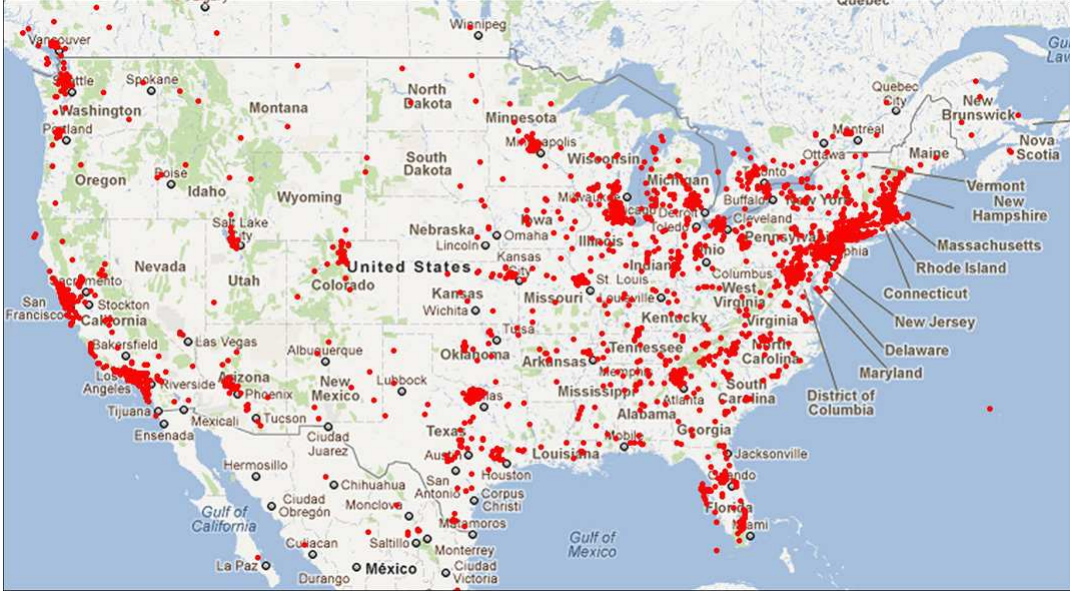


Figure 3.6: The User Distribution over the USA

months to learn our model parameters as in Eq. (4.5). For example, when testing **gSCorr** on September data, we use the data from March to August to train our recommendation model.

For each month from July to December, we construct its test set and ground truth based on the observation of their corresponding cold-start check-in distributions in four geo-social circles. Table 3.4 lists detailed statistical information of the observed cold-start check-in distribution in four geo-social circles on the check-in data in July. Due to the space limit, we do not present the statistical information from the other months since they all have the similar distributions. We define **“Social Co-occurrence Check-ins” (SCCs)** as the cold-start check-ins whose check-in locations can be found from the user’s different social circles before its checking in time. The check-in data in July contains 213,702 check-ins, with 77,581 cold-start check-ins performed at the locations that have never been visited before (the July test data is a closed set in the sense that it does not consider the historical check-ins

before July, as the same as the test data from other months). Among the 77,581 cold-start check-ins, around 44.5% SCCs can be found from the $S_{\bar{F}\bar{D}}$, 7.26% from $S_{F\bar{D}}$, 4.62% from S_{FD} and 50.82% from $S_{\bar{F}D}$. Only 10.61% SCCs are from a user’s direct friendship circle. In other words, only 8,235 among 77,581 cold-start check-ins co-occurred with check-ins of the user’s friendships. $S_{\bar{F}\bar{D}}$ has a large proportion of co-occurrences, indicating that user would like to go to a new location where his local non-friends in the state usually go. The number of SCCs of $S_{\bar{F}\bar{D}} \cup S_{F\bar{D}} \cup S_{FD}$ doesn’t increase much compared to $S_{\bar{F}\bar{D}}$, indicating that local non-friends have already covered most of the co-occurrences. Finally, we found that more than 50% of SCCs are correlated to $S_{\bar{F}D}$, which is difficult to capture for location prediction as the unknown effect. Note that there are 2.2% “Others”, indicating that at the time of check-in, 1,672 cold-start check-ins cannot be found from any of the four social circles. We consider this as an unknown effect and merge it into $S_{\bar{F}D}$.

We use location recommendation to evaluate our correlation measures and model performance. **User similarities** are computed based on the check-in data in the first half year by cosine similarity, while each user is represented by a check-in vector, and the entry in the vector indicates the visiting frequency of the user at the location. For each test month, the **test set** is selected as the SCCs of $S_{\bar{F}\bar{D}} \cup S_{F\bar{D}} \cup S_{FD}$, and the **ground truth** is the corresponding check-in locations. We do not consider $S_{\bar{F}D}$ because from a user’s perspective, friends and local non-friends are the ones that are reachable, while the distant non-friend users are too weak in relation.

3.3.3 Geo-Social Correlation Measure Selection

Before we discuss the performance of our proposed model **gSCorr**, we first evaluate the 3 geo-social correlation measures. Each measure can be directly applied to the test set and generates a ranking list of location probabilities $P_u^t(l|S_x)$ with re-

Table 3.3: Statistical Information of the Dataset

Duration	Jan 1, 2011-Dec 31, 2011
No. of users	11,326
No. of check-ins	2,290,997
No. of unique locations	187,218
No. of links	47,164
Average check-ins per user	202
Clustering coefficient	0.1560
Average degree	8.33

Table 3.4: Statistical Information of the July Data

Social Circle	No. of SCCs	Ratio
$S_{\bar{F}\bar{D}}$	34,523	44.50%
$S_{F\bar{D}}$	5,636	7.26%
S_{FD}	3,588	4.62%
$S_{\bar{F}D}$	39,423	50.82%
Others	1,672	2.2%
$S_{\bar{F}\bar{D}} \cup S_{F\bar{D}}$	35,277	45.47%
$S_{\bar{F}\bar{D}} \cup S_{FD}$	35,784	46.12%
$S_{F\bar{D}} \cup S_{FD}$	8,235	10.61%
$S_{\bar{F}\bar{D}} \cup S_{\bar{F}D} \cup S_{FD}$	36,486	47.03%

spect to the geo-social circles. We select the location with the highest $P_u^t(l|S_x)$ as recommended location for the cold-start check-in, and evaluate the performance with accuracy. The purpose of this comparison is to select the best correlation measure for each geo-social circle, and utilize the most suitable ones for $P_u^t(l|S_x)$ in Eq. (3.1). The results are shown in Table 3.5, Table 3.6, and Table 3.7 with some observations summarized below:

- **S.Lf.Uf** is the best measure for capturing the social correlations of local friends $S_{F\bar{D}}$. It also performs well on the other two geo-social circles especially on $S_{F\bar{D}}$. It considers the user frequency, location frequency and user similarities together, and obtains 1% relative improvement compared to the second best rated (**S.LF**), and 24.88% relative improvement compared to the worst rated (**Uf**).
- **S.Lf** shows good performance in capturing the social correlations of distant friends S_{FD} . It considers the location frequency and user similarity without the user frequency. One possible reason of this may be due to the smaller number of distant friends (2.68 per user on average) compared with the number of local friends (5.64 per user on average), which makes it a weak measure by counting the user frequency of distant friends.
- The performance on $S_{F\bar{D}}$ indicates that its best correlation measure is **S.Uf**, suggesting that a user would like to go to a location that has been visited by a large proportion of local non-friend users, no matter how frequently the location is visited by each individual user. This is consistent to the confounding effect that people who live in similar environment tend to share similar behavior, which is exactly the geo-social circle $S_{F\bar{D}}$ supposed to capture.

Due to the varied performances of each correlation measure on each geo-social circle, we conclude that measure selection is necessary for computing geo-social correla-

Table 3.5: Location Recommendation for Measure Selection on $S_{F\bar{D}}$

Ranking Strategy	$S_{F\bar{D}}$					
	Jul	Aug	Sep	Oct	Nov	Dec
S.LF	6.30%	6.73%	6.99%	7.32%	7.04%	7.90%
S.UF	5.38%	5.83%	5.77%	5.97%	5.96%	6.58%
S.LF.UF	6.51%	6.85%	7.02%	7.37%	7.11%	7.76%

Table 3.6: Location Recommendation for Measure Selection on S_{FD}

Ranking Strategy	S_{FD}					
	Jul	Aug	Sep	Oct	Nov	Dec
S.LF	3.65%	3.52%	4.15%	4.63%	4.37%	4.91%
S.UF	3.14%	3.00%	3.43%	3.86%	3.76%	4.01%
S.LF.UF	3.64%	3.57%	4.19%	4.56%	4.31%	4.64%

tion probabilities. Hence, we apply **S.Lf.Uf**, **S.Lf** and **S.Uf** to compute $P_u^t(l|S_{F\bar{D}})$, $P_u^t(l|S_{FD})$ and $P_u^t(l|S_{\bar{F}\bar{D}})$ respectively in the following experiments, considering their good performance on the corresponding geo-social circles. We do not report the results on $S_{\bar{F}\bar{D}}$, since for the unknown effect $P_u^t(l|S_{\bar{F}\bar{D}})$, all the measures applied to $S_{\bar{F}\bar{D}}$ perform as a random guess in our experiment, one possible reason may be the large number of users and candidate locations within this geo-social circle. Therefore, to reduce the time complexity, we consider $P_u^t(l|S_{FD})$ as a probability of a random jump to a location in current location vocabulary that user u has not checked-in before.

3.3.4 Performance of gSCorr

We compare **gSCorr** with four baselines, one is from the observation of the measure selection, and the other three are selected as the existing most popular location

Table 3.7: Location Recommendation for Measure Selection on $S_{\bar{F}\bar{D}}$

Ranking Strategy	$S_{\bar{F}\bar{D}}$					
	Jul	Aug	Sep	Oct	Nov	Dec
S.LF	17.84%	18.75%	18.60%	19.56%	19.72%	22.38%
S.UF	18.37%	19.40%	19.45%	20.21%	20.34%	22.82%
S.LF.UF	17.75%	18.86%	18.80%	19.74%	20.10%	22.34%

recommendation model on LBSNs.

- **S.LF.UF**: We select **S.LF.UF** to capture geo-social correlations and predict cold-start check-ins. It performs well on all the geo-social circles from Table 3.5 to Table 3.7, and achieve the best performance on $S_{\bar{F}\bar{D}}$ and many times on $S_{F\bar{D}}$. We apply it to the whole test set to predict the cold-start check-ins.
- **Periodic & Social Mobility Model (PSMM)**: **PSMM** ranks the locations based on a user’s periodic and social patterns [13]. Since the periodic patterns can only recommend existing locations, we adopt the social patterns to recommend the cold-start check-ins.
- **Social-Historical Model (SHM)**: **SHM** integrates a user’s historical ties and social ties to recommend/predict the next check-in location [22]. Similar to PSMM, we leverage the social model which utilizes the social ties to recommend cold-start check-in locations.
- **Collaborative Filtering (CF)**: **CF** is a state-of-the-art approach for recommender systems. It computes a user’s interest in a location based on other users’ interests in that location. Since it can recommend new locations to a user, we apply it to each test case of our test set and consider that a correct

recommendation happens when the recommended location is the same as the ground truth of the test case. We choose user-based collaborative filtering for such recommendation [71] as shown below:

$$P_u^t(l) = \frac{\sum_{v \in \mathcal{U}} s(u, v) r_{v,l}}{\sum_{v \in \mathcal{U}} s(u, v)}. \quad (3.13)$$

where \mathcal{U} is the set of users who have visited l , $r_{v,l}$ is the preference of user v on location l , which in our experiment is chosen as proportional to number of v 's check-ins on l normalized by v 's total number of check-ins, i.e., $\frac{N_v^t(l)}{N_v^t}$.

The results are shown in Table 3.8, we summarize several interesting observations as listed below:

- Both **PSMM** and **SHM** do not perform well in recommending the cold-start check-in locations. **SHM** performs better than **PSMM**, but still only achieve a low accuracy. They recommend a user's next location based on the observation of his friends' check-in history. The performance indicates that a user does not follow his friends' check-in sequence a lot on LBSNs, especially when performing a cold-start check-in.
- **CF** has comparable performance with **S.LF.UF**. Applying **S.LF.UF** to the whole test set is actually similar to user-based collaborative filtering, resulting in a close performance to **S.LF.UF** according to Table 3.8. This also demonstrates the practicability of our proposed correlation measures.
- **gSCorr** performs the best among all the approaches. To demonstrate the significance of its improvement over other baseline methods, we launch a random guess approach to recommend the cold-start check-ins. The recommendation accuracy of the random guess is always below 0.005%, indicating that **gSCorr** significantly improves the baseline methods.

Table 3.8: Performance Comparison for Location Recommendation

Dataset	Jul	Aug	Sep	Oct	Nov	Dec
S.LF.UF	18.31%	19.58%	19.71%	20.79%	21.10%	23.53%
PSMM	1.04%	1.19%	1.24%	1.22%	1.26%	1.23%
SHM	5.30%	5.08%	5.39%	5.65%	5.03%	5.58%
CF	18.24%	19.57%	19.45%	20.74%	20.84%	23.59%
gSCorr	19.21%	20.25%	20.36%	21.26%	21.42%	24.13%

3.3.5 Effect of Geo-Social Correlation Strengths and Measures

To further evaluate **gSCorr**, we consider the effect of both geo-social correlation strength and measures in capturing the user’s “new check-in” behavior. Therefore, we set up five baselines to compare the POI recommendation performance with **gSCorr**, as shown in Table 3.9. Each baseline adopts a different combination of correlation strength and measures, where “Es”, “Rs”, “Vs”, “Sm”, “Vm” represent “Equal Strength” (set all geo-social correlation strengths as 1), “Random Strength” (randomly assign the geo-social correlation strengths), “Various Strength” (the same as **gSCorr**), “Single Measure” (use *S.Lf.Uf* to measure the correlation probabilities for all the geo-social circles) and “Various Measures” (the same as **gSCorr**) respectively. Note that **gSCorr** is a various strength and various metrics approach. Following the evaluation metrics of recommendation system, we use top- k accuracy as evaluation metric and set $k = 1, 2, 3$ in the experiment. For each random strength approach (RsSm and RsVm), we run 30 times and report the average accuracy.

Table 3.10 shows the detailed recommendation performance of each method for further comparison. We summarize the essential observations below:

- The geo-social correlations from different geo-social circles contribute variously

Table 3.9: Evaluation Metrics

	Single Measure	Various Measures
Equal Strength	EsSm	EsVm
Random Strength	RsSm	RsVm
Various Strength	VsSm	gSCorr

to a user’s check-in behavior. Both **VsSm** and **gSCorr** perform better than their equal strength versions (i.e., **EsSm** and **EsVm**), respectively, indicating that the geo-social correlations are not equally weighted.

- The randomly assigned strength approaches (**RsSm** and **RsVm**) perform the worst compared to the other approaches, where the performance of **VsSm** has a 10.50% relative improvement over **RsSm**, and **gSCorr** has a 26.11% relative improvement over **RsVm**, indicating that social correlation strengths do affect check-in behavior.
- The single metric approaches (**EsSm**, **RsSm**, **VsSm**) always perform worse than the various metrics approaches (**EsVm**, **RsVm**, **gSCorr**), suggesting that for different social circles, there are different suitable correlation metrics.

gSCorr performs the best among all the approaches. To demonstrate the significance of its improvement over other methods, we launch a random guess approach to recommend the “cold-start” check-ins. The prediction accuracy of the random guess is always below 0.005% for top-1 prediction, and below 0.01% for top-2 and top-3 prediction, indicating that gSCorr significantly improves the baseline methods, suggesting the advantage of gSCorr as considering different geo-social correlation strength and metrics for each geo-social circle.

Table 3.10: POI Recommendation with Different Geo-Social Correlation Strengths and Measures

Methods	Top-1	Top-2	Top-3
EsVm	17.88%	24.06%	27.86%
EsSm	16.20%	21.92%	25.43%
VsSm	16.49%	22.28%	25.92%
RsSm	14.93%	20.30%	23.70%
RsVm	15.23%	20.85%	24.50%
gSCorr	19.21%	25.19%	28.69%

PERSONALIZED GEO-TEMPORAL POI RECOMMENDATION

In this section, we study the geo-temporal patterns for personalized POI recommendation on LBSNs. Temporal information is closely associated with the geographical check-ins. Figure 4.1 illustrates a user’s check-in history at various POIs with check-in time stamps. The temporal information embedded in these check-ins indicates two types of temporal patterns, temporal cyclic patterns and temporal chronological patterns. Firstly, the time stamps indicate the cyclic patterns (e.g., hour of the day, day of the week) of a user’s check-in behavior. Secondly, the time stamps work as an order indicator to connect check-in POIs chronologically for generating a user’s historical location trajectories. In the following sections, we investigate each pattern individually, and discuss their complementary effect for personalized POI recommendation.

4.1 Temporal Cyclic Patterns

As suggested in [83, 12, 51], human geographical movement exhibits significant temporal cyclic patterns on LBSNs and is highly relevant to the location property,

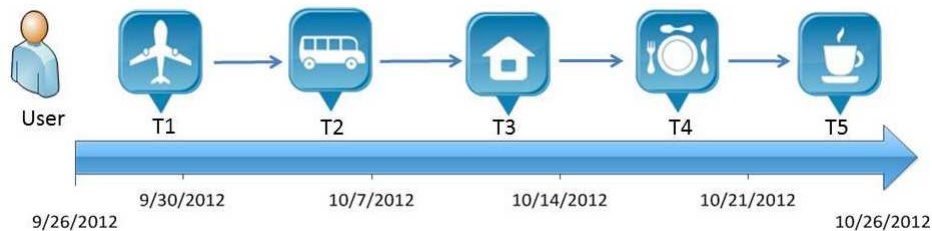


Figure 4.1: Geo-Temporal Patterns of Check-in Behavior

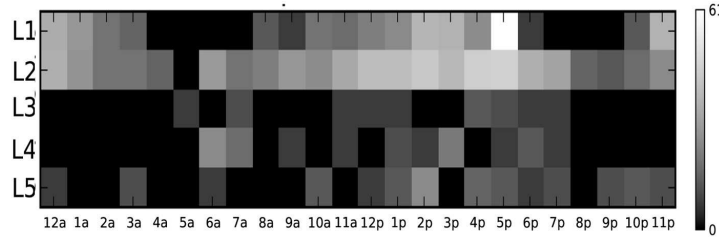


Figure 4.2: Daily Check-in Activities on LBSN

while the daily pattern (hours of the day) is one of the most fundamental patterns that reflects a user’s mobile behavior. For example, a user may regularly arrive to the office around 9:00 am, go to a restaurant for lunch at 12:00 pm, and watch movies at night around 10:00 pm. Therefore, investigating the features embedded in daily patterns enables us to better understand human mobile behavior, providing a potential opportunity to design more advanced POI recommender systems on LBSNs.

4.1.1 Temporal Non-uniformness and Consecutiveness

Previous work reports that a user’s preferences change continuously over time [74, 24], indicating two temporal properties of a user’s daily check-in preferences: (1) **non-uniformness**: a user exhibits distinct check-in preferences at different hours of the day; and (2) **consecutiveness**: a user tends to have more similar check-in preferences in consecutive hours than in non-consecutive hours. Figure 4.2 plots an illustrative example of a user’s aggregated check-in activities on his top 5 most visited POIs over 24 hours on our LBSN data. Each cell represents the total number of check-in activities happened at a specific POI during the corresponding hour, colored from black (least active) to white (most active). The user’s check-in behavior presents a different check-in POI distribution at each hour, which changes continually over time.

