Point-of-interest Recommendation in Location-based Social Networks

ZHAO, Shenglin

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
November 2017

Thesis Assessment Committee

Professor CHAN Siu On(Chair)

Professor LYU Rung Tsong Michael (Thesis Supervisor)

Professor KING Kuo Chin Irwin (Thesis Co-supervisor)

Professor LEUNG Kwong Sak (Committee Member)

Professor CHEUNG Yiu Ming (External Examiner)

Abstract of thesis entitled:

Point-of-interest Recommendation in Location-based Social Networks

Submitted by ZHAO, Shenglin

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in November 2017

Location-based social networks (LBSNs) have become popular recently because of the explosive increase of smart phones that makes users easily to access to the LBSN Apps. More than 2.3 billion people worldwide use smart phones in 2017 predicted by EMarketer, which prospers the online LBSNs. A typical LBSN such as Foursquare collects users' check-in information including visited locations' geographical information (latitude and longitude) and users' comments at the location and allows users to make friends and share information as well. Driven by the collected big data in LBSNs, point-of-interest (POI) recommendation arises to improve the user experience in the App, which attempts to suggest each user a list of POIs that the user may feel interesting and be willing to visit in the future.

Developing POI recommendation systems requires analytics of the human mobility with respect to real-world POIs. Different from watching on Netflix or shopping on Amazon, checking-in at a POI in LBSNs is a physical activity, which causes the most important feature in POI recommendation: geographical influence. In addition, check-ins exhibit specific temporal characteristics.

For instance, users check-in at POIs around the office in the day time while at bars in the evening. These geographical and temporal features make the POI recommendation more challenging than traditional recommendation systems.

In this thesis, we systematically study the problem of POI recommendation in LBSNs. In particular, we review the literature in the area of POI recommendation, analyze the user mobility in LBSNs, and develop POI recommendation systems. First, we review state-of-the-art POI recommendation techniques and discover the challenges in POI recommendation systems. Second, we analyze the user mobility in LBSNs from geographical and temporal perspective respectively and show how to capture the geographical and temporal influence in a POI recommendation system. Third, we develop two POI recommendation systems: Geo-Teaser and STELLAR. Finally, we conclude this thesis and point out future work directions.

論文題目：位置社交網絡中的興趣點推薦
作者　　：趙胜林
學校　　：香港中文大學
學系　　：計算機科學與工程學系
修讀學位：哲學博士
摘要　　：

最近基於位置的社交網絡（LBSN）由於智能手機的爆炸式增長而變得流行起來。由 EMarketer 預測，在 2017 年全球超過 23 億人使用智能手機, 這使得用户可以輕松訪問手機應用程序而普及了基於位置的社交網絡。Foursquare 等典型的基於位置的社交網絡會收集用户的簽到信息，包括訪問地點的地理位置信息（緯度和經度）以及用户在該位置的評論，并允許用户交朋友和共享信息。在位置社交網絡收集到的大數據的驅動下，興趣點（POI）推薦應用應運而生以改善位置社交網絡應用中的用户體驗。興趣點推薦應用旨在向每個用户建議用户可能感興趣并願意在將來訪問的興趣點列表。

開發興趣點推薦系統需要對現實世界的興趣點進行人員移動性的分析。不同於在 Netflix 上觀影或在亞馬遜購物，在位置社交網絡中的興趣點進行簽到是一種親身體驗行为，這形成了興趣點推薦中最重要的特徵：地理影響。另外，簽到行为也表現出特別的時間特徵。例如，用户在白天常在辦公室附近的興趣點簽到, 而在晚上則是在酒吧活動。這些地理和時間特徵使興趣點推薦比傳統的推薦系統更具挑戰性。

在本論文中，我們系統的研究了位置社交網絡中的興趣點推薦問題。具體的，我們回顧了興趣點推薦領域的文獻，分析了位

置社交網絡中的用户移動性，并提出了原創的興趣點推薦系統。首先，我們綜述了主流的興趣點推薦技術，并分析興趣點推薦系統存在的挑戰。第二，我們分別從地理和時間角度分析位置社交網絡中的用户移動性，并展示如何在興趣點推薦系統中利用地理和時間影響提高系統表現。第三，我們開發了兩個興趣點推薦系統：Geo-Teaser 和 STELLAR。最後，我們總結本論文并指出未來的工作方向。

Acknowledgement

First and foremost, I would like to thank my supervisors, Prof. Michael R. Lyu and Prof. Irwin King. Without their supervision, I cannot finish my Ph.D. study at CUHK. Their inspiring guidance and patience on my research help me go through the tough Ph.D. period. More importantly, I benefit a lot from their rigorous requirements for the writing and presentation, not only on knowledge but also on attitude in doing research. In addition, I also want to thank Prof. Anthony Man-Cho So and Prof. Shiqian Ma, whose interesting and inspiring courses help much in my research.

I am grateful to my thesis assessment committee members, Prof. Kwong-Sak LEUNG, Prof. Siu-On Chan, and Prof. Yufei Tao, for their constructive comments and valuable suggestions to this thesis and all my term presentations. Great thanks to Prof. Yiu-ming CHEUNG from Hong Kong Baptist University who kindly serves as the external examiner for this thesis. Also, I want to thank my mentors when I intern in Huawei Noah's Ark Lab, Dr. Jia Zeng and Dr. Mingxuan Yuan, for the insightful discussions and happy time.

I would like to thank Tong Zhao, Haiqin Yang, Hongyi Zhang, Yu Kang, Xixian Chen, Guang Ling, Chen Cheng, for their contributions and suggestions for my research work in this thesis. I am also thankful to my other group fellows, Chao Zhou, Yuxin

Su, Qirun Zhang, Baichuan Li, Shouyuan Chen, Jieming Zhu, Zibin Zheng, Yangfan Zhou, Yilei Zhang, Cuiyun Gao, Hui Xu, Jichuan Zeng, Pinjia He, Jiani Zhang, Ken, Han Shao, Wang Chen, Yue Wang, Pengpeng Liu. In addition, I own the thanks to my research collaborators Prof. Qi Xie from Southwest University for Nationalities, Mr. Jiajun Cheng from National University of Defense Technology, Mr. Sheng Zhang and Prof. Jianguo Yao from Shanghai Jiaotong University. The collaborations broaden my view and deepen my understanding with the machine learning algorithms, beyond the area of recommendation systems.

Last but the most important, I would like to thank my dear family. Their deep love and constant support are the driving force when I feel frustrated in the research.

To my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents our research of POI recommendation in LBSNs, which is an important research field of search and recommendation for location-based services. We provide a brief overview of the research problem in Section 1.1, and highlight the main contributions of this thesis in Section 1.2. Section 1.3 outlines the thesis structure.

1.1 Overview

Location-based services play an important role in this Internet of Things (IoT) era. To monitor the status of devices connected to the Internet, analyze the collected data from different kinds of devices, and provide personalized services for device users, the location information of the device is indispensable for data analysis. For instance, the smart watch collects the location information of users and records their daily trajectories; and the map applications such as Google Maps in smart phones collect users' location information and guide users to anywhere in real-time. Figure 1.1 shows the location-based services in six aspects: search and recommendation, transportation, healthcare, public

Figure 1.1: Demonstration of location-based services

safety, game, environment monitoring, etc. Specifically, the location-based search and recommendation work for two kinds of applications: search engines such as Google and Baidu and LBSNs such as Yelp and Foursquare. In this work, we focus on the search and recommendation task in LBSNs.