The temporal non-uniformness property, i.e., a user exhibits distinct check-in pref-

erences at different hours of the day, is straightforward to evaluate with a two-sided hypothesis testing on the check-in behavior of two temporal states for each user. Experiments show that this property does hold in the LBSN dataset. Due to the space limit, I will ignore its evaluation details and focus on evaluating the temporal consecutiveness. Firstly, define the check-in similarity of a user between two temporal states t_i and t_j as:

$$sim_u(t_i, t_j) = \frac{\mathbf{C}_{t_i}(u, :) \cdot \mathbf{C}_{t_j}(u, :)}{|\mathbf{C}_{t_i}(u, :)|_2 \times |\mathbf{C}_{t_j}(u, :)|_2}, \quad (4.1)$$

where $\mathbf{C}_t(u, :)$ is the check-in vector of user u at temporal state t . $|\bullet|_2$ is the 2-norm of a vector.

To evaluate the temporal consecutiveness, we calculate two similarities for each user u , i.e., consecutive similarity $\mathbf{S}_c(u)$ and non-consecutive similarity $\mathbf{S}_n(u)$. $\mathbf{S}_c(u)$ is the average similarity of all $sim_u(t_i, t_j)$ where t_i and t_j are consecutive temporal states. Note that T temporal states have in total T consecutive temporal similarities, i.e., $sim_u(t_1, t_2)$, $sim_u(t_2, t_3), \dots, sim_u(T - 1, T)$, and $sim_u(T, 1)$. Similarly, $\mathbf{S}_n(u)$ is the average similarity of all $sim_u(t_i, t_j)$ where t_i and t_j are non-consecutive temporal states. For fairly comparison, we randomly sample T non-consecutive temporal similarities $sim_u(t_i, t_j)$ to ensure that both $\mathbf{S}_c(u)$ and $\mathbf{S}_n(u)$ have the same sample size, and then take the average for calculating $\mathbf{S}_n(u)$.

We conduct a two-sample t-test on vectors \mathbf{S}_c and \mathbf{S}_n . The null hypothesis is H_0 : $\mathbf{S}_c \leq \mathbf{S}_n$, i.e., check-ins between consecutive temporal states are less or equally similar than that between non-consecutive temporal states, and the alternative hypothesis is H_1 : $\mathbf{S}_c > \mathbf{S}_n$. In our experiment, the null hypothesis is rejected at significant level $\alpha = 0.001$ with p-value of $5.6e-191$, i.e., a user's check-in in two consecutive temporal states have higher similarity than that in non-consecutive temporal states.

4.1.2 POI Recommendation with Geo-Temporal Patterns

To the best of our knowledge, these properties have not been exploited for POI recommendation on LBSNs. Furthermore, the large-scale check-in data on LBSNs is usually very sparse due to the user-driven check-in property [67, 55, 22]. To solve large-scale recommendation problems, matrix factorization is state-of-the-art technology that has been proven to be successful in the Netflix Competition [36], and is being used for item recommendation and trust prediction on product review sites like Epinions and Ciao for research purposes [73, 74]. Therefore, in this section, we leverage the temporal cyclic patterns on LBSNs with low-rank matrix factorization for POI recommendation.

A Basic POI Recommendation Model

We first introduce a basic POI recommendation model based on low-rank matrix factorization without considering temporal effects. Let $\mathbf{u} = \{u_1, u_2, \dots, u_m\}$ be the set of users, and $\mathbf{l} = \{l_1, l_2, \dots, l_n\}$ be the set of POIs, where m and n denote the number of users and POIs, respectively. $\mathbf{C} \in \mathbb{R}^{m \times n}$ is a user-POI matrix with each element \mathbf{C}_{ij} representing the number of check-ins made by user u_i at POI l_j . Let $\mathbf{U} \in \mathbb{R}^{m \times d}$ be the user check-in preferences and $\mathbf{L} \in \mathbb{R}^{n \times d}$ be the POI characteristics, with $d \ll \min(m, n)$ being the number of latent preference factors. The basic location recommendation model approximates u_i 's check-in preference on an unvisited l_j via solving the following optimization problem:

$$\min_{\mathbf{U}_i \geq 0, \mathbf{L}_j \geq 0} \sum_i^m \sum_j^n \mathbf{W}_{ij} (\mathbf{C}_{ij} - \mathbf{U}_i \mathbf{L}_j^\top)^2, \quad (4.2)$$

where $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a check-in indicator matrix, $\mathbf{W}_{ij} = 1$ indicating that u_i has checked in at l_j , $\mathbf{W}_{ij} = 0$ otherwise.

After obtaining \mathbf{U}_i and \mathbf{L}_j , the missing value in \mathbf{C} , represented as $\tilde{\mathbf{C}}_{ij}$, indicating the preference of a user u_i at an unvisited POI l_j , is then approximated by $\mathbf{U}_i\mathbf{L}_j^\top$. To avoid over-fitting, two smoothness regularizations are added on \mathbf{U}_i and \mathbf{L}_j respectively. Eq.(5.1) can then be represented in matrix format as

$$\min_{\mathbf{U} \geq 0, \mathbf{L} \geq 0} \|\mathbf{W} \odot (\mathbf{C} - \mathbf{U}\mathbf{L}^\top)\|_F^2 + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{L}\|_F^2, \quad (4.3)$$

where α and β are non-negative parameters to control the capability of \mathbf{U} and \mathbf{L} . \odot is the Hadamard product operator, where $(A \odot B)_{i,j} = A_{i,j} \times B_{i,j}$. $\|\cdot\|_F$ is the Frobenius norm of a matrix.

Modeling Temporal Non-uniformness

According to the temporal property of non-uniformness as described above, users exhibit distinct check-in preferences at different hours of the day. This inspires us to consider a user’s check-in behavior as a set of time-dependent check-in preferences, with each preference corresponding to an hour of the day. To model this property, we first introduce temporal state $t \in [1, T]$ to represent the hour of the day, where $T = 24$ is the total number of temporal states. For example, $t = 1$ for check-in time at “2012-10-24 00:30:00pm”, indicating the check-in happens during hour 0 to 1.

We then define $\mathbf{U}_t \in \mathbb{R}^{m \times d}$ as the time-dependent user check-in preferences under temporal state t . As observed in [83], POI characteristics are inherent properties that do not change much as time goes by. Therefore, we define POI characteristics to be time-independent, denoted as $\mathbf{L} \in \mathbb{R}^{n \times d}$. By approximating the check-in activities at each temporal state t and minimizing their aggregation, we obtain time-dependent user check-in preferences via the following optimization problem:

$$\min_{\mathbf{U}_t \geq 0, \mathbf{L} \geq 0} \sum_{t=1}^T \|\mathbf{W}_t \odot (\mathbf{C}_t - \mathbf{U}_t\mathbf{L}^\top)\|_F^2 + \alpha \sum_{t=1}^T \|\mathbf{U}_t\|_F^2 + \beta\|\mathbf{L}\|_F^2, \quad (4.4)$$

where $\mathbf{C}_t \in \mathbb{R}^{m \times n}$ contains the check-in activities at temporal state t , and \mathbf{W}_t is the corresponding indicator matrix.

Modeling Temporal Consecutiveness

Inspired by the temporal consecutiveness property, which implies that users on LBSNs tend to have closer check-in preferences on consecutive temporal state, we propose a temporal regularization to minimize the following terms:

$$\min \sum_{t=1}^T \sum_{i=1}^m \psi_i(t, t-1) \|\mathbf{U}_t(i, :) - \mathbf{U}_{t-1}(i, :)\|_2^2, \quad (4.5)$$

where $\psi_i(t, t-1) \in [0, 1]$ is defined as a temporal coefficient that measures the temporal closeness of u_i 's check-in preferences between temporal state t and $t-1$. The larger $\psi_i(t, t-1)$ is, the closer u_i 's check-in preferences between t and $t-1$. We use cosine similarity to measure $\psi_i(t, t-1)$, defined as

$$\psi_i(t, t-1) = \frac{\mathbf{C}_t(i, :) \cdot \mathbf{C}_{t-1}(i, :)}{\sqrt{\sum_j \mathbf{C}_t^2(i, j)} \sqrt{\sum_j \mathbf{C}_{t-1}^2(i, j)}}. \quad (4.6)$$

Note that we consider the temporal state $t-1$ as T when $t=1$, e.g., $U_{t-1} = U_T$ when $t=1$. After some derivations, we can get the matrix form of temporal regularization,

$$\sum_{t=1}^T \text{Tr}((\mathbf{U}_t - \mathbf{U}_{t-1})^\top \Sigma_t (\mathbf{U}_t - \mathbf{U}_{t-1})), \quad (4.7)$$

where Σ_t is the diagonal temporal coefficient matrix among m users, defined as

$$\Sigma_t = \begin{bmatrix} \psi_1(t, t-1) & 0 & \cdots & 0 \\ 0 & \psi_2(t, t-1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_m(t, t-1) \end{bmatrix}. \quad (4.8)$$

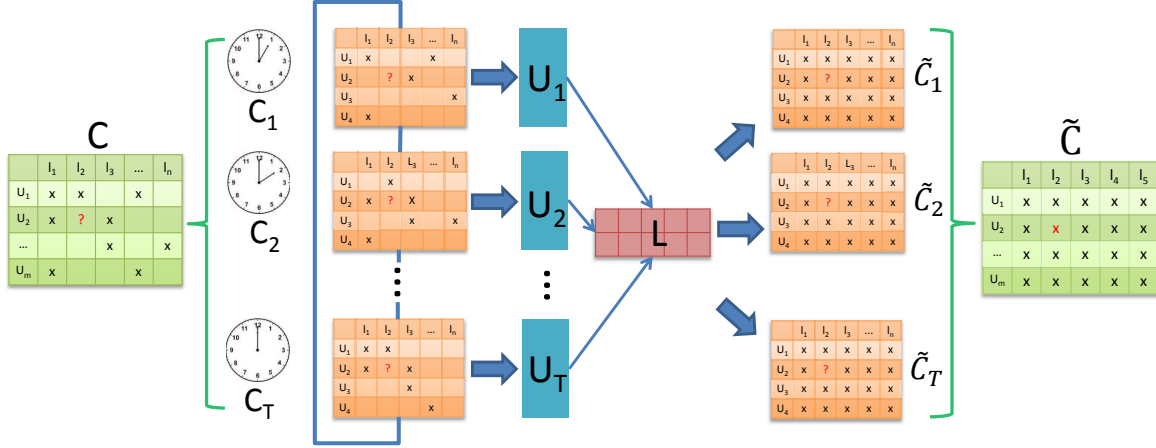


Figure 4.3: POI Recommendation Framework with Geo-Temporal Patterns

4.1.3 LRT: Location Recommendation with Temporal Effects

Figure 4.3 illustrates the working flow of our POI recommendation framework **LRT**. “ x ” denotes the observed check-in frequency by the user on the corresponding POI, and “?” represents the user’s check-in preferences on an unvisited POI that the framework is going to infer. The whole framework consists of three steps: temporal division, temporal factorization, and temporal aggregation. Firstly, the original user-POI matrix \mathbf{C} is divided into T sub-matrices according to the T temporal states, with each sub-matrix only containing check-in actions that happened at the corresponding temporal state. Secondly, each \mathbf{C}_t is factorized into the user check-in preference \mathbf{U}_t and the POI characteristics \mathbf{L} based on the above proposed model, while \mathbf{L} is shared by all of \mathbf{U}_t . Finally, the corresponding low-rank approximation $\tilde{\mathbf{C}}_t$ is constructed and aggregated (with ensemble method) into $\tilde{\mathbf{C}}$, representing the user check-in preferences of each POI. The POI recommendation will then be performed based on the final user check-in preference $\tilde{\mathbf{C}}(i, j)$.

Since the temporal division is straightforward to implement, in the following we

will describe in details the second and third steps, i.e., learning user temporal check-in preferences at each temporal state and aggregating temporal check-in preferences for location recommendation.

Learning Temporal Check-in Preferences

Based on the discussion of modeling temporal non-uniformness and consecutiveness properties in the above sections, the user temporal check-in preferences can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{U}_t \geq 0, \mathbf{L} \geq 0} & \sum_{t=1}^T \|\mathbf{W}_t \odot (\mathbf{C}_t - \mathbf{U}_t \mathbf{L}^\top)\|_F^2 + \alpha \sum_{t=1}^T \|\mathbf{U}_t\|_F^2 + \beta \|\mathbf{L}\|_F^2 \\ & + \lambda \sum_{t=1}^T \text{Tr}((\mathbf{U}_t - \mathbf{U}_{t-1})^\top \Sigma_t (\mathbf{U}_t - \mathbf{U}_{t-1})), \end{aligned} \quad (4.9)$$

where λ is a non-negative parameter to control the temporal regularization.

$$\begin{aligned} \mathcal{J} = & \sum_{t=1}^T \text{Tr}((\mathbf{W}_t^\top \odot \mathbf{C}_t^\top)(\mathbf{W}_t \odot \mathbf{C}_t) - (\mathbf{W}_t^\top \odot \mathbf{C}_t^\top)(\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top) \\ & - (\mathbf{W}_t \odot \mathbf{C}_t)(\mathbf{W}_t^\top \odot \mathbf{L} \mathbf{U}_t^\top) + (\mathbf{W}_t^\top \odot \mathbf{L} \mathbf{U}_t^\top)(\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top)) \\ & + \lambda \sum_{t=1}^T \text{Tr}((\mathbf{U}_t - \mathbf{U}_{t-1})^\top \Sigma_t (\mathbf{U}_t - \mathbf{U}_{t-1})) \\ & + \alpha \sum_{t=1}^T \text{Tr}(\mathbf{U}_t^\top \mathbf{U}_t) + \beta \text{Tr}(\mathbf{L}^\top \mathbf{L}) \\ & - \sum_{t=1}^T \text{Tr}(\Gamma_{\mathbf{U}_t} \mathbf{U}_t^\top) - \text{Tr}(\Gamma_{\mathbf{L}} \mathbf{L}^\top). \end{aligned} \quad (4.10)$$

where $\Gamma_{\mathbf{U}_t}$ and $\Gamma_{\mathbf{L}}$ are Lagrangian multipliers for non-negativity of \mathbf{U}_t and \mathbf{L} , respectively. By taking the derivation of \mathcal{J} with respect to \mathbf{U}_t and \mathbf{L} , we obtain

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{U}_t} &= -2(\mathbf{W}_t \odot \mathbf{C}_t)\mathbf{L} + 2(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + 2\lambda\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1}) \\ &\quad + 2\alpha\mathbf{U}_t - \Gamma_{\mathbf{U}_t}, \\ \frac{\partial \mathcal{J}}{\partial \mathbf{L}} &= -2\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t + 2\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t \\ &\quad + 2\beta\mathbf{L} - \Gamma_{\mathbf{L}}.\end{aligned}\tag{4.11}$$

Let $\frac{\partial \mathcal{J}}{\partial \mathbf{U}_t} = 0$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{L}} = 0$, we obtain

$$\begin{aligned}\Gamma_{\mathbf{U}_t} &= -2(\mathbf{W}_t \odot \mathbf{C}_t)\mathbf{L} + 2(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + 2\lambda\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1}) \\ &\quad + 2\alpha_t\mathbf{U}_t, \\ \Gamma_{\mathbf{V}} &= -2\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t + 2\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t + 2\beta\mathbf{V}.\end{aligned}\tag{4.12}$$

According to the Karush-Kuhn-Tucker condition,

$$\begin{aligned}\mathbf{U}_t(i, k)\Gamma_{\mathbf{U}_t}(i, k) &= 0, \forall i \in [1, m], k \in [1, d], t \in [1, T] \\ \mathbf{L}(i, k)\Gamma_{\mathbf{L}}(i, k) &= 0, \forall i \in [1, n], k \in [1, d].\end{aligned}\tag{4.13}$$

We obtain the following updating formula of \mathbf{U}_t and \mathbf{L} with a similar derivation process in [15]

$$\begin{aligned}\mathbf{U}_t(i, k) &\leftarrow \mathbf{U}_t(i, k)\sqrt{\frac{[(\mathbf{W}_t \odot \mathbf{C}_t)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_{t-1}](i, k)}{[(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_t + \alpha\mathbf{U}_t](i, k)}} \\ \mathbf{L}(i, k) &\leftarrow \mathbf{L}(i, k)\sqrt{\frac{[\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t](i, k)}{[\sum_{t=1}^T(\mathbf{W}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t + \beta\mathbf{L}](i, k)}}.\end{aligned}\tag{4.14}$$

Temporal Aggregation for POI Recommendation

By solving the above optimization problem, the user check-in preferences $\tilde{\mathbf{C}}_t(i, j)$ at each temporal state can be computed through $\mathbf{U}_t(i, :)\mathbf{L}(j, :)^{\top}$. To recommend locations

to a user w.r.t. each $\tilde{\mathbf{C}}_t(i, j)$, we define an aggregation function $f(\cdot)$ to compute the final user check-in preferences $\tilde{\mathbf{C}}(i, j)$.

$$\tilde{\mathbf{C}}(i, j) = f(\tilde{\mathbf{C}}_1(i, j), \tilde{\mathbf{C}}_2(i, j), \dots, \tilde{\mathbf{C}}_T(i, j)), \quad (4.15)$$

In this work, we propose four aggregation strategies for $f(\cdot)$, defined as below:

- **Sum:** we consider a user's check-in preferences on a location as the sum of his check-in preferences from each temporal state, i.e., $\tilde{\mathbf{C}}(i, j) = \sum_{t=1}^T \tilde{\mathbf{C}}_t(i, j)$.
- **Mean:** we consider a user's check-in preferences on a location as the average non-zero preferences from each temporal state, i.e., $\tilde{\mathbf{C}}(i, j) = \frac{\sum_{t=1}^T \tilde{\mathbf{C}}_t(i, j)}{|\{\mathbf{C}_t(i, j) | \mathbf{C}_t(i, j) \neq 0\}|}$.
- **Maximum:** we consider a user's check-in preferences on a location as his maximum temporal check-in preferences, i.e., $\tilde{\mathbf{C}}(i, j) = \max(\tilde{\mathbf{C}}_1(i, j), \dots, \tilde{\mathbf{C}}_T(i, j))$.
- **Voting:** Each $\tilde{\mathbf{C}}_t(i, j)$ acts as a recommender, and nominates top n locations to a user. The final recommended locations are those locations that have been nominated by most $\tilde{\mathbf{C}}_t(i, j)$.

The location recommendation will then be performed based on the final user check-in preference $\tilde{\mathbf{C}}(i, j)$.

Algorithm Analysis and Time Complexity

Algorithm 1 presents the detailed procedures of the proposed framework. Compared to the temporal division of C and temporal aggregation for $\tilde{\mathbf{C}}$, the updating rules for \mathbf{U}_t and L in each iteration corresponds to the major cost of Algorithm 1. Therefore, we next analysis the time complexity of updating \mathbf{U}_t and L . For the updating rule of \mathbf{U}_t , $(\mathbf{W}_t \odot \mathbf{C}_t)\mathbf{L}$ takes $O(md^2)$ operations due to the sparsity of \mathbf{W}_t and \mathbf{C}_t . Since Σ_t is a diagonal matrix, the time complexity of $\lambda \Sigma_t \mathbf{U}_{t-1}$ is $O(md)$. $(\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top)\mathbf{L}$

takes $O(mnd)$ operations, while the time complexity of $\lambda \Sigma_t \mathbf{U}_t$ and $\alpha \mathbf{U}_t$ is $O(md)$. Therefore, it takes $O(mndT)$ operations for updating all the \mathbf{U}_t . Similarly, the time complexity of $\sum_{t=1}^T (\mathbf{W}_t \odot \mathbf{C}_t)^\top \mathbf{U}_t$ for updating \mathbf{L} is $O(md^2T)$. $\sum_{t=1}^T (\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top)^\top \mathbf{U}_t$ takes $O(mndT)$ operations and $\beta \mathbf{L}$ takes $O(nd)$ operations, resulting in the time complexity of updating \mathbf{L} as $O(mndT)$. Since T is usually a constant of small value, in sum, the time complexity of Algorithm 1 is $O(mnd)$.

4.1.4 Experiments

In this section, we evaluate the performance of our framework **LRT** for location recommendation. In particular, we evaluate the following: (1) how the proposed framework fares in comparison with state-of-the-art models that capture static check-in preferences; (2) how the proposed framework recommends locations with various temporal aggregation strategies; and (3) whether other temporal patterns could be leveraged for location recommendation with the proposed framework. Before we delve into experiment details, we first discuss an LBSN dataset and evaluation metrics.

4.1.5 Dataset and Experiment Setup

We crawled the experimental dataset from Foursquare and obtained check-ins for three months (Jan 2011 - Mar 2011) to evaluate our proposed framework. We select check-in locations which have been visited by at least two distinct users, and users who have checked in at least 10 distinct locations. The statistics of the final dataset are shown in Table 4.1. The majority of check-ins happened in the United States.

We organize the dataset as a user-location matrix. The check-in density of the matrix is 8.84×10^{-4} . Logistic function $\frac{1}{1+(e^x)^{-1}}$ is commonly used in recommender system [50] to map each matrix element into $[0,1]$. We notice that in contrast with online item recommendation, where x (the rating of an item) is usually ranging from

Algorithm 1 Location Recommendation with Temporal Effects

Input: user-location check-in matrix \mathbf{C} , α , β , possible temporal states $\{1, 2, \dots, T\}$

Output: approximated user-location preference matrix $\tilde{\mathbf{C}}$

- 1: Divide \mathbf{C} into $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_T\}$ according to T
 - 2: Generate $\{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_T\}$ based on $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_T\}$
 - 3: Construct $\{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \dots, \boldsymbol{\Sigma}_T\}$ based on $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_T\}$
 - 4: Initialize $\{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_T\}$ and \mathbf{L} randomly
 - 5: **while** Not Convergent **do**
 - 6: **for** $t = 1$ to T **do**
 - 7: **for** $i = 1$ to m **do**
 - 8: **for** $k = 1$ to d **do**
 - 9: $\mathbf{U}_t(i, k) \leftarrow \mathbf{U}_t(i, k) \sqrt{\frac{[(\mathbf{W}_t \odot \mathbf{C}_t) \mathbf{L} + \lambda \boldsymbol{\Sigma}_t \mathbf{U}_{t-1}](i, k)}{[(\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top) \mathbf{L} + \lambda \boldsymbol{\Sigma}_t \mathbf{U}_t + \alpha \mathbf{U}_t](i, k)}}$
 - 10: **end for**
 - 11: **end for**
 - 12: **end for**
 - 13: **for** $i = 1$ to n **do**
 - 14: **for** $k = 1$ to d **do**
 - 15: $\mathbf{L}(i, k) \leftarrow \mathbf{L}(i, k) \sqrt{\frac{[\sum_{t=1}^T (\mathbf{W}_t \odot \mathbf{C}_t)^\top \mathbf{U}_t](i, k)}{[\sum_{t=1}^T (\mathbf{W}_t \odot \mathbf{U}_t \mathbf{L}^\top)^\top \mathbf{U}_t + \beta \mathbf{L}](i, k)}}$
 - 16: **end for**
 - 17: **end for**
 - 18: **end while**
 - 19: **for** $t = 1$ to T **do**
 - 20: Set $\tilde{\mathbf{C}}_t = \mathbf{U}_t \mathbf{L}^\top$
 - 21: **end for**
 - 22: Set $\tilde{\mathbf{C}} = f(\tilde{\mathbf{C}}_1, \tilde{\mathbf{C}}_2, \dots, \tilde{\mathbf{C}}_T)$
 - 23: **return** $\tilde{\mathbf{C}}$
-

Table 4.1: Statistical Information of the Dataset

duration	Jan 1, 2011-Mar 31, 2011
No. of users	5,269
No. of check-ins	288,079
No. of unique locations	26,381
Average check-ins per user	55
Check-in density	8.84×10^{-4}

1 to 5, in location recommendation, the value of x (check-in frequency of a location) is commonly large, while the function “ $(e^x)^{-1}$ ” would result in very small and indistinguishable values, with x being larger than 7. Therefore, we adjust the mapping function as $\frac{1}{1+x^{-1}}$, with x corresponding to $\tilde{\mathbf{C}}(i, j)$ in our data, which works better than the logistic function in our experiment.

For each individual user in the dataset, we randomly mark off 20% and 40% of all locations that he has checked-in for testing. The rest of the user-location pairs are used as training data to infer \mathbf{U}_t and \mathbf{L} for location recommendation. The random selection is conducted 5 times individually, and we report the average results.

To evaluate the recommendation performance, we are interested in: (1) how many previously marked off locations are recommended to the users among the total number of recommended locations, and (2) how many previously marked off locations are recommended to the users among the total number of marked off locations. Thus, we utilize *precision@N* and *recall@N* as our evaluation metrics, defined as follows:

$$precision@N = \frac{\sum_{u_i \in U} |TopN(u_i) \cap L(u_i)|}{\sum_{u_i \in U} |TopN(u_i)|} \quad (4.16)$$

$$recall@N = \frac{\sum_{u_i \in U} |TopN(u_i) \cap L(u_i)|}{\sum_{u_i \in U} |L(u_i)|}, \quad (4.17)$$

where $TopN(u_i)$ is the set of recommended locations to user u_i that u_i has not visited

in the training set. $L(u_i)$ is the set of locations that has been visited by u_i in the testing set. In our experiment, N is set to 5 and 10, respectively.

All the parameters in this work are set through cross-validation. For the proposed method, the experimental results use $d=10$ dimensions to represent the latent features, the regularization coefficients α and β are set to 2, and λ is set to 1. As suggested in [84], the effectiveness of recommender systems with sparse datasets (i.e., low-density user-item matrix) is usually not high. For example, the reported top 5 precision is 5% over a dataset with 8.02×10^{-3} density and 3.5% over a dataset with 4.24×10^{-5} density [84, 86]. Therefore, the low precision obtained in our experiment is reasonable. In this work, we focus on **comparing the relative performance of algorithms instead of their absolute performance.**

Comparison of Various Recommendation Models

Three baseline methods are introduced w.r.t. time-dependent and static check-in preferences, as defined below:

- **User-Based Collaborative Filtering (CF)**

User-based collaborative filtering is a state-of-the-art approach for recommender systems. We adopt the user-based recommender [101] for location recommendation. It computes a user’s interest in a location based on other users’ interests in that location. Temporal information is not considered in this approach.

- **Non-negative Matrix Factorization (NMF)**

Non-negative Matrix Factorization (NMF) [38] computes non-negative user check-in preferences under the whole user-location matrix, which is our basic location recommendation model, as defined in Eq. (4.3), without temporal effects.

- **Random LRT (R-LRT)**

We randomly divide the original user-location matrix \mathbf{C} into 24 pieces \mathbf{C}_t without considering the temporal state, and then apply the same recommendation process in Figure 4.3.

Figure 4.4 and Figure 4.5 show the comparison results of **LRT** with the proposed baseline methods [20]. The aggregation strategy is selected as voting due to its good performance. (more details on the comparison of aggregation strategies will be discussed in the next subsection). The results discover several observations which we summarize below:

- CF performs the worst among all the approaches. The data sparseness could be one reason to explain this performance. Due to the low density of the user-location matrix, the collaborative filtering approach fails to accurately recommend locations, and performs worse than matrix factorization approaches which leverage the low-rank approximation of user check-in preferences.
- Both NMF and R-LRT performance better than CF, indicating their ability in dealing with sparse data for location recommendation. Furthermore, the better performance of **LRT** than NMF suggests that time-dependent check-in preference capture user mobile behavior better than static check-in preferences.
- **LRT** performs better than R-LRT, suggesting that the division strategy is important to the recommendation effectiveness. Our model with the consideration of temporal effects is able to improve the location recommendation performance, while the matrix divide-aggregation strategy without a appropriate temporal division could result in a bad performance.

LRT performs the best among all the baseline methods. It considers time-dependent check-in preferences and outperforms approaches that capture static check-in preferences. The standard deviation of the performance from each method is less

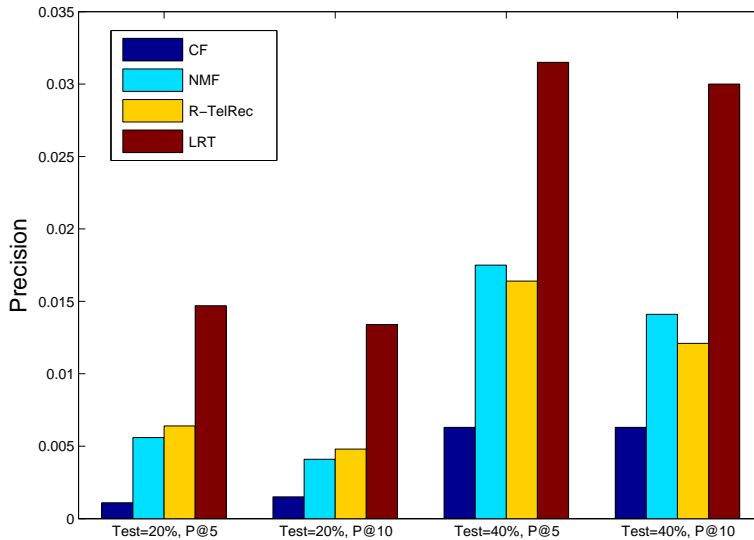


Figure 4.4: Recommendation Performance (Precision)

than 2×10^{-4} , confirming the reliability of our comparison results. As discussed before, due to the low matrix density, the low precision obtained in our experiment is reasonable. To further evaluate the significance of our framework, we launch a random recommendation [85]. For each user, we randomly select 5/10 locations from the total 26,381 locations (excluding those locations that have been previously visited by the user), and recommend them to the user. The recommendation performance is shown in Table 4.2. Compared to the random recommendation, our proposed framework is, on average, 73.27 times better than the random performance, demonstrating the power of temporal effects for improving location recommendation performance.

Location Recommendation with Various Aggregation Strategies

In this subsection, we discuss the performance of various aggregation strategies. We compare the recommendation performance with four aggregation strategies, and list

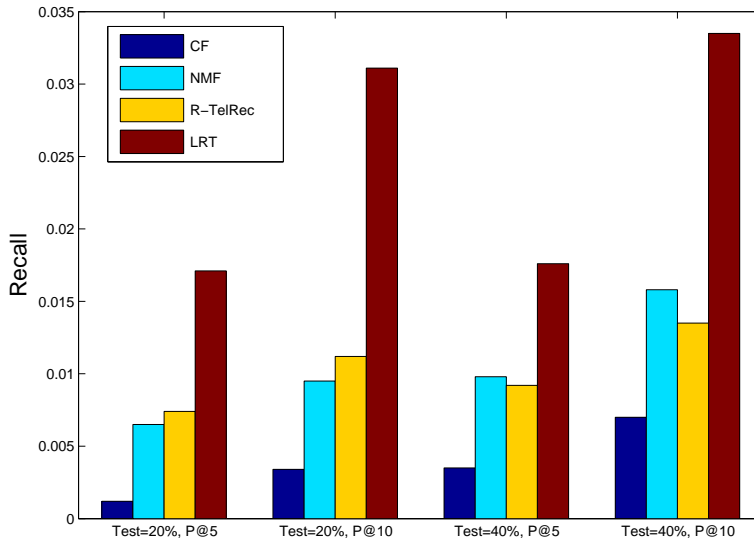


Figure 4.5: Recommendation Performance (Recall)

Table 4.2: Performance of Random Recommendation

Testing	Metrics	@5	@10
20%	Precision	0.0152%	0.0190%
	Recall	0.0177%	0.0442%
40%	Precision	0.0266%	0.0361%
	Recall	0.0149%	0.0403%

the results in Table 4.3 and Table 4.4. We summarize the essential observations below:

- The **mean** performs the worst among all the aggregation strategies. This is because taking average on all the temporal preferences degrades the preference variance, and makes the personal preferences indistinguishable. It validates that a user’s check-in preferences are highly dependent on the temporal state, approaches regardless of this may fail in recommending the right locations.

Table 4.3: Comparison of Aggregation Strategies (Precision)

Testing	Metrics	Sum	Mean	Max	Voting
20%	P@5	1.37%	0.03%	1.35%	1.47%
	P@10	1.31%	0.03%	1.30%	1.34%
40%	P@5	3.08%	0.46%	3.10%	3.20%
	P@10	2.95%	0.44%	2.95%	3.00%

Table 4.4: Comparison of Aggregation Strategies (Recall)

Testing	Metrics	Sum	Mean	Max	Voting
20%	R@5	1.60%	0.03%	1.57%	1.71%
	R@10	3.05%	0.08%	3.03%	3.11%
40%	R@5	1.73%	0.03%	1.74%	1.79%
	R@10	3.25%	0.05%	3.30%	3.35%

- The **maximum** has similar performance to the **sum**, suggesting that if a user’s check-in preferences are strongly indicated by one temporal state, then with high probability it indicates the true preferences of the user. This is also consistent with the observations in [24] that a user’s check-in behavior presents gaussian distribution over hours of the day, in which a user mostly checks-in at a location during a specific period of time, and rarely visits it during other time periods.
- The **voting** performs the best among all the aggregation strategies. Compared to the **sum**, it filters controversial location candidates at each temporal state, and reduces the uncertainty brought by the noisy location candidates, demonstrating its robustness in dealing with noisy data.

Exploring Various Temporal Patterns

LRT is designed to recommend locations to a user by taking advantage of temporal patterns. So far we have evaluated its recommendation performance with daily patterns, while its recommendation ability is not limited to one specific temporal pattern. By taking different definitions of temporal state, many other temporal patterns can be utilized for location recommendation with **LRT**, as long as they contain the non-uniformness and consecutiveness properties. For example, we could define the temporal state as $t=[1,T]$, with $T=7$ for weekly (day of the week) patterns, $T=2$ for weekday/weekend patterns, and $T=12$ for monthly (month of the year) patterns. The only change is to divide the original user-location matrix \mathbf{C} into a set of \mathbf{C}_t according to the corresponding temporal state. Table 4.5 shows the recommendation results of **LRT** with weekly patterns and weekday/weekend patterns. Due to the space limit, we only present the results on testing size = 40%. The results indicate that weekly patterns and weekday/weekend patterns can also capture users' temporal check-in preferences, and improve the location recommendation performance.

Table 4.5: Comparison of Temporal Patterns

Temporal Patterns	Metrics	@5	@10
Day of the Week	Precision	2.32%	2.18%
	Recall	1.30%	2.45%
Weekday/Weekend	Precision	2.23%	2.04%
	Recall	1.21%	2.28%

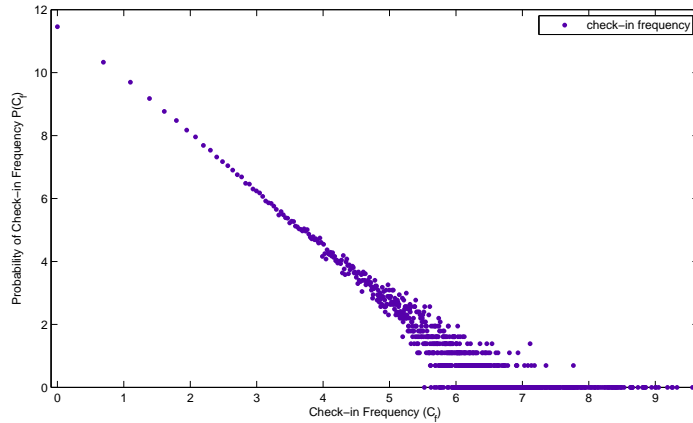


Figure 4.6: Power-law Distribution of Check-ins from All the Users

4.2 Temporal Chronological Patterns

The chronological information connects a user’s check-in history into a POI trajectory, presenting two properties on LBSNs: (1) a user’s check-in history approximately follows a power-law distribution, i.e., a user goes to a few places many times and to many places a few times. Figure 4.6 shows the distribution of check-in frequency (in log scale) in our dataset. The figure suggests that the check-in history follows a power-law distribution and the corresponding exponent is approximately 1.42. The check-in distribution of an individual also shows the power-law property, as shown in Figure 4.7; and (2) chronological patterns have short-term effect. As illustrated in Figure 4.1, a user arrives at the airport and then takes a shuttle to the hotel. After his dinner, he sips a cup of coffee. The previous check-ins at the airport, shuttle stop, hotel and restaurant have different tie strengths with respect to the latest check-in at the coffee shop. Furthermore, the tie strength decreases over time.

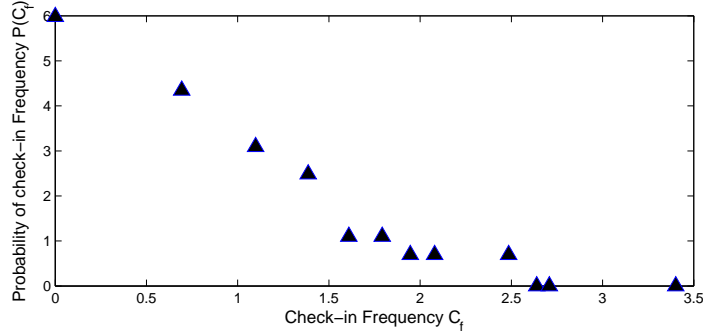


Figure 4.7: Power-law Distribution of Individual Check-ins

4.2.1 Modeling Power-Law Distribution and Short-Term Effect

Capturing the long tail of power-law distribution is a challenging task [4]. In addition, the determination of tie strength between the current check-in and various previous check-ins relies on the considering of variation of check-in time. To address these challenges, we introduce a language model to help in analyzing temporal chronological patterns, specifically, the power-law distribution and short-term effect.

Language Modeling and LBSN Mining

There are many common features shared between language processing and LBSNs mining. First, the text data and check-in data have similar structures, as shown in Table 4.6. For example, a document in language processing can correspond to a individual check-in sequence in LBSNs, while a word in the sentence corresponds to a check-in POI. Second, the power-law distribution and short term effect observed in LBSNs have also been found in natural language processing, where the word distribution is closely approximated by power-law [103]; and the current word is more relevant to its adjacent words than distant ones. Thus, language models for language processing is potentially applicable to LBSNs due to these common features.

Table 4.6: Corresponding Features between Language and LBSN Modeling

Language Modeling		LBSN Modeling	
Corpus		Check-in collection	
Document		Individual check-ins	
Document Structure	Paragraph	Check-in Structure	Monthly check-in sequence
	Sentence		Weekly check-in sequence
	Phrase		Daily check-in sequence
	Word		Check-in location

Pitman-Yor process [63, 62, 33] is a state-of-the-art language model that generates a power-law distribution of word tokens [28]. Furthermore, its hierarchical extension, i.e., Hierarchical Pitman-Yor (HPY) process [76, 77], assumes that the earliest word has least importance to the latest word, which has potential to be leveraged to capture the short-term effect in LBSNs. Therefore, we propose to utilize the power of language model in LBSNs for modeling check-in behavior.