LBSNs such as Foursquare and Facebook Places allow users to share their check-in behaviors, make friends, and write comments on visited locations, also called POIs [10, 106]. LBSNs are very popular now—for instance, Foursquare has attracted over 50 million people worldwide to use its service each month and recorded over 10 billion check-ins in total until Oct. 2017.[1] To improve user experience in LBSNs by suggesting favorite locations, a typical search and recommendation task namely POI recommendation [107, 9, 15, 111, 127, 126] comes out, which mines users' check-in sequences to recommend places where an individual may feel interested and be willing to check-

[1]https://foursquare.com/about

in in the future. The POI recommendation applications are of significance in two aspects: helping users explore new interesting places in a city and facilitating business owners to launch advertisements to the target customers.

Developing POI recommendation systems requires analyzing the human check-in activity in LBSNs. The check-in activity represents the user interactions with real-world POIs and exhibits specific geographical and temporal characteristics. From the geographical perspective, most of the check-ins happen in some constrained regions such as the district around the user's home or office. From the temporal perspective, the check-in activity also exhibits some specific patterns. For instance, users check-in at POIs around the office in the day time while bars in the evening. These unique features make the POI recommendation different from traditional recommendation systems. Hence, we need to comprehensively understand the human mobility in LBSNs and develop new algorithms for POI recommendation.

In this thesis, we study the POI recommendation in LBSNs. In particular, we review the literature in POI recommendation, analyze the user mobility in LBSNs, and develop POI recommendation systems. First, we survey state-of-the-art POI recommendation techniques and point out the challenges in POI recommendation systems. Second, we analyze the user mobility in LBSNs from geographical and temporal perspective respectively and show how to capture the geographical and temporal influence to enhance the POI recommendation system. Third, we develop a **Geo-te**mpor**a**l **s**equential **e**mbedding **r**ank (Geo-Teaser) model for POI Recommendation. Fourth, we develop a **s**patial-**te**mpora**l** **la**tent **r**anking (STELLAR) model for successive POI recommendation. Finally, we conclude this thesis and point out possible future work.

1.2 Thesis Contributions

In this thesis, we make contributions of POI recommendation in LBSNs in the following ways.

1. **Literature review of POI recommendation in LBSNs.** POI recommendation as a new application comes out with the popularity of LBSNs, which recommends users locations where they may feel interested and plan to check-in. POI recommendation helps users explore the city and find the useful services in LBSNs, and also helps the businesses discover target customers to launch advertisements. The significance of POI recommendation for users and businesses makes it attract much academic and industrial attention. In this chapter, we offer a systematic review of this field, summarizing the contributions of individual efforts and exploring their relations. We discuss the new properties and challenges in POI recommendation, comparing with traditional recommendation problems, e.g., movie recommendation. Then, we present a comprehensive review in two aspects: influential factors for POI recommendation and methodologies employed for POI recommendation. Moreover, we show the available datasets and the metrics.

2. **Understanding human mobility from geographical perspective.**
POI recommendation that suggests new locations for people to visit is an important application in LBSNs. Compared with traditional recommendation problems, e.g., movie

recommendation, geographical influence is a special feature that plays an important role in recommending POIs. In this chapter, we understand the user mobility in LBSNs from the geographical perspective and capture the geographical influence for POI recommendation. Our contributions are as follows. First, we propose Gaussian Mixture Model (GMM) to automatically learn users' activity centers via exploring their check-in history records. Moreover, we propose GA-GMM that employs a genetic algorithm based GMM to eliminate outliers. Finally, we conduct experiments on a real-world LBSN dataset and demonstrate that the proposed models capture the geographical information better and improve the accuracy of POI recommendation.

3. **Understanding human mobility from temporal perspective.**
   Understanding the user mobility from the temporal perspective is the key to POI recommendation. Because the user mobility in LBSNs exhibits strong temporal patterns—for instance, users would like to check-in at restaurants at noon and visit bars at night. Hence, capturing the temporal influence is necessary to ensure the high performance in a POI recommendation system. In this chapter, we understand the user mobility in LBSNs from the temporal perspective. Our contributions are four-fold: (1) To the best of our knowledge, this is the first temporal tensor factorization method for POI recommendation, subsuming all the three temporal properties: periodicity, consecutiveness, and non-uniformness. (2) We propose a novel model, Aggregated Temporal Tensor Factorization (ATTF), to capture temporal effect in POI recommendation at different

time scales. Experimental results show that our model outperforms prior temporal model more than 20%. (3) The proposed ATTF model is a general framework to capture the temporal features at different scales, which outperforms single temporal factor model and achieves 10% improvement in the top-5 POI recommendation task on Gowalla data. (4) We understand the ATTF model from the embedding neural network perspective, verifying the effectiveness of the embedding neural network that is a general framework for latent factor models, including rating estimation models (e.g., MF [34]) and ranking models (e.g., our ATTF model).