Modeling Power-Law Distribution

Firstly, we introduce how to use PY process to capture the power-law property. The PY process generates a distribution over distributions over a probability space. Given a user with his/her check-in history, the next check-in location distribution is formulated as:

$$G \sim PY(d, \gamma, G_0), \tag{4.18}$$

where G is the next check-in location distribution based on the observed check-in history, $d \in [0, 1)$ is a discount parameter to control the power-law property, γ is a strength parameter, and G_0 is a base distribution over the location space. Let \mathcal{L} be

the location space which is a fixed and finite vocabulary of m locations, i.e., $m = |\mathcal{L}|$. The base distribution G_0 is a uniform distribution providing a prior probability of the location before observing any data. It satisfies $G_0(l) = 1/m$, where $G_0(l)$ is the probability of location $l \in \mathcal{L}$ being checked-in. Furthermore, when the discount parameter d is regarded as zero, this process reduces to the Dirichlet process [18].

Next, we illustrate how to generate a check-in sequence with this process. Let c_1, c_2, \dots, c_n be a sequence of check-ins coming one by one. The first arrived check-in chooses a location drawn from the distribution G_0 , then uses this location to form a location node and adheres to it. The subsequent check-in could either choose to adhere to a previous location node as its check-in location, or choose a new location node with its check-in location drawn from G_0 . The choosing rule is: the k -th location node with probability $\frac{N_k - d}{\gamma + n}$ while a new location node with probability $\frac{\gamma + td}{\gamma + n}$, where N_k denotes the number of check-ins adhered to location node k , $n = \sum_k N_k$ the length of check-in sequence, and t the current number of location nodes. Notice that each location node represents a check-in location. Since a new draw from G_0 may generate a previously appeared location, there may be multiple location nodes corresponding to one check-in location. Therefore, by marginalizing on the location node, the predictive probability of a new check-in c_{n+1} at location l given the previous check-in sequence is,

$$\begin{aligned} & P(c_{n+1} = l | c_1, c_2, \dots, c_n) \\ &= \sum_k \frac{N_k - d}{\gamma + n} \delta_k^l + \frac{\gamma + td}{\gamma + n} G_0 = \frac{N_l - t_l d}{\gamma + n} + \frac{\gamma + td}{\gamma + n} G_0, \end{aligned}$$

where δ_k^l is a function that satisfies:

$$\delta_k^l = \begin{cases} 1 & \text{location node } k \text{ represents location } l \\ 0 & \text{location node } k \text{ does not represent location } l, \end{cases}$$

$N_l = \sum_k N_k \delta_k^l$ denotes the current number of check-ins adhered to the location node at location l , which is the current number of check-ins at location l , and t_l is the current number of location nodes that represent location l . This generating process indicates that a new check-in would either choose a previous appeared location l with probability proportional to $N_l - t_l d$, or choose a location drawn from G_0 with probability proportional to $\gamma + td$.

Figure 4.8 illustrates a generating process of the next check-in c_{11} with 10 previous check-ins $\{c_1 = l_1, c_2 = l_1, c_3 = l_2, c_4 = l_3, c_5 = l_1, c_6 = l_3, c_7 = l_4, c_8 = l_1, c_9 = l_3, c_{10} = l_4\}$. The green nodes are location nodes and each one represents a location corresponding to a location icon. Red nodes are check-ins that adhered to the location nodes, which indicates the check-ins happened at that location. The probability of next check-in c_{11} at location l_2 consists of three parts: (1) c_{11} adheres to the location node L_2 with probability $\frac{1-d}{\gamma+10}$; (2) c_{11} adheres to the location node L_4 with probability $\frac{2-d}{\gamma+10}$; and (3) c_{11} forms a new location node representing the check-in location l_2 with probability $\frac{\gamma+4d}{\gamma+10}G_0(l_2)$. Therefore the probability of the next check-in $c_{11} = l_2$ is:

$$P(c_{11} = l_2 | c_1, \dots, c_{10}) = \frac{3 - 2d}{\gamma + 10} + \frac{\gamma + 4d}{\gamma + 10}G_0(l_2), \quad (4.19)$$

This generating process shows two properties: (1) the rich-get-richer property indicates a user would like to visit a previously visited location; and (2) the more check-ins occurred, the more new locations would appear as drawn from the base distribution G_0 .

Modeling Short-Term Effect

The PY process models the power-law property and generates the unigram check-in distribution for a check-in sequence. However, a unigram distribution is not sufficient

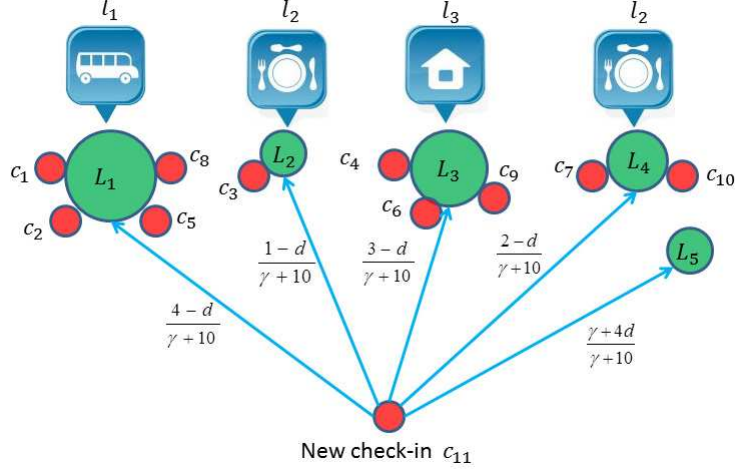


Figure 4.8: The Generating Process of Check-in Sequence

to capture the short-term effect, therefore we adopt the hierarchical extension of PY process, i.e. Hierarchical Pitman Yor process [76, 77] to consider the historical context of a particular check-in. It is an n-gram model that naturally captures the short-term effect while keeping the power-law property in distribution. It models the probability of the next check-in, denoted as G_u , given a history context u as:

$$G_u \sim PY(d_{|u|}, \gamma_{|u|}, G_{\pi(u)}), \quad (4.20)$$

where $G_u(l), l \in \mathcal{L}$, is the probability of the next check-in occurring at location l given the history context u . The discount parameter $d_{|u|}$ and strength parameter $\gamma_{|u|}$ are functions of the historical context u . $\pi(u)$ is the suffix of u consisting of all but the earliest check-in, therefore $G_{\pi(u)}$ is the probability of next check-in given all but the earliest check-in in the history context u . $G_{\pi(u)}$ is then computed with the parameter $d_{|\pi(u)|}, \gamma_{|\pi(u)|}$ and $G_{\pi(\pi(u))}$. This process is repeated until we get the empty historical context \emptyset ,

$$G_{\emptyset} \sim PY(d_0, \gamma_0, G_0). \quad (4.21)$$

Note that this iterative process drops the earliest check-in first in each iteration. It assumes that the earliest check-in would have the least importance in determining the distribution over the next check-ins, which in turn captures the short-term effect.

4.2.2 HM: Historical Model

Regarding the modeling of power-law distribution and short-term effect of a user's check sequence as discussed above, we propose a historical model to capture the user's check-in behavior in terms of historical ties.

Based on Eq. (4.20), the predictive probability of the next check-in c_{n+1} at location l with context u is defined as:

$$\begin{aligned} & P_u^{HPY}(c_{n+1} = l | c_1, c_2, \dots, c_n) \\ &= \frac{N_{ul} - t_{ul}d_{|u|}}{\gamma_{|u|} + n_u} + \frac{\gamma_{|u|} + t_{ul}d_{|u|}}{\gamma_{|u|} + n_u} G_{\pi(u)}(c_{n+1} = l | c_1, c_2, \dots, c_n), \end{aligned} \quad (4.22)$$

where N_{ul} is the number of check-ins at l following the history context u and $n_u = \sum_{l'} N_{ul'}$. $t_u = \sum_l t_{ul}$ is the sum of all t_{ul} , which is a latent variable satisfying:

$$\begin{cases} t_{ul} = 0 & \text{if } N_{ul} = 0; \\ 0 \leq t_{ul} \leq N_{ul} & \text{if } N_{ul} > 0; \end{cases}$$

Since we always consider a user's complete check-in history as historical context u , we remove the notion u in the following sections. To model the historical tie effect, we define our *historical model* as:

$$P_H^i(c_{n+1} = l) = P_{HPY}^{i,i}(c_{n+1} = l). \quad (4.23)$$

where $P_{HPY}^{i,i}(c_{n+1} = l)$ is the probability of u_i 's check-in c_{n+1} at location l generated by HPY process with user u_i 's observed check-in history.

Table 4.7: Average Number of Check-ins between Two Users

	Common check-ins
between friends	11.8306
between strangers	4.3226

4.2.3 SHM: Social-Historical Model

Social correlation [3] suggests to consider users’ social ties since human movement is usually affected by their social context, such as visiting friends, going out with colleagues, traveling while following friends’ recommendations, and so on. The historical ties and social ties, can shape a user’s check-in experience on LBSNs, while each tie gives rise to a different probability of check-in activity, which indicates that people in different spatial-temporal-social circles have different interactions. Thus, exploring a user’s social-historical ties is crucial to analyze his check-in behavior and therefore understand the corresponding movement.

We explore the social tie effect by proposing a social-historical model to understand the user’s check-in behavior on LBSNs. First, we investigate the social correlation of check-in behavior, more specifically, we ask whether the friendships a user has affect his check-in behavior.

We first compare the number of common check-ins between two friends and two strangers. As shown in Table 4.7, on average, a pair of strangers share approximately 4.32 check-ins, while a pair of friends share approximately 11.83 check-ins, which is as almost 3 times large as the former.

Next, we define the check-in similarity between two users and compare the similarity between users with friendship and those without. For each user, let $\mathbf{f} \in \mathbb{R}^m$ be his check-in vector with each element $\mathbf{f}(k)$ equal to the number of check-ins at

location $l_k \in \mathcal{L}$, where $m = |\mathcal{L}|$ is the vocabulary size. The cosine similarity of two users u_i and u_j is defined as:

$$\text{sim}(u_i, u_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{|\mathbf{f}_i|_2 \times |\mathbf{f}_j|_2}, \quad (4.24)$$

where $|\bullet|_2$ is the 2-norm of a vector.

We define the similarity between u_i and a group G of other users as the average similarity between user u_j and the users in group G ,

$$S_G(u_i) = \frac{\sum_{u_j \in G} \text{sim}(u_i, u_j)}{|G|}. \quad (4.25)$$

For each u_i , we calculate two similarities, i.e., $\mathbf{S}_F(u_i)$ is the average similarity of u_i and his friendship network; $\mathbf{S}_R(u_i)$ is the average similarity of u_i and randomly chosen users, who are not in the friendship network of u_i . The number of the randomly chosen users is the same as that of u_i 's friends.

We conduct a two-sample t-test on the vectors \mathbf{S}_F and \mathbf{S}_R . The null hypothesis is $H_0: \mathbf{S}_F \leq \mathbf{S}_R$, i.e., users with friendship share less common check-ins than those without, and the alternative hypothesis is $H_1: \mathbf{S}_F > \mathbf{S}_R$. In our experiment, the null hypothesis is rejected at significant level $\alpha = 0.001$ with p-value of $2.6e-6$, i.e., users with friendship have higher check-in similarity than those without.

The evidence from both shared check-ins and t-test suggests that with high probability, users with friend relationships have larger check-in correlation than those without, which demonstrates that a user's social ties contain important evidence for the user's movement. In this work, we propose an effective model to integrate both effects, in order to explore the social-historical ties. To do so, we add a user's social ties as a regularization part to his historical ties. A parameter $\eta \in [0, 1]$ is introduced to control the weight between historical and social ties. For a user u_i , the probability

of the next check-in location is defined as:

$$P_{SH}^i(c_{n+1} = l) = \eta P_H^i(c_{n+1} = l) + (1 - \eta) P_S^i(c_{n+1} = l). \quad (4.26)$$

We denote this model as *social-historical model*, where $P_H^i(c_{n+1} = l)$ is the probability of u_i 's check-in at location l observed from his historical ties, defined in Eq. (4.23); $P_S^i(c_{n+1} = l)$ is the check-in probability based on u_i 's social ties, defined as:

$$P_S^i(c_{n+1} = l) = \sum_{u_j \in \mathcal{N}(u_i)} sim(u_i, u_j) P_{HPY}^{i,j}(c_{n+1} = l). \quad (4.27)$$

where $\mathcal{N}(u_i)$ is the set of u_i 's friends. $P_{HPY}^{i,j}(c_{n+1} = l)$ is the probability of u_i 's next check-in at location l computed by HPY process with u_j 's check-in history as training data. Note that only the check-ins before the prediction time are included in the training data.

4.2.4 Experiments

We evaluate our proposed models: historical model and social-historical model through the following aspects: (1) how the proposed historical model fares in comparison with baseline models; (2) how the proposed historical model behaves over time; (3) whether social ties help location prediction as we discussed earlier; and (4) under what circumstances, the two types of ties complement each other.

4.2.5 Dataset and Experiment Setup

We use Foursquare dataset to study the social-historical ties on LBSNs. For a particular user on Foursquare, we get his check-in history with timestamps ranging from August, 2010 to November, 2011. We also collect the user's friendship information. In our experiment, we consider the users who have at least 10 check-ins. We obtain 43,108 unique geographical locations as the location vocabulary. Some key statistics of the dataset are shown in Table 4.8.

Table 4.8: Statistical Information of the Dataset

number of users	18,107
number of check-ins	2,073,740
number of links	123,325
average check-ins per user	101
clustering coefficient	0.1841
average degree	10.58

We separate the check-in sequence of each user into 9 time bins, and each time bin has approximately equal time interval. Let the timestamp at the end of each time bin be $\mathcal{T} = \{T_1, T_2, \dots, T_9\}$. We predict the check-in location at each timestamp for the user, with his historical check-ins before that time as observed context. Denote the prediction for user u at time t as $P_t(u)$, the prediction accuracy is defined as:

$$accuracy(T_i) = \frac{|\{u|u \in \mathcal{U}, P_{T_i}(u) = l_{T_i}(u)\}|}{|\mathcal{U}|}. \quad (4.28)$$

where \mathcal{U} is the set of users, $l_{T_i}(u)$ denotes the actual check-in location l of user u at time T_i .

Baseline Models

To evaluate the historical model (HM) and social-historical model (SHM), we choose three baseline models, i.e., Most Frequent Check-in model (MFC), Most Frequent Time model (MFT), and Order- k Markov Model based on our review of related work (to discuss later). The MFC baseline model considers the power-law property simply in aspect of rich-get-richer effect. The MFT model considers the temporal pattern only, which was used in [13] for comparison with their periodic model. Since our proposed models do not attempt to model periodic behavior, we focus on the social

and historical sequence of check-ins. Integrating periodic patterns in HM and HPY will be an extension of this work. The Order- k Markov Model considers the short-term effect of historical check-ins, which is reported as a state-of-the-art prediction algorithm for location prediction [69]. We give detailed information of these three baselines below:

- **Most Frequent Check-in Model:** In [9], a logistic regression model was proposed and found that the strongest predictor is the check-in frequency of the historical check-ins made by the user. In this work, we use this rule as one baseline, denoted as the most frequent check-in model (MFC). It assigns the probability of next check-in c_{n+1} at location l as the probability of l appearing in the check-in history,

$$P_{MFC}(c_{n+1} = l | \mathcal{C}) = \frac{|\{c_r | c_r \in \mathcal{C}, c_r = l\}|}{|\{c_r | c_r \in \mathcal{C}\}|}, \quad (4.29)$$

where $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ is the set of check-in history.

- **Most Frequent Time Model:** People tend to go the the same place at the similar time of the day as a routine activity. For example, an individual might like to have coffee after lunch; therefore, it would be common for him to check-in at Starbucks around 1pm. We choose the most frequent time model (MFT) as another baseline considering the temporal patterns of the check-ins. Let $t_{n+1} = h$ denote that the time at the $(n + 1)$ -th check-in is h , where $h \in \mathcal{H} = \{1, 2, \dots, 24\}$ is a discrete set of 24 hours. MFT model assigns the probability of next check-in c_{n+1} at location l at time h as the probability of the location l occurring at time h in the previous check-in history,

$$\begin{aligned} & P_{MFT}(c_{n+1} = l | \mathcal{C}, t_{n+1} = h) \\ &= \frac{|c | c_r \in \mathcal{C}, c_r = l, t_r = h|}{|c_r | c_r \in \mathcal{C}, t_r = h|}. \end{aligned} \quad (4.30)$$

- **Order- k Markov Model:** The third baseline is the order- k Markov Model. It considers the latest k check-in context, and searches for frequent patterns to predict the next location. The probability of the next check-in c_{n+1} at location l with order- k Markov model is defined as:

$$\begin{aligned}
P_{order-k}(c_{n+1} = l|\mathcal{C}) &= P(c_{n+1} = l|c_{n-k+1}, \dots, c_n) \\
&= \frac{|c_r|_{c_r \in \mathcal{C}, c_r = l, c_{r-j} = c_{n-j+1}}}{|c_r|_{c_r \in \mathcal{C}, c_{r-j} = c_{n-j+1}}}, 0 < j < k, j \in \mathbb{Z}.
\end{aligned} \tag{4.31}$$

We consider the Order-1 and Order-2 Markov models as baseline methods, note that the MFC is actually Order-0 Markov model.

Results and Discussions

Figure 4.9 shows the comparison results at 9 time stamps. The Order-2 Markov model performs the worst while SHM obtains the best performance for all the 9 time stamps. MFC model performs well but its accuracy decreases greatly with time. The Order-1 Markov model has better performance than MFC after T6, while the MFT model performs stable without impressive accuracy. To further investigate their performance, we summarize several interesting observations below:

- The MFC performs better than MFT, Order-1, and Order-2 Markov models before T6. Since it predicts the next location as the most frequent one in history, it considers the rich-get-richer property of power-law effect. However, it ignores the short-term effect; therefore, as time goes by, it can not distinguish which check-ins are more important in the long history and its accuracy decreases quickly. In contrast, the Order-1 Markov model outperforms MFC after T6. That is because Order-1 Markov model considers the short-term effect more than power-law property, it is not affected by the length of the history as much as the MFC.

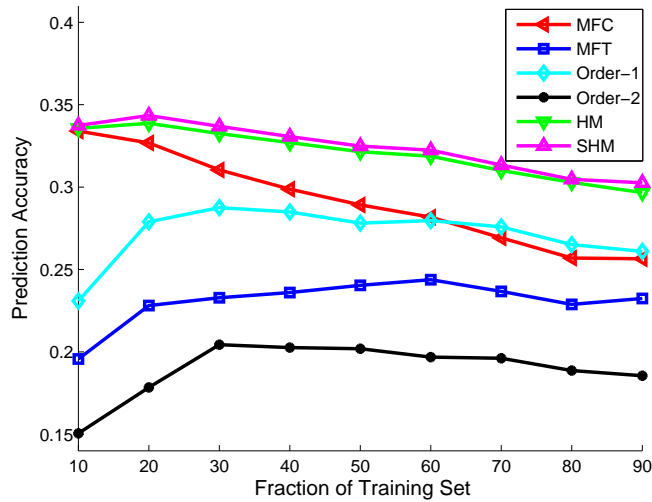


Figure 4.9: Performance Comparison of Recommendation Models

- All the models have a decreasing trend in accuracy after certain time point, especially the SHM, HM, Order-1, and Order-2 Markov models have similar decreasing rate with time. This phenomenon can be explained by the increasing number of appeared unique check-ins. We prove this by launching a random guess of next check-in location at each time point. We denote the average number of unique check-ins per user that appeared before time t as W_t . The probability of accurately predict the next location at this time by random guess is the inverse of W_t , which reflects the difficulty of prediction. We denote this as the random guess accuracy A_{W_t} . The statistics information of W_t and its corresponding A_{W_t} from our data is shown in Table 4.9. From T1 to T9, the accuracy of random guess keeps decreasing from 20.49% to 4.33% (approximately 78.87% relative decrement), while our historical model only decreases 3.9% from 39.56% to 29.66% (approximately 11.62% relative decrement). In sum, the performance of our historical model is considerable, and even slight improvement

in experiment is significant considering the difficulty of this prediction task.

- The MFC has the highest decreasing rate among all the models. This phenomenon is caused by two factors. Firstly, MFC is affected by the number of appeared unique check-ins as well as SHM, HM, Order-1, and Order-2 Markov models as described above. Secondly, it suffers from the short-term effect. Since even the number of appeared unique check-ins does not increase, it can not distinguish the most important check-ins to current time through the long history. Therefore, suffering from both unique check-ins and short-term effect, it has the greatest decreasing rate among all the five models.
- In our data, there are 14.47% of users with check-in sequence length between 10 to 20. For these users at time T1, only 1 to 2 check-ins are observed, which significantly intensifies the prediction difficulty. Specially, SHM, HM and MFC are very close to each other at T1, because all are suffering from the lack of observed data. The MFT, Order-1, and Order-2 Markov models perform even worse than SHM, HM, and MFC due to their strict pattern rules. With insufficient data, few patterns can be found and used to determine the next location by these three models; while as time goes by, more and more data are observed which improves their performance. This suggests that SHM, HM and MFC are more robust to the situation when the observation sequence is insufficient. The Order-2 Markov model is too strict on its pattern rule therefore it performs the worst due to over-fitting.
- The HM obtains better performance than all baselines, which considers both power-law property and short-term effect. Furthermore, the smoothing strategy on the n-gram context gives it better performance than the Order-2 Markov model, which suffers severely from over-fitting. The MFT performs stable,

which suggests the importance of temporal information. We will further investigate the temporal effect on check-in behavior in our future work.

We note that SHM consistently outperforms HM, and SHM considers both historical and social ties. To investigate the contribution of social ties and historical ties in affecting user’s behavior, we increase the parameter η from 0 to 1 with an increment step of 0.01 and observe the prediction performance at each η . We only show the prediction accuracy at times T3, T6 and T9 in Figure 4.10, Figure 4.11, and Figure 4.12, since similar performance can be observed at other time points. Some interesting insights can be observed:

- When $\eta = 0$, the social-historical model only considers social ties. Its performance is always worst, suggesting that considering social information only is not enough to capture the check-in behavior.
- By increasing η , the performance shows the following pattern: first increasing, reaching its peak value and then decreasing. Most of the time, the best performance is achieved at around $\eta = 0.7$. A big weight is given to historical ties, indicating that historical ties are more important than social ties.
- When $\eta = 1$, the social-historical model boils down to the historical model. Its performance is not the best, suggesting that social ties are also important.
- Comparing with the previous time, the social ties make the greatest improvement on performance of historical ties at T9, indicating that social ties are complementary to the historical ties, especially when the historical model does not perform well due to the long and noisy history.

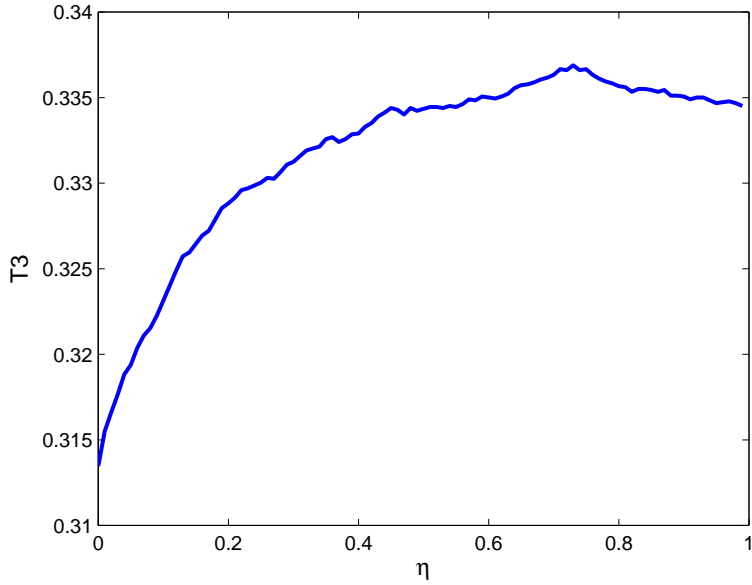


Figure 4.10: The Performance of Social-historical Model w.r.t. η ($T3$)

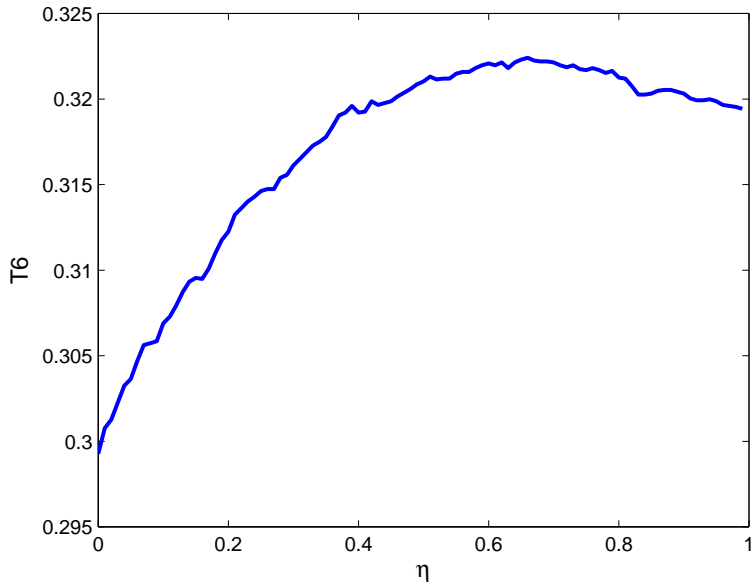


Figure 4.11: The Performance of Social-historical Model w.r.t. η ($T6$)

Table 4.9: Number of Unique Check-ins at Each Time Point

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
W_t	4.88	7.578	9.95	12.20	14.40	16.59	18.75	20.92	23.11
A_{W_t}	20.49%	13.20%	10.05%	8.20%	6.94%	6.03%	5.33%	4.78%	4.33%

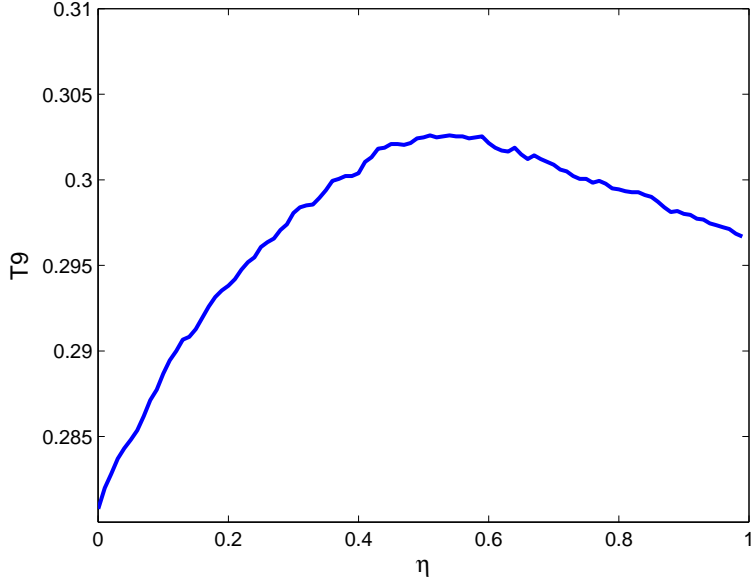


Figure 4.12: The Performance of Social-historical Model w.r.t. η (T9)

4.3 Complementary Effect of Temporal Cyclic and Chronological Patterns

The recommendation ability of temporal cyclic patterns and temporal chronological patterns has been demonstrated in previous sections. Each of them represents one type of human movement indication. The temporal cyclic patterns indicate the probability of next check-in happening at a specific POI given the next check-in time, while temporal chronological patterns indicate the probability of next check-in happening at a specific POI given the previous check-in POI sequence. To investigate their complementary effect, we formulate the POI recommendation problem as rec-

ommendation a POI to a user given his previous check-in POI sequence and the time when recommendation is performed, as shown below,

$$P_u(c_t = l|t, H_{u,t}), \quad (4.32)$$

where $c_t = l$ indicates that user u 's check-in at time t happening at POI l , t indicates the check-in time, and $H_{u,t}$ indicates the u 's check-in POI sequence before time t . Note that t here is a periodic time indicating the cyclic time stamp of the check-in, such as a specific hour (e.g., 23:00pm), a day of the week (e.g., Monday), a month (e.g., January) or even a year. Using Bayes' rule, the probability in Eq. (4.32) is equivalent to:

$$P_u(c_t = l|t, H_{u,t}) \propto P_u(c_t = l, t|H_{u,t}) = P_u(t|c_t = l, H_{u,t})P_u(c_t = l|H_{u,t}), \quad (4.33)$$

The two factorized terms $P_u(t|c_t = l, H_{u,t})$ and $P_u(c_t = l|H_{u,t})$ represent exactly temporal cyclic patterns and temporal chronological patterns [21], with their product capturing the complementary effect. These two probabilities restrain each other and complement each other in the form of spatio-temporal context, the candidate POI l with the highest probability $P(c_t = l|t, H_{u,t})$ is the one that most reasonably happens at time t , after the observation of previous check-ins $H_{u,t}$. The experimental results show impressive performance which are presented in [24, 21].

PERSONALIZED GEO-CONTENT POI RECOMMENDATION

In this chapter, we study the geo-content indications for personalized POI recommendation on LBSNs. As discussed in Chapter 1, content information on LBSNs is related to a user’s check-in action, containing three types of information regarding *POI properties*, *User Interests*, and *Sentiment Indications*. These three types of information are representatives of POI-Associated Content and User-Generated Content, as shown in Figure 5.1.

However, since each type of content information represents a different facet of check-in action, how to systematically model them for POI recommendation becomes a challenging problem, which relies on the investigation of their relationship to check-in actions and their complementary effects in affecting POI recommender systems. In this chapter, we study the three types of content information and propose a unified framework to model them. Specifically, we propose to:

- Study the relationship between users’ check-in behavior and content information on LBSNs in terms of POI properties, user interests, and sentiment indications.
- Incorporate the three types of content information into a unified framework for POI recommendation on LBSNs.
- Investigate the recommendation effort of each type of content information on a real-world LBSN dataset.

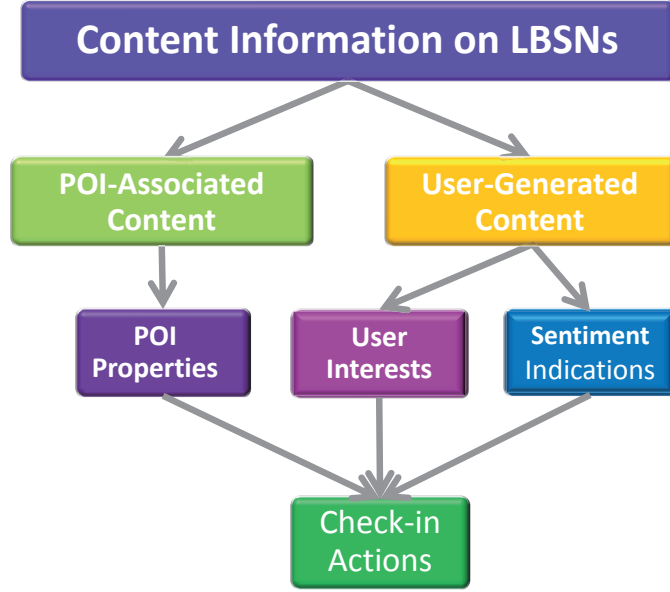


Figure 5.1: Content Information on LBSNs

5.1 A POI Recommendation Model with Geo-Content Indications

In chapter 3, we have introduced a basic matrix factorization model for location recommendation. In this section, we introduce a tri-factorization model and then discuss how to incorporate content information. The reason of using a tri-factorization model instead of a bi-factorization model as a basic model will be explained in later sections. Let $\mathbf{U} \in \mathbb{R}^{M \times K}$ be the users' latent interests, $\mathbf{V} \in \mathbb{R}^{N \times K}$ be the POIs' latent properties, and $\mathbf{H} \in \mathbb{R}^{K \times K}$ be the data-dependent dense matrix with $K \ll \min(M, N)$ being the number of latent factors. The basic POI recommendation model approximates u_i 's latent interests in an unvisited v_j by solving the following optimization problem:

$$\min_{\mathbf{U}, \mathbf{H}, \mathbf{V}_{j \geq 0}} \frac{1}{2} \sum_i^m \sum_j^n \mathbf{W}_{ij} (\mathbf{C}_{ij} - \mathbf{U}_i \mathbf{H} \mathbf{V}_j^\top)^2, \quad (5.1)$$

where $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a check-in weighting matrix with $\mathbf{W}_{ij} = 1$ indicating that u_i has checked in at v_j , $\mathbf{W}_{ij} = 0$ otherwise. A large value of \mathbf{W}_{ij} will force $\mathbf{U}_i \mathbf{H} \mathbf{V}_j^T$ to tightly fit the check-in \mathbf{C}_{ij} , suggesting that the corresponding check-in is more important to the user, while a lower value of \mathbf{W}_{ij} allows $\mathbf{U}_i \mathbf{H} \mathbf{V}_j^T$ to loosely fit \mathbf{C}_{ij} , suggesting the corresponding check-in is less important to the user.

The above recommendation model learns an optimal set of $\{\mathbf{U}, \mathbf{H}, \mathbf{V}\}$ whose product $\hat{\mathbf{C}} = \mathbf{U} \mathbf{H} \mathbf{V}^T$ is a non-sparse matrix which approximates the original check-in matrix \mathbf{C} . POI recommendation is then performed for each user based on the ranking among her unvisited POIs in $\hat{\mathbf{C}}$. Note that a non-negative constraint has been applied to \mathbf{U}_i , \mathbf{H} , and \mathbf{V}_j , respectively, as we consider that a user’s latent interests and a POI’s latent properties could have real-world explanations on LBSNs.

5.1.1 Modeling User Sentiment Indications

The basic POI recommendation model learns user latent interests and POI latent properties to approximate observed check-in actions. As discussed in the above section, \mathbf{W} is a weighting matrix applied to indicate the importance of check-ins, i.e., how likely the check-in action should be considered based on its observability, which is critical for improving recommendation performance [64, 57]. Previous work has discussed its potential effect when combined with information such as user reputation [75] or user activities [41] to better capture the importance of observed actions for recommendation. This inspires us to investigate how to incorporate sentiment information for capturing check-in behavior.

Sentiment information is embedded in the tips or comments that reflect users’ check-in experience. For example, if a user leaves a positive tip (or she likes it), the corresponding check-in is more important; otherwise it is less so. Thus, sentiment information can play a role as \mathbf{W} in Eq. (5.1) does in determining the importance of

check-ins. To incorporate sentiment information, we propose a **sentiment-enhanced weighting scheme** as a function $\hat{\mathbf{W}} = f(\mathbf{W}, \mathbf{S})$ which assigns weights on check-ins based on the corresponding check-in observability and sentiment indications. We use $\hat{\mathbf{W}}$ to replace the original weighting matrix while the function $f(\cdot)$ should have the following properties:

- **Sentiment Consistency**

For an observed check-in action, a positive sentiment indication on this check-in should increase its importance, while a negative sentiment indication should decrease its importance.

- **Sentiment Scaling**

To avoid over-weighting or under-weighting of sentiment information, the sentiment score in \mathbf{S} should be adjusted to an appropriate scale before adopting it for recommendation.

- **Non-Negativity**

The value in \mathbf{W} generated by $f(\cdot)$ should be non-negative according to the learning model in Eq. (5.1).

In this work, we empirically set $f(\cdot)$ as below, which works well in our model,

$$\hat{\mathbf{W}} = \mathbf{W} + \eta * \mathbf{S}, \quad \eta \in [0, 1], \quad (5.2)$$

where η is a scalar to control the weight from sentiment indications corresponding to the **Sentiment Scaling** property. Since observed check-ins have original weight $\mathbf{W}_{ij} = 1$, while the corresponding sentiment score $\mathbf{S}_{ij} \in [-1, 1]$, the hybrid score $\hat{\mathbf{W}}$ on an observed check-in is guaranteed to have the **Non-Negativity** property.

In Eq. (5.2), the importance of check-in actions is related to the corresponding sentiment score. A higher sentiment score \mathbf{S}_{ij} results in a greater value of $\hat{\mathbf{W}}_{ij}$, which

forces $\mathbf{U}_i\mathbf{H}\mathbf{V}_j^T$ to tightly fit the check-in \mathbf{C}_{ij} while $\mathbf{U}_i\mathbf{H}\mathbf{V}_j^T$ will loosely approximate \mathbf{C}_{ij} when $\hat{\mathbf{W}}_{ij}$ is smaller (corresponding to a lower \mathbf{S}_{ij}). In an extreme case, when $\hat{\mathbf{W}}_{ij}$ is 0, the check-in action is not considered at all; thus, its likelihood of being recommended to other users is reduced. This is consistent to the user’s sentiment indication, as $\hat{\mathbf{W}}_{ij} = 0$ happens when u_i presents the most negative score (-1) towards v_j (assuming $\eta=1$), which meets the **Sentiment Consistency** property.

5.1.2 Modeling User-Interest Content and POI-Property Content

Besides user sentiment indications, user-interest content is also embedded in tips and comments on LBSNs. Tips and comments contain semantic words that reflect a user’s interested topics regarding POIs, e.g., environment, taste, service. On the other hand, a user’s interests towards POIs are also indicated by her check-in behavior through corresponding visiting actions. According to the common assumption of transfer learning [59], we consider user interests as the intermediary to connect tipping/commenting behavior and check-in behavior, where the knowledge embedded in tips and comments can be leveraged as auxiliary information on check-in actions to better capture user interests. In the meantime, it can also help address the data sparsity problem of check-in actions to a certain extent, as insufficient observation of check-in behavior can be compensated by the observed tipping/commenting behavior for inferring user interests. Thus, we propose the leveraging of information from tips and comments to improve the learning of user latent interests, as shown below,

$$\min \frac{1}{2} \sum_i^M \sum_j^N (\mathbf{A}_{ij} - \mathbf{U}_i\mathbf{G}_j)^2, \quad (5.3)$$

where \mathbf{G}_j represents the word properties on latent topics in user-interest content.

Similarly, information from POI-property content can also be leveraged to learn

the POI latent properties, as shown below,

$$\min \frac{1}{2} \sum_i^M \sum_j^N (\mathbf{B}_{ij} - \mathbf{V}_i \hat{\mathbf{G}}_j)^2, \quad (5.4)$$

where $\hat{\mathbf{G}}_j$ represents the word properties on latent topics in POI-property content.

Both \mathbf{G}_j and $\hat{\mathbf{G}}_j$ represent word latent topics, where the former is in user context related to user-interest content, and the latter is in POI context related to POI-property content. Thus, we expect these two latent topics to be different but with certain overlaps, and propose a ℓ -1 norm to capture such relationship,

$$\min \|\mathbf{G} - \hat{\mathbf{G}}\|_1 \quad (5.5)$$

where $\|\cdot\|$ is the ℓ -1 norm regularization, with $\|X\|_1 = \sum_i \sum_j |X_{ij}|$.

5.1.3 CAPRF: Content-Aware POI Recommendation Framework

According to the model described in above sections, our content-aware POI recommendation framework, **CAPRF**, aims to solve the following optimization problem,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{H}, \mathbf{V} \geq 0} & \frac{1}{2} \|\hat{\mathbf{W}} \odot (\mathbf{C} - \mathbf{U}\mathbf{H}\mathbf{V}^\top)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{A} - \mathbf{U}\mathbf{G}\|_F^2 \\ & + \frac{\lambda_2}{2} \|\mathbf{B} - \mathbf{V}\hat{\mathbf{G}}\|_F^2 + \delta \|\mathbf{G} - \hat{\mathbf{G}}\|_1 \\ & + \frac{\alpha}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{H}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{G}\|_F^2 + \|\hat{\mathbf{G}}\|_F^2), \end{aligned} \quad (5.6)$$

where λ_1 and λ_2 are introduced to control the weight of user-interest content and POI-property content. δ is to control the closeness between \mathbf{G}_j and $\hat{\mathbf{G}}$. The regularization terms $\|\mathbf{U}\|_F^2$, $\|\mathbf{H}\|_F^2$, $\|\mathbf{V}\|_F^2$, $\|\mathbf{G}\|_F^2$, and $\|\hat{\mathbf{G}}\|_F^2$ are used to avoid overfitting.

Table 5.1 lists the relevant notations. Our content-aware POI recommendation framework is illustrated in Figure 5.2. Check-in action \mathbf{C} is directly related to sentiment indications \mathbf{S} , user interests \mathbf{U} , and POI properties \mathbf{V} , where the latter two

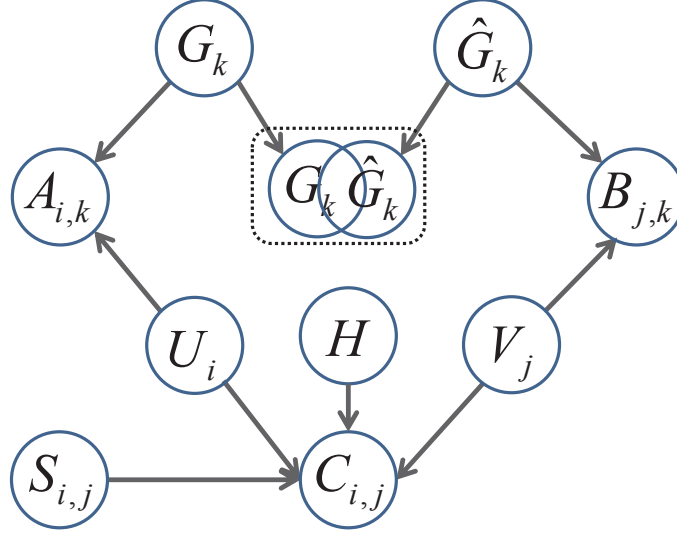


Figure 5.2: Content-Aware POI Recommendation Framework

are learned from the factorization of \mathbf{C} with the consideration of a data dependent matrix \mathbf{H} for model flexibility. User interest \mathbf{U} is also related to user-interest content \mathbf{A} , which represents tipping/commenting actions factorized to \mathbf{U} and word properties \mathbf{G} . POI properties \mathbf{V} is also related to POI-property content \mathbf{B} , which represents POI descriptions factorized to \mathbf{V} and word properties $\hat{\mathbf{G}}$. \mathbf{G} and $\hat{\mathbf{G}}$ are considered to be close to each other. The input of our framework is user check-in action \mathbf{C} , user sentiment indications \mathbf{S} , user-interest content \mathbf{A} , and POI-property content \mathbf{B} , and the output is \mathbf{U} , \mathbf{H} , and \mathbf{V} , whose product $\mathbf{U}\mathbf{H}\mathbf{V}^\top$ is used for POI recommendation.

5.1.4 Parameter Estimation

Previous sections have discussed the modeling of different types of content information under a unified framework for POI recommendation. In this section, we introduce a learning method for the parameters involved in our model.

In Eq. (5.6), \mathbf{G} and $\hat{\mathbf{G}}$ are both constrained by ℓ -1 norm regularization, resulting

Table 5.1: Mathematical Notation

Symbol	Size	Description
C	$M \times N$	User-POI Check-in Matrix
W	$M \times N$	User-POI Weighting Matrix
S	$M \times N$	User-POI Sentiment Matrix
A	$M \times T$	User-Word Matrix
B	$N \times T$	POI-Word Matrix
U	$M \times K$	User Latent Interests
V	$N \times K$	POI Latent Properties
H	$K \times K$	Data Dependent Dense Matrix
G, $\hat{\mathbf{G}}$, \mathbf{D}	$K \times T$	Word Latent Factors

in the difficulty to solve each of them. In this work, we propose an equivalent way to solve this problem via representing $\mathbf{G} - \hat{\mathbf{G}}$ as one variable \mathbf{D} , which rephrases the original problem as solving the ℓ_1 norm regarding \mathbf{D} , as shown below:

$$\mathbf{D} = \mathbf{G} - \hat{\mathbf{G}}. \quad (5.7)$$

According to Eq. (5.6) and Eq. (5.7), the original optimization problem can be rephrased as

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{H}, \mathbf{V} \geq 0} \mathcal{J} &= \frac{1}{2} \|\hat{\mathbf{W}} \odot (\mathbf{C} - \mathbf{U}\mathbf{H}\mathbf{V}^\top)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{A} - \mathbf{U}\mathbf{G}\|_F^2 \\ &+ \frac{\lambda_2}{2} \|\mathbf{B} - \mathbf{V}(\mathbf{G} - \mathbf{D})\|_F^2 + \delta \|\mathbf{D}\|_1 \\ &+ \frac{\alpha}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{H}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{G}\|_F^2) \end{aligned} \quad (5.8)$$

Since there are multiple variables in the object function, and alternative algorithm is commonly used, as it is difficult to provide a direct closed-form solution for the above optimization problem. Thus, we apply an alternative algorithm to find optimal

solutions for the five variables \mathbf{U} , \mathbf{H} , \mathbf{V} , \mathbf{G} , and \mathbf{D} . The key idea is to minimize the objective function w.r.t. one variable while fixing the other variables, as similar to [56]. The algorithm will keep updating the variables until convergence or reaching the number of maximum iterations.

Computation of $\{\mathbf{U}, \mathbf{H}, \mathbf{V}, \mathbf{G}\}$

The objective function \mathcal{J} in Eq. (5.8) is differentiable at \mathbf{U} , \mathbf{H} , \mathbf{V} , and \mathbf{G} , the derivation of \mathcal{J} with respect to them are

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{U}} &= -(\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot \mathbf{C})\mathbf{V}\mathbf{H}^\top + (\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot (\mathbf{U}\mathbf{H}\mathbf{V}^\top))\mathbf{V}\mathbf{H}^\top \\
&\quad - \lambda_1 \mathbf{A}\mathbf{G}^\top + \lambda_1 \mathbf{U}\mathbf{G}\mathbf{G}^\top + \alpha \mathbf{U} \\
\frac{\partial \mathcal{J}}{\partial \mathbf{H}} &= -\mathbf{U}^\top (\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot \mathbf{C})\mathbf{V} + \mathbf{U}^\top (\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot (\mathbf{U}\mathbf{H}\mathbf{V}^\top))\mathbf{V} + \alpha \mathbf{H} \\
\frac{\partial \mathcal{J}}{\partial \mathbf{V}} &= -(\hat{\mathbf{W}}^\top \odot \hat{\mathbf{W}}^\top \odot \mathbf{C}^\top)\mathbf{U}\mathbf{H} + (\hat{\mathbf{W}}^\top \odot \hat{\mathbf{W}}^\top \odot (\mathbf{V}\mathbf{H}^\top \mathbf{U}^\top))\mathbf{U}\mathbf{H} \\
&\quad - \lambda_2 (\mathbf{B} - \mathbf{V}\mathbf{G} + \mathbf{V}\mathbf{D})(\mathbf{G}^\top - \mathbf{D}^\top) + \alpha \mathbf{V} \\
\frac{\partial \mathcal{J}}{\partial \mathbf{G}} &= -\lambda_1 \mathbf{U}^\top \mathbf{A} + \lambda_1 \mathbf{U}^\top \mathbf{U}\mathbf{G} - \lambda_2 \mathbf{V}^\top (\mathbf{B} - \mathbf{V}\mathbf{G} + \mathbf{V}\mathbf{D}) + \alpha \mathbf{G} \tag{5.9}
\end{aligned}$$

The gradient descent optimization method is widely applied to update the above variables, and usually works well in recommender systems [36]. For the non-negative constraints on \mathbf{U} , \mathbf{H} , and \mathbf{V} , we applied projected strategy, which projects a negative parameter value to 0 in each iteration. The detailed updating rules are shown in Algorithm 3, where $\gamma_{\mathbf{u}}$, $\gamma_{\mathbf{h}}$, $\gamma_{\mathbf{v}}$, and $\gamma_{\mathbf{g}}$ are learning steps, which are chosen to satisfy Goldstein Conditions [34].

Computation of \mathbf{D}

Optimizing Eq. (5.8) w.r.t. \mathbf{D} is equivalent to

$$\min_{\mathbf{D}} \frac{\lambda_2}{2} \|\mathbf{B} - \mathbf{V}(\mathbf{G} - \mathbf{D})\|_F^2 + \delta \|\mathbf{D}\|_1 \tag{5.10}$$

Since ℓ_1 -norm regularization is applied to \mathbf{D} , the objective function in Eq. (5.11) is a non-smooth convex problem. Proximal gradient descent has recently received increasing attention, and is able to deal with the non-smooth convex problem. It considers the objective function as a composite of a smooth part and a non-smooth part, as shown below:

$$\begin{aligned} f(\mathbf{D}) &= \frac{\lambda_2}{2} \|\mathbf{V}\mathbf{D} - (\mathbf{V}\mathbf{G} - \mathbf{B})\|_F^2 \\ F(\mathbf{D}) &= f(\mathbf{D}) + \delta \|\mathbf{D}\|_1 \end{aligned} \quad (5.11)$$

It is known that $f(\mathbf{D})$ is convex and differentiable, while $\delta \|\mathbf{D}\|_1$ is non-smooth but convex. In each iteration of the proximal gradient descent, the value of \mathbf{D} is updated as below

$$\mathbf{D}_{t+1} = \arg \min_{\mathbf{D}} L_{\gamma_t}(\mathbf{D}, \mathbf{D}_t), \quad (5.12)$$

where

$$\begin{aligned} L_{\gamma_t}(\mathbf{D}, \mathbf{D}_t) &= f(\mathbf{D}_t) + \langle \nabla f(\mathbf{D}_t), \mathbf{D} - \mathbf{D}_t \rangle \\ &\quad + \frac{\gamma_t}{2} \|\mathbf{D} - \mathbf{D}_t\|^2 + \delta \|\mathbf{D}\|_1 \end{aligned} \quad (5.13)$$

By ignoring the terms in $L_{\gamma_t}(\mathbf{D}, \mathbf{D}_t)$ that are independent of \mathbf{D} , the original optimization problem in Eq. (5.11) boils down to

$$\mathbf{D}_{t+1} = \arg \min_{\mathbf{D}} \frac{1}{2} \|\mathbf{D} - \mathbf{Y}_t\|_F^2 + \frac{\delta}{\gamma_t} \|\mathbf{D}\|_1, \quad (5.14)$$

where $\mathbf{Y}_t = \mathbf{D}_t - \frac{1}{\gamma_t} \nabla f(\mathbf{D}_t)$. $\nabla f(\mathbf{D}_t)$ is the gradient of $f(\mathbf{D}_t)$. In our problem, $\nabla f(\mathbf{D}_t)$ is defined as:

$$\nabla f(\mathbf{D}_t) = -\lambda_2 \mathbf{V}^\top \mathbf{V} \mathbf{D} - \lambda_2 \mathbf{V}^\top \mathbf{V} \mathbf{G} + \lambda_2 \mathbf{V}^\top \mathbf{B} \quad (5.15)$$

Eq. (5.14) can be further decomposed into k separate sub problems as

$$\mathbf{d}_{t+1}^i = \arg \min \|\mathbf{d}^i - \mathbf{y}_t^i\|_2^2 + \frac{\delta}{\gamma_t} \|\mathbf{d}^i\|_1, \quad (5.16)$$

where \mathbf{d}_{t+1}^i , \mathbf{d}^i , and \mathbf{y}_t^i are the i -th rows of \mathbf{D}_{t+1}^i , \mathbf{D}^i , and \mathbf{Y}_t^i , respectively. It has a closed form solution according to [46].

$$\mathbf{d}_{t+1}^i = \text{sign}(\mathbf{y}_t^i) \odot \max(|\mathbf{y}_t^i| - \frac{\delta}{\gamma_t}, 0) \quad (5.17)$$

The convergence rate of the above method is $O(\frac{1}{\epsilon})$. As suggested in [45], it can be further accelerated to achieve the convergence rate of $O(\frac{1}{\sqrt{\epsilon}})$ with Nesterov's method [52], which is based on a linear combination of \mathbf{D}_{t+1}^i and \mathbf{D}^i as search points

$$\mathbf{Z}_t = \mathbf{D}_t + \frac{\sigma_{t-1} - 1}{\sigma_t} (\mathbf{D}_t - \mathbf{D}_{t-1}), \quad (5.18)$$

where $\{\sigma_t\}_{t \geq 1}$ is conventionally set to be $\sigma_{t+1} = \frac{1 + \sqrt{1 + 4\sigma_t^2}}{2}$. The detailed learning algorithm of \mathbf{D} with above accelerated method is shown in Algorithm 2.

5.1.5 Algorithm Analysis

The detailed learning algorithm of our content-aware recommendation framework is shown in Algorithm 3. In lines 1-2, all the parameters are firstly initialized randomly, where \mathbf{W} is generated through check-in action \mathbf{C} , based on which $\hat{\mathbf{W}}$ is constructed with \mathbf{S} . From lines 3 to 10, the algorithm iteratively updates \mathbf{U} , \mathbf{H} , \mathbf{V} , \mathbf{G} , and \mathbf{D} until convergence. The final output of this algorithm is $\hat{\mathbf{C}}$, which is the product of \mathbf{U} , \mathbf{H} , and \mathbf{V} . We perform POI recommendation for each user based on the corresponding ranking among her unvisited POIs in $\hat{\mathbf{C}}$.

Compared to other operations, the updating rules for \mathbf{U} , \mathbf{H} , \mathbf{V} , \mathbf{G} , and \mathbf{D} in each iteration correspond to the major cost of Algorithm 3. Therefore, we analyze the time complexity of updating operations. For the updating rule of \mathbf{U} , $(\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot \mathbf{C})\mathbf{V}\mathbf{H}^\top$ and $(\hat{\mathbf{W}} \odot \hat{\mathbf{W}} \odot (\mathbf{U}\mathbf{H}\mathbf{V}^\top))\mathbf{V}\mathbf{H}^\top$ take $O(MKN)$ operations, and $\lambda_1 \mathbf{A}\mathbf{G}^\top$ and $\lambda_1 \mathbf{U}\mathbf{G}\mathbf{G}^\top$ take $O(MKT)$ operations, while the time complexity for $\alpha\mathbf{U}$ is $O(MK)$. Therefore, it takes $O(MKN) + O(MKT)$ operations to update \mathbf{U} . Similarly, the time complexity

Algorithm 2 The Learning Algorithm of **D**

Input: $\{\mathbf{B}, \mathbf{V}, \mathbf{G}, \lambda_2, \delta\}$, and max iteration number q

Output: **D**

- 1: Initialize **D** randomly
 - 2: Set $\mathbf{D}_1 = \mathbf{D}_0 = \mathbf{D}$, $\sigma_0 = 0$, $\sigma_1 = 1$, $t = 1$, $\gamma_1 = 1$
 - 3: **for** $t = 1$ **to** q **do**
 - 4: Set $\mathbf{Z}_t = \mathbf{D}_t + \frac{\sigma_{t-1}-1}{\sigma_t}(\mathbf{D}_t - \mathbf{D}_{t-1})$
 - 5: Set $\nabla f(\mathbf{Z}_t) = -\lambda_2 \mathbf{V}^\top \mathbf{V} \mathbf{Z}_t - \lambda_2 \mathbf{V}^\top \mathbf{V} \mathbf{G} + \lambda_2 \mathbf{V}^\top \mathbf{B}$
 - 6: **while** true **do**
 - 7: Set $\mathbf{Y}_t = \mathbf{Z}_t - \frac{1}{\gamma_t} \nabla f(\mathbf{Z}_t)$
 - 8: Compute $\mathbf{D}_{t+1} = \arg \min_D L_{\gamma_t}(\mathbf{D}, \mathbf{Z}_t)$ according to Eq. (5.17)
 - 9: **if** $f(\mathbf{D}_{t+1}) \leq L_{\gamma_t}(\mathbf{D}_{t+1}, \mathbf{Z}_t)$ **then**
 - 10: Set $\gamma_{t+1} = \gamma_t$, break
 - 11: **end if**
 - 12: Set $\gamma_t = 2 * \gamma_t$
 - 13: **end while**
 - 14: Set $\mathbf{D} = \mathbf{D}_{t+1}$
 - 15: Set $\sigma_{t+1} = \frac{1 + \sqrt{1 + 4\sigma_t^2}}{2}$
 - 16: Set $t = t + 1$
 - 17: **end for**
 - 18: **return** **D**
-

of updating \mathbf{H} , \mathbf{V} , and \mathbf{G} are $O(MKN)$, $O(MKN) + O(NKT)$, and $O(MKT) + O(NKT)$, respectively. To update D , it takes $O(NKT)$ operations to compute the gradient and $O(K)$ operations for the ℓ_1 -norm regularization part. In sum, the time complexity of Algorithm 3 is $(\# \text{ of iterations}) * (O(MKN) + O(NKT) + O(MKT))$.

Algorithm 3 The Learning Algorithm of the Proposed Model

Input: user-POI check-in matrix \mathbf{C} , sentiment indication matrix \mathbf{S} , user-interest content \mathbf{A} , POI-property content \mathbf{B} , parameters $\{\eta, \lambda_1, \lambda_2, \delta, \alpha\}$

Output: approximated user-POI preference matrix $\tilde{\mathbf{C}}$

- 1: Initialize \mathbf{U} , \mathbf{H} , \mathbf{V} , and \mathbf{G} randomly
 - 2: Set $\mathbf{W} = \text{sign}(\mathbf{C})$, $\hat{\mathbf{W}} = \mathbf{W} + \eta * \mathbf{S}$
 - 3: **while** Not Convergent **do**
 - 4: Calculate $\frac{\partial \mathcal{J}}{\partial \mathbf{U}}$, $\frac{\partial \mathcal{J}}{\partial \mathbf{H}}$, $\frac{\partial \mathcal{J}}{\partial \mathbf{V}}$, and $\frac{\partial \mathcal{J}}{\partial \mathbf{G}}$
 - 5: Update $\mathbf{U} \leftarrow \max(\mathbf{U} - \gamma_{\mathbf{u}} \frac{\partial \mathcal{J}}{\partial \mathbf{U}}, 0)$
 - 6: Update $\mathbf{H} \leftarrow \max(\mathbf{H} - \gamma_{\mathbf{h}} \frac{\partial \mathcal{J}}{\partial \mathbf{H}}, 0)$
 - 7: Update $\mathbf{V} \leftarrow \max(\mathbf{V} - \gamma_{\mathbf{v}} \frac{\partial \mathcal{J}}{\partial \mathbf{V}}, 0)$
 - 8: Update $\mathbf{G} \leftarrow \mathbf{G} - \gamma_{\mathbf{g}} \frac{\partial \mathcal{J}}{\partial \mathbf{G}}$
 - 9: Update \mathbf{D} according to Algorithm 2
 - 10: **end while**
 - 11: **return** $\tilde{\mathbf{C}} = \mathbf{U}\mathbf{H}\mathbf{V}^\top$
-

5.2 Experiments

In this section, we evaluate the performance of our proposed framework **CAPRF** for POI recommendation. In particular, we evaluate the following: (1) how the proposed framework fares in comparison with state-of-the-art recommendation systems; and (2) how different kinds of content information perform in the POI recommendation task. Before we delve into experiment details, we first discuss an LBSN dataset

and evaluation metrics.

5.2.1 Foursquare Dataset

We use Foursquare dataset to study the content information on LBSNs. We collect users whose Foursquare profiles indicate their hometown as being California or New York state. We then obtain their corresponding check-in tweets through Twitter’s public REST API with the same crawling strategy as proposed in [67, 22], and collect check-ins that happened in the corresponding state. A check-in tweet contains a unique URL that directs to a Foursquare web page indicating check-in POI information. Based on the venue id extracted from check-in tweets, we obtain the POI category through the “Venue API”¹ of Foursquare. We select check-ins happened at POI of the “Food” category, which is the largest category among all the POIs in Foursquare. We obtain the POI-associated content (tags) through the venue API as well.

To collect user-generated content, we combine both check-in tips and check-in comments. Check-in tips are collected through the “Tip API”² of foursquare, while check-in comments are embedded in the check-in tweets in which we remove the system-generated comment formatted “I’m at xxx 4sq.com/xxx.” Note that tips and comments have no overlaps, as tipping is an independent action on Foursquare that requires a user to specifically post, while comments can be left when a user performs check-in actions. In our experiment, we consider users who have checked-in at least 2 distinct POIs. The statistics of the final dataset are shown in Table 5.2. Figure 5.3 and Figure 5.4 show the check-in distribution over California and New York State in our dataset, respectively.

¹<https://developer.foursquare.com/docs/venues/venues>

²<https://developer.foursquare.com/docs/users/tips>

Table 5.2: Statistical Information of the Dataset

	CA	NY
Number of Users	4,287	6,043
Number of Check-ins	134,556	207,591
Number of POIs	5,878	5,937
Number of Tips	19,741	40,539
Number of Comments	56,718	78,290
Check-in Duration	May, 2008-Sep, 2013	

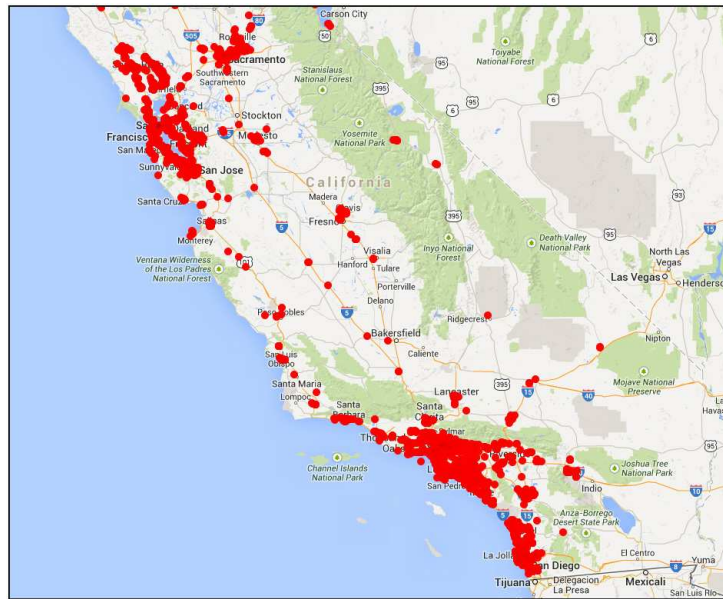


Figure 5.3: Check-in Distribution over the California State

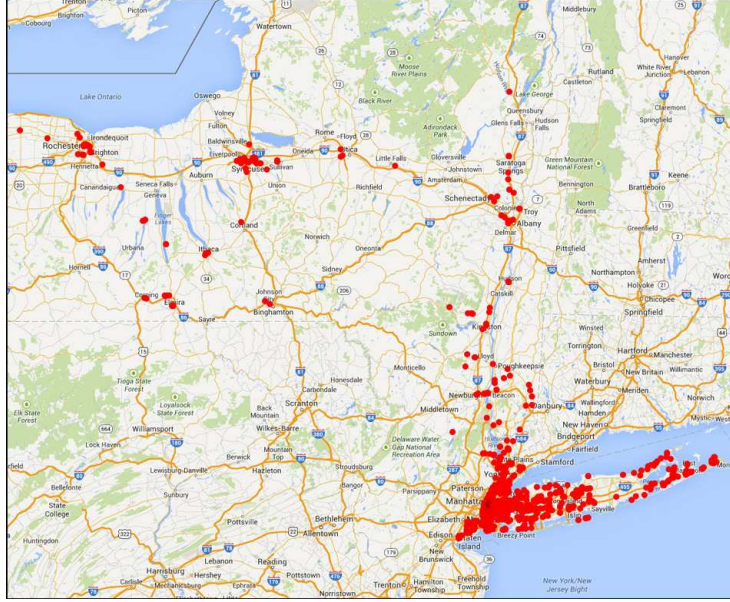


Figure 5.4: Check-in Distribution over the New York State

5.2.2 Experimental Setup

The input of our framework is the observed check-in action matrix \mathbf{C} and three types of content information \mathbf{S} , \mathbf{A} , and \mathbf{B} . We first introduce how we organize this input information in our experiments, then discuss the experimental settings and evaluation metrics.

- **Check-in Matrix \mathbf{C}**

We organize the check-in actions as a user-POI matrix \mathbf{C} . The check-in density of the matrix is 5.34×10^{-4} for CA data, and 5.79×10^{-4} for NY data. We adopt the mapping function $\frac{1}{1+x^{-1}}$, which has been proven to work well for POI recommendation in our previous work [21].

- **Sentiment Indication Matrix \mathbf{S}**

We generate the user sentiment indication matrix \mathbf{S} from tips and comments with an unsupervised sentiment classification method. For each tip/comment,

we remove stop words and employ a word-matching scheme to compute its sentiment score based on a sentiment lexicon. Sentiment polarity of a word is obtained from the pre-defined sentiment lexicon, i.e., -1 for negative and $+1$ for positive. The overall sentiment score of a tip/comment is computed as the summation of sentiment scores of the words in the tip/comment, and normalized to $[-1,1]$ by taking the average on the tip/comment length. We adopt the MPQA Subjectivity Lexicon³, which is a widely used manually labeled sentiment lexicon containing 2,718 positive and 4,902 negative words.

- **User-Interest Matrix \mathbf{A} & POI-Property Matrix \mathbf{B}**

We select the common words of user-interest content and POI-property content, construct them as a user-word matrix \mathbf{A} and a POI-word matrix \mathbf{B} , with the matrix entry representing the frequency of a word used by corresponding user/POI. The total number of common words in the CA dataset and NY dataset are 1,810 and 1,906, respectively.

For each individual user in the check-in matrix, we randomly mark off 20% of all POIs that he has checked-in for testing. The rest of the observed user-POI pairs are used as training data for POI recommendation. The random selection is conducted 5 times individually, and we report the average results. Since only the observed check-in actions (corresponding to $\mathbf{W}_{ij} = 1$) are considered in Eq. (5.1), following the standard strategy of solving one-class CF problems [57, 58], we sample 10% of unobserved check-ins from the training matrix, deem them as the check-in frequency of 0 and set their corresponding \mathbf{W}_{ij} to 1. The same strategy is also performed on baseline methods.

³http://mpqa.cs.pitt.edu/lexicons/subj_lexicon

To evaluate the recommendation performance, we use $precision@N$ and $recall@N$ in Eq. (4.17) as our evaluation metrics. In our experiment, N is set to be 5 and 10. All the parameters in this work are set through cross-validation. For the proposed method, the experimental results use $d=20$ dimensions to represent the latent features, the parameters $\{\eta, \lambda_1, \lambda_2, \delta, \alpha\}$ are set to $\{0.3, 0.1, 0.1, 0.8, 0.1\}$ in the CA dataset and $\{0.2, 0.3, 0.3, 0.5, 0.1\}$ in the NY dataset.

5.2.3 Performance Evaluation

In this section, we compare our POI recommendation framework with existing state-of-the-art methods. Five baseline methods are introduced w.r.t. different types of content, as defined below:

- **User-Based Collaborative Filtering (UCF)**

User-based collaborative filtering is a state-of-the-art approach for recommender systems. We adopt the user-based recommender [101] for POI recommendation. It computes a user’s interests in a location based on other users’ interests in that POI. Content information is not considered in this approach.

- **Probabilistic Matrix Factorization (PMF)**

PMF is a classical matrix factorization approach which factorizes the user-item actions into user interests and item properties for recommendation [64]. In this work, we regard POIs as items while content information is not used.

- **Non-negative Matrix Factorization (NMF)**

Non-negative Matrix Factorization (NMF) [38] computes non-negative user check-in preferences under the user-POI matrix. In this work, we adopt the tri-factorization model, which is our basic POI recommendation model, as defined in Eq. (5.1), without considering content effects.

- **Spatial Topic Location Recommender (STLR)**

STLR performs location recommendation under a topic model which considers the user-interest content [31]. It approximates the user interests and location properties based on the content information left by a user at a location. The POI-property content and sentiment information is not used.

- **Sentiment-Enhanced Location Recommender (SELR)**

SELR is a recommendation model based on probabilistic matrix factorization that uses sentiment information and venue categories for location recommendation [16]. From a content view, user-interest content and POI-property content (tags) are not used in this work.

Note that there are existing works that are using POI-property content only for POI recommendation [88]. In this work, we do not consider them as baseline methods, as these methods are “location-aware”, i.e., when making a recommendation, the system is aware of the user’s current position in terms of a city or activity region, and selects POI candidates within the region for recommendation. Since our model does not have this assumption, for fair comparison, we do not perform comparison with these methods. However, in the next section, we investigate the recommendation efforts of each type of content information with our proposed model, which gives the interpretation on the effect of POI-property content.

Table 5.3 and Table 5.4 report the comparison results of **CAPIR** with the proposed baseline methods. The results precipitate several observations, which we summarize below:

- UCF performs the worst among all the approaches. Data sparseness is a possible reason for its performance. Due to the low density of the check-in matrix, the user-based collaborative filtering approach fails to accurately recommend POIs

Table 5.3: Performance Comparison (CA)

Methods	Precision		Recall	
	P@5	P@10	R@5	R@10
UCF	0.0083	0.0077	0.0117	0.0216
PMF	0.0114	0.0104	0.0160	0.0292
NMF	0.0126	0.0111	0.0177	0.0310
STLR	0.0173	0.0150	0.0243	0.0422
SELR	0.0134	0.0121	0.0188	0.0340
CAPRF	0.0186	0.0169	0.0261	0.0474

Table 5.4: Performance Comparison (NY)

Methods	Precision		Recall	
	P@5	P@10	R@5	R@10
UCF	0.0052	0.0047	0.0056	0.0103
PMF	0.0109	0.0099	0.0119	0.0151
NMF	0.0114	0.0102	0.0124	0.0223
STLR	0.0138	0.0125	0.0150	0.0273
SELR	0.0124	0.0113	0.0136	0.0247
CAPRF	0.0158	0.0143	0.0172	0.0311

and performs worse than matrix factorization approaches which leverage the low-rank approximation of user check-in preferences.

- Both PMF and NMF perform better than CF, demonstrating their ability in dealing with sparse data for POI recommendation. However, since there is no content information used in PMF and NMF, their recommendation performance is worse than content-based POI recommendation approaches.
- SELR and STLR perform better than UCF, PMF, and NMF, suggesting the importance of sentiment information and user-interest content. Furthermore, the better performance of STLR over SELR indicates that user-interest content seems to be more effective than sentiment information for POI recommendation. We will further discuss this in the next section.
- Among all the approaches, our proposed model CAPRF performs the best, suggesting the importance of content information on LBSNs for POI recommendation. Our model, with the consideration of different types of content information w.r.t. sentiment indications, user-interest content, and POI-property content, is able to improve POI recommendation performance, which indicates its ability to capture the relationship between content information and check-in actions on LBSNs.

It is worth noting that precision and recall in our experiments are not high. As discussed in Chapter 4, the effectiveness of recommender systems with sparse datasets is usually very low. For example, the reported top 5 precision is 5% over a dataset with 8.02×10^{-3} density [84, 86]. Therefore, the low precision obtained in our experiment is reasonable. In this work, we focus on **comparing algorithms' relative performance instead of their absolute performance**.

5.2.4 Evaluation of Different Types of Content Information

In this section, we discuss the recommendation efforts of different types of content information on LBSNs, i.e., Sentiment Indications (**SI**), User-Interest Content (**UIC**), and POI-Property Content (**PPC**). To evaluate each type of content information and their complementary effect, we propose to consider their different combinations by setting the corresponding parameter η (for sentiment indications), λ_1 (for user-interest content), and λ_2 (for POI-property content). For each parameter, if the corresponding type of content is considered, we set it to the optimal value; otherwise, 0. Since the parameter δ relates to both user-interest content and POI-property content, we set it to the optimal value when both types of content information have been considered, and 0 if only one or none of them has been considered. In the next section, we will specifically investigate this parameter.

Table 5.5 and Table 5.6 lists the comparison results with the consideration of different types of content information and their combinations. We use \surd to indicate that the corresponding type of content information is used, and \times otherwise. From the result, we observe the recommendation effect of different content information with their complementary effects for POI recommendation. We summarize the key observations below:

- Sentiment information is helpful in improving the POI recommendation performance. It consistently improves the performance based on existing content information. For example, it achieves approximately 3.60% relative improvement over the “None” model, and 3.10% relative improvement over the “UIC+PPC” model on CA data. Similar improvement can also be observed on NY data. However, it seems that the recommendation effect of sentiment information is not as great as user-interest content and POI-property content. One possi-

ble reason could be that sentiment information is quite noisy in user-generated content, while the non-perfect lexicon-based sentiment classification approach exacerbates the capturing of essential user attitude content, resulting in a piece of valuable but noisy or even inaccurate information for POI recommendation.

- User-interest content presents more recommendation effect than the other two types of content information. Compared to POI-property content, user-interest content achieves, on average, 7% \sim 10% relative improvement with either individual content information or multiple content information on both datasets. One possible reason of this improvement could be the different frequency of words (tags) in the two types of content information. In POI-property content, a word/tag is commonly only mentioned once with a POI, while in user-interest content, a word/tag can be mentioned many times in users' tips or comments. Thus, word information will be more helpful to distinguish user interests in user-interest content than POI properties in POI-property content.
- The combination of all three types of content information, i.e., CAPRF, has the best performance among all the other methods. This indicates the potential complementary effect among the three types of content information. According to Table 1.1, this information constitutes the key factors of POI recommender systems regarding a set of check-in related considerations, i.e., "what is the POI about" (POI properties), "am I interested" (user interests), and "how good is the POI" (sentiment indications).

5.2.5 Parameter Analysis

In this section, we analysis the parameters in our recommendation framework **CAPRF** w.r.t. η (weight of sentiment indications), λ_1 (weight of user-interest con-

Table 5.5: Recommendation Effect of Different Types of Content Information

Information	SI	UIC	PPC	CA			
				P@5	R@5	P@10	R@10
NONE	×	×	×	0.0126	0.0177	0.0111	0.0310
SI	✓	×	×	0.0130	0.0182	0.0115	0.0323
UIC	×	✓	×	0.0164	0.0230	0.0148	0.0416
PPC	×	×	✓	0.0154	0.0217	0.0138	0.0387
UIC+SI	✓	✓	×	0.0172	0.0242	0.0154	0.0433
PPC+SI	✓	×	✓	0.0157	0.0221	0.0141	0.0396
UIC+PPC	×	✓	✓	0.0178	0.0250	0.0162	0.0456
SI+UIC+PPC	✓	✓	✓	0.0186	0.0261	0.0169	0.0474

Table 5.6: Recommendation Effect of Different Types of Content Information

Information	SI	UIC	PPC	NY			
				P@5	R@5	P@10	R@10
NONE	×	×	×	0.0110	0.0120	0.0102	0.0223
SI	✓	×	×	0.0117	0.0127	0.0108	0.0234
UIC	×	✓	×	0.0138	0.0151	0.0128	0.0279
PPC	×	×	✓	0.0130	0.0141	0.0120	0.0261
UIC+SI	✓	✓	×	0.0140	0.0152	0.0130	0.0283
PPC+SI	✓	×	✓	0.0133	0.0144	0.0123	0.0267
UIC+PPC	×	✓	✓	0.0149	0.0163	0.0138	0.0301
SI+UIC+PPC	✓	✓	✓	0.0154	0.0168	0.0143	0.0311

tent), λ_2 (weight of POI-property content), and δ (control of overlapped semantic space). We investigate each parameter by evaluating the model performance when varying the value of one parameter and keeping the other parameters fixed as their optimal values. Figure 5.5 to Figure 5.8 plot the performance w.r.t these parameters. Since $\eta \in [0, 1]$, we increase its value from 0 to 1 with a step 0.1, and observe its performance. For λ_1 , λ_2 , and δ , we set their values as $\{0, 1e-4, 1e-3, 1e-2, 1e-1, 1, 10, 100, 1000\}$. From the figures, we observe the following:

- When η increases from 0 to 1, the recommendation performance on both CA and NY datasets exhibits a similar trend, i.e., it first increases, reaches its peak, then decreases, indicating the sensitivity of η to the model performance. When η is small, the sentiment indications are not fully considered; thus the learning of user interests and POI properties are affected more by check-in actions. With the increasing of η , the model considers sentiment information more than check-in actions, resulting in a poor recommendation performance as the former is much noisier and sparser than the latter.
- The varying of λ_1 and λ_2 presents a similar trend on the recommendation performance. According to Figure 5.6 and Figure 5.7, the optimal values of these two parameters are between 0 and 1, while the recommendation performance rapidly reduces when λ_1/λ_2 is smaller than 0 or larger than 1. This suggests that content information can be helpful to improve POI recommendation when it is considered auxiliary information on the check-in information for learning user interests and POI properties. A large λ_1/λ_2 will place a strong constraint on user interests/POI properties, therefore making the model severely over-fitted on content information, resulting in the poor recommendation performance.
- Increasing δ immediately improves the recommendation performance, suggest-

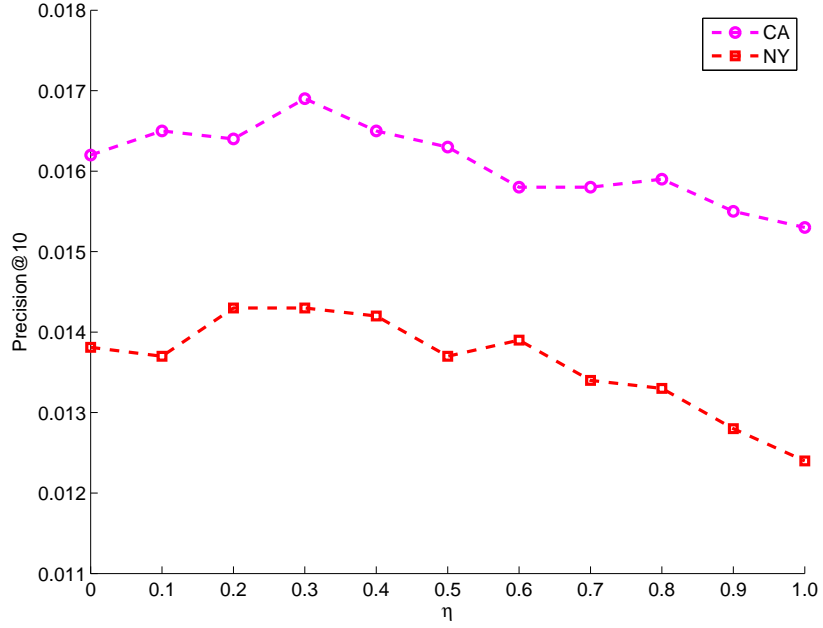


Figure 5.5: Sentiment Indications- η

ing its importance. δ controls the sparseness of $\mathbf{G} - \hat{\mathbf{G}}$, under the assumption that words in user context and POI context should have similar semantic properties on latent topics. However, when δ takes a large value, the performance stops increasing and decreases a bit, suggesting that some words may not have the same semantic properties due to their different appearance in the two contexts, while forcing them to be the same may decrease the performance.

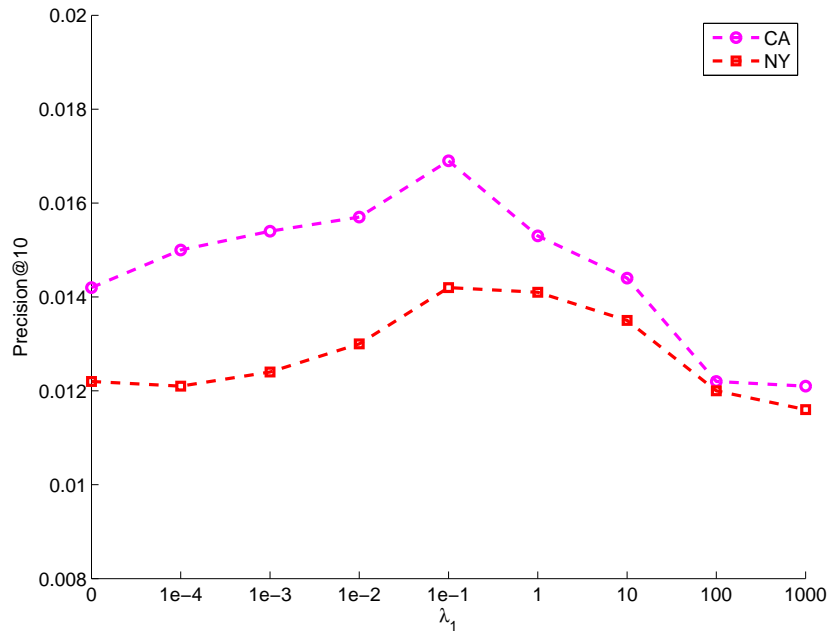


Figure 5.6: User-Interest Content- λ_1

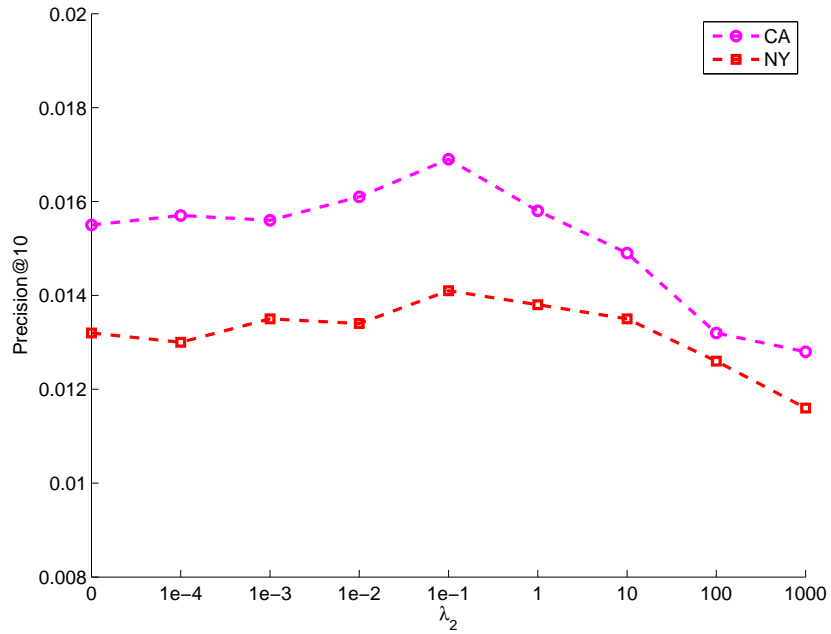


Figure 5.7: POI-Property Content- λ_2

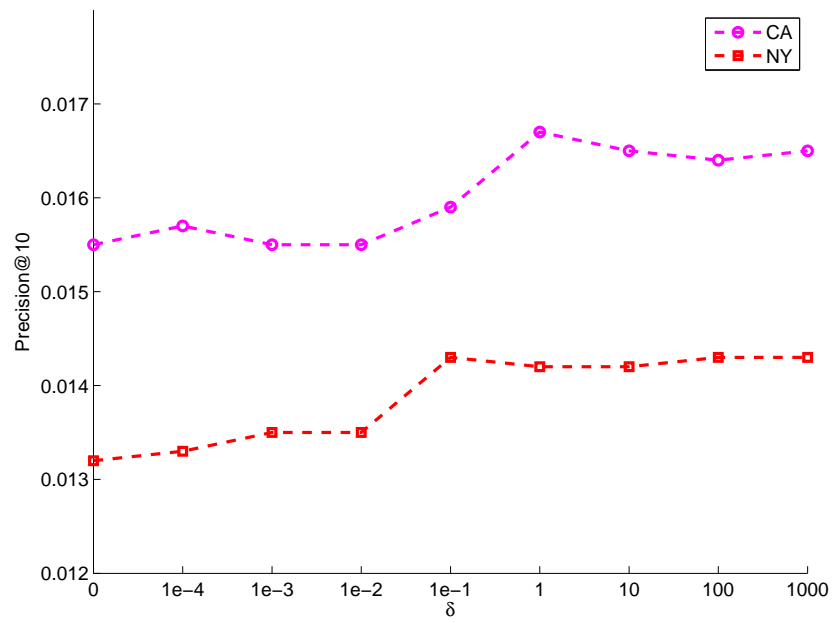


Figure 5.8: Semantic Overlapping- δ

CONCLUSION AND FUTURE WORK

In this dissertation, we study personalized POI recommendation on LBSNs. We investigate various LBSN-related properties to design the POI recommender systems, including geo-temporal patterns, geo-social correlations, and geo-content indications. We evaluate the performance of proposed models on a real-world LBSN dataset and summarize key observations from the experimental results.

In personalized geo-temporal POI recommender system, we study both temporal cyclic and temporal chronological patterns and their combinational effect. We leverage the temporal non-uniformness and temporal consecutiveness properties to model the temporal cyclic patterns with a matrix factorization model. We use power-law distribution and short-term effect to model the temporal chronological patterns with a Hierarchical Pitman-Yor language model.

In personalized geo-social POI recommender system, we consider the relationships between geographical distance and social friendships, and study them as a component w.r.t four elements, i.e., local friends, distant friends, local non-friends, and distant non-friends, which correspond to four types of correlations, i.e., local influence, distant influence, confounding effect, and unknown effect. We model them in a unify framework, gSCorr, and discover the recommendation ability of each element.

In personalized geo-content POI recommender system, we study the user-generated content and POI-associated content, and recognized three types of content information including sentiment indications, user interests, and POI properties. We model the sentiment as a enhanced component on check-in preference, and connect users' latent interests and POIs' latent profiles through the factorization of user-generated

content and POI-associated content.

There are many extensions and work that are worth further explorations. We summarize the future work as below:

- **Temporal-based Content Analysis**

Content information has been proven to be useful for personalized POI recommendation. By investigating the sentiment and topics embedded in content information, one can infer a user's interests in POIs and perform better recommendation services to him. However, a user's interests may change over the time. A user may not eat any spicy food before but love it now due to certain reasons. Such change is reflected through his check-in content over a certain period of time. Thus, temporal-based content analysis could help capture the change of check-in interests and provide the most up-to-date recommendations.

- **Relationships Among Multiple Information**

Although most of the existing work studies more than two types of information, e.g., spatio-temporal, socio-spatial, spatial-content, etc., individual information are commonly combined together through fused method, which restricts the understanding of their deep relationships. In the future, it is possible to study more coherent relationships among multiple types of information, such as the geo-social correlations. This also relies on the discovery of anthropology and social theories of these relationships, which can be helpful for guiding the relationship modeling.

- **Tensor-Based POI Recommender Systems**

Most of the existing POI recommender systems study temporal information with other types of information. When the temporal information is considered, it is natural to organize the check-in actions as a tensor. Thus, tensor-based ap-

proaches can be used to study user preferences, which is more compact and reasonable. Furthermore, tensor-based approaches consider different information as a whole component, providing us an opportunity to study their relationships and complementary effect for personalize POI recommendation.

- **Location-Based Mobile Applications**

Personalized POI recommendation has been a popular topic in academic during the last five years. Due to its close relationship to human mobility, it also exhibits great developing potential in industry. Location-based social networking services such as Foursquare and Yelp have already started to use their POI recommendation model to help users find interested POIs. We expect to see more applications, especially mobile applications, to be developed in the next decade and significantly facilitate users' daily life.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [2] D. Agarwal and B.-C. Chen. Recommender problems for web applications. In *16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Tutorial*, 2010.
- [3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 7–15. ACM, 2008.
- [4] C. Anderson. *The long tail: Why the future of business is selling less of more*. Hyperion Books, 2008.
- [5] A. Beeharee and A. Steed. Exploiting real world knowledge in ubiquitous applications. *Personal and Ubiquitous Computing*, 11(6):429–437, 2007.
- [6] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [7] F. Cairncross. *The death of distance: How the communications revolution is changing our lives*. Harvard Business Press, 2001.
- [8] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2):1009–1020, 2010.
- [9] J. Chang and E. Sun. Location 3: How users share and respond to location-based data on social networking sites. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.
- [10] C. Cheng, H. Yang, I. King, and M. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. *AAAI, Toronto, Canada*, 2012.
- [11] C. Cheng, H. Yang, M.R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2605–2611. AAAI Press, 2013.
- [12] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.
- [13] E. Cho, S. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090. ACM, 2011.

- [14] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.
- [15] C. Ding, T. Li, and M. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 183–192. IEEE, 2008.
- [16] Z. Y. Dingqi Yang, Daqing Zhang and Z. Wang. A sentiment-enhanced personalized location recommendation system. *ACM Hypertext*, 2013.
- [17] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.
- [18] T. Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.
- [19] H. Gao and H. Liu. Data analysis on location-based social networks. In *Mobile Social Networking*, pages 165–194. Springer, 2014.
- [20] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.
- [21] H. Gao, J. Tang, X. Hu, and H. Liu. Modeling temporal effects of human mobile behavior on location-based social networks. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1673–1678. ACM, 2013.
- [22] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, 2012.
- [23] H. Gao, J. Tang, and H. Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. *21st ACM International Conference on Information and Knowledge Management*, 2012.
- [24] H. Gao, J. Tang, and H. Liu. Mobile location prediction in spatio-temporal context. *Nokia Mobile Data Challenge Workshop*, 2012.
- [25] M. Gastner and M. Newman. The spatial structure of networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 49(2):247–252, 2006.
- [26] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 983–991. ACM, 2011.

- [27] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 899–908. ACM, 2010.
- [28] S. Goldwater, T. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power-law generators. *Advances in neural information processing systems*, 18:459, 2006.
- [29] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.
- [30] T. Horozov, N. Narasimhan, and V. Vasudevan. Using location for personalized poi recommendations in mobile environments. In *Applications and the Internet, 2006. SAINT 2006. International Symposium on*, pages 6–pp. IEEE, 2006.
- [31] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 25–32. ACM, 2013.
- [32] B. Hu and M. Ester. Social topic modeling for point-of-interest recommendation in location-based social networks. In *IEEE International Conference on Data Mining series*, 2014.
- [33] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.
- [34] S. W. Jorge Nocedal. *Numerical Optimization*. Springer, 1999.
- [35] E.-y. Kang, H. Kim, and J. Cho. Personalization method for tourist point of interest (poi) recommendation. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 392–400. Springer, 2006.
- [36] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.
- [37] J. Kulshrestha, F. Kooti, A. Nikraves, and K. Gummadi. Geographic dissection of the twitter network. 2012.
- [38] D. Lee, H. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [39] K. W.-T. Leung, D. L. Lee, and W.-C. Lee. Clr: a collaborative location recommendation framework based on co-clustering. In *SIGIR*, pages 305–314, 2011.

- [40] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 34. ACM, 2008.
- [41] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the CIKM*, pages 959–968. ACM, 2010.
- [42] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–11628, Aug. 2005.
- [43] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1051. ACM, 2013.
- [44] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. *Proc. of SDM13*, pages 396–404, 2013.
- [45] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $l_2, 1$ -norm minimization. In *Proceedings of the UAI*, pages 339–348. AUAI Press, 2009.
- [46] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the SIGKDD*, pages 323–332. ACM, 2010.
- [47] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 407–416. IEEE, 2011.
- [48] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *Selected Areas in Communications, IEEE Journal on*, 16(6):922–936, 1998.
- [49] X. Long and J. Joshi. A hits-based poi recommendation algorithm for location-based social networks. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 642–647. ACM, 2013.
- [50] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.
- [51] E. Malmi, T. Do, and D. Gatica-Perez. Checking in or checked in: Comparing large-scale manual and automatic location disclosure patterns. *The 11th International Conference on Mobile and Ubiquitous Multimedia (MUM 2012)*, 2012.

- [52] Y. Nesterov and I. E. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. 2004.
- [53] J. Nocedal and S. Wright. *Numerical optimization*. Springer verlag, 1999.
- [54] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *ICDM*, pages 1038–1043, 2012.
- [55] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. *Proceeding of the 5th International AAAI Conference on Weblogs and Social Media*, 2011.
- [56] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, (2), 1994.
- [57] R. Pan and M. Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the SIGKDD*, pages 667–676. ACM, 2009.
- [58] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 502–511. IEEE, 2008.
- [59] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [60] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust management*, pages 224–239. Springer, 2005.
- [61] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian users preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.
- [62] J. Pitman. *Combinatorial stochastic processes*, volume 1875. Springer-Verlag, 2006.
- [63] J. Pitman and M. Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.
- [64] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, volume 1, pages 2–1, 2007.
- [65] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.
- [66] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: Geo-social metrics for online social networks. In *Proceedings of the 3rd conference on Online social networks*, pages 8–8. USENIX Association, 2010.

- [67] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. *Proceeding of the 5th International AAAI Conference on Weblogs and Social Media*, 11, 2011.
- [68] R. Simon and P. Fröhlich. A mobile application framework for the geospatial web. In *Proceedings of the 16th international conference on World Wide Web*, pages 381–390. ACM, 2007.
- [69] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM 2004*, volume 2, pages 1414–1424. IEEE, 2004.
- [70] S. Sridharan. *Usability and Reliability of the User Action Framework: A Theoretical Foundation for Usability Engineering Activities*. PhD thesis, Citeseer, 2001.
- [71] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.
- [72] Y. Takeuchi and M. Sugimoto. Cityvoyager: an outdoor recommendation system based on user location history. In *Ubiquitous intelligence and computing*, pages 625–636. Springer, 2006.
- [73] J. Tang, H. Gao, and H. Liu. mtrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 93–102. ACM, 2012.
- [74] J. Tang, H. Gao, H. Liu, and A. Sarma. etrust: Understanding trust evolution in an online world. 2012.
- [75] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *Proceedings of the IJCAI*, pages 2712–2718. AAAI Press, 2013.
- [76] Y. Teh. A bayesian interpretation of interpolated kneser-ney. 2006.
- [77] Y. Teh. A hierarchical bayesian language model based on pitman-yor processes. In *ACL*, pages 985–992. Association for Computational Linguistics, 2006.
- [78] P. Venetis, H. Gonzalez, C. S. Jensen, and A. Halevy. Hyper-local, directions-based ranking of places. *Proceedings of the VLDB Endowment*, 4(5):290–301, 2011.
- [79] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. Barabási. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1100–1108. ACM, 2011.
- [80] H. Wang, M. Terrovitis, and N. Mamoulis. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 364–373. ACM, 2013.

- [81] X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing*, 2012.
- [82] J. Ye, Z. Zhu, and H. Cheng. Whats your next move: User activity prediction in location-based social networks. In *Proc. of the SIAM International Conference on Data Mining*. SIAM, 2013.
- [83] M. Ye, K. Janowicz, C. Mülligann, and W. Lee. What you are is when you are: the temporal dimension of feature types in location-based social networks. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 102–111. ACM, 2011.
- [84] M. Ye, X. Liu, and W. Lee. Exploring social influence for recommendation - a probabilistic generative approach. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2012.
- [85] M. Ye, P. Yin, and W. Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461. ACM, 2010.
- [86] M. Ye, P. Yin, W. Lee, and D. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2011.
- [87] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 1–10. IEEE, 2009.
- [88] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229. ACM, 2013.
- [89] J. Ying, E. Lu, W. Kuo, and V. Tseng. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 63–70. ACM, 2012.
- [90] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.
- [91] R. Zafarani, M. A. Abbasi, and H. Liu. *Social Media Mining: An Introduction*. Cambridge University Press, 2014.

- [92] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei. Evaluating geo-social influence in location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1442–1451. ACM, 2012.
- [93] J.-D. Zhang and C.-Y. Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 324–333. ACM, 2013.
- [94] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*, volume 10, pages 236–241, 2010.
- [95] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.
- [96] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Towards mobile intelligence: Learning from gps history data for collaborative recommendation. *Artificial Intelligence*, 184:17–37, 2012.
- [97] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):2, 2011.
- [98] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–40, 2010.
- [99] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1):5, 2011.
- [100] Y. Zheng, L. Zhang, X. Xie, and W. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800. ACM, 2009.
- [101] D. Zhou, B. Wang, S. Rahimi, and X. Wang. A study of recommending locations on location-based social network by collaborative filtering. *Advances in Artificial Intelligence*, pages 255–266, 2012.
- [102] K. Zickuhr. Three-quarters of smartphone owners use location-based services. *Pew Internet & American Life Project*, May 11, 2012.
- [103] G. Zipf. Selective studies and the principle of relative frequency in language (cambridge, mass, 1932). *Human Behavior and the Principle of Least-Effort (Cambridge, Mass, 1949)*, 1949.

BIOGRAPHICAL SKETCH

Huiji Gao is a Ph.D. candidate of Computer Science and Engineering at Arizona State University. He obtained his B.S. and M.S. at Beijing University of Posts and Telecommunications in 2007 and 2010, respectively. He was awarded the 2014 ASU Graduate Education Dissertation Fellowship, the 2014 ASU President's Award for Innovation, the 3rd Place Dedicated Task 2 Next Location Prediction of Nokia Mobile Data Challenge 2012, and various Student Travel Awards and Scholarships. His research interests include social computing, crowdsourcing for disaster management system, recommender systems, and mobile data mining on location-based social networks. He has published innovative works in highly ranked journals and top conference proceedings such as DMKD, IEEE Intelligent Systems, SIGKDD, CIKM, WWW, RecSys, WSDM, ICWSM, and ICDM. He has interned in IBM Research Almaden in 2013 and LinkedIn in 2014. Updated information can be found at <http://www.nini2yoyo.com>.