4. **Geo-Teaser system for POI recommendation.** POI recommendation is an important application for LBSNs, which learns the user preference and mobility pattern from check-in sequences to recommend POIs. To model the user preference, check-in sequential pattern, and the user spatial and temporal mobility pattern, in this chapter, we propose a **Geo**-**Te**mpor**a**l **s**equential **e**mbedding **r**ank (Geo-Teaser) model for POI recommendation. The contributions are three-fold: (1) We propose the temporal POI embedding model, which captures the check-ins' sequential contexts and the various temporal characteristics on different days. In particular, we introduce the word2vec framework to project every POI as one object in an embedding space for learning the sequential relations among POIs. Furthermore, we learn the temporal POI representations from the check-in sequence under some specific temporal state. (2) We propose a new way to incorporate the geographical influence into the pairwise preference ranking method

through discriminating the unvisited POIs according to geographical information. In particular, we define a hierarchical pairwise preference relation for each user check-in: the user prefers the visited POI than the unvisited neighboring POI, and the user prefers the unvisited neighboring POI than the unvisited non-neighboring POI. Then we learn the hierarchical pairwise preference to capture the geographical influence and user preference. (3) We propose the Geo-Teaser model as a unified framework combining the temporal POI embedding model and the geographically hierarchical pairwise preference ranking model. Experimental results on two real-life datasets show that the Geo-Teaser model outperforms state-of-the-art models. Compared with the best baseline competitor, the Geo-Teaser model improves at least 20% on both datasets for all metrics.

5. **STELLAR system for successive POI recommendation.**
   Successive POI recommendation in LBSNs becomes a significant task since it helps users to navigate a large number of candidate POIs and provide the best POI recommendations based on users' most recent check-in knowledge. However, all existing methods for successive POI recommendation only focus on modeling the correlation between POIs based on users' check-in sequences, but ignore an important fact that successive POI recommendation is a time-subtle recommendation task. To capture the impact of time on successive POI recommendation, in this chapter, we propose a **s**patial-**te**mpora**l** **la**tent **r**anking (STELLAR) model to explicitly formulate the interactions among user,

POI, and time. Our contributions are three-fold: (1) We propose a time-aware successive POI recommendation method—the STELLAR model, by considering the time information. In this model, we employ a new POI latent feature representation means to resolve the problem of coupled interaction. Experimental results demonstrate our STELLAR model outperforms state-of-the-art successive POI recommendation method. (2) We design a three-slice time indexing scheme to represent the timestamps, which captures the user check-ins specific characteristics: preference variance and periodicity. Experimental results show that our model better captures the temporal effect than state-of-the-art temporal models for POI recommendation. (3) We introduce a new interval-aware weight utility function to differentiate successive check-ins' correlations, which improves the successive POI recommendation accuracy.

1.3 Thesis Structure

The remainder of this thesis is organized as follows.

- **Chapter 2**
  In this chapter, we provide a systematic review in the area of POI recommendation. First, we report the problem definition of POI recommendation in Section 2.1. Next, we categorize the POI recommendation systems in two aspects: influential factors and methodology. More specifically, we propose two taxonomies to classify POI recommendation systems. In Section 2.2, we categorize the systems by the influential factors for check-in activities, including

the geographical information, temporal influence, social relationship, and content indications. In Section 2.3, we categorize the systems by the methodology, including systems modeled by fused methods and joint learning methods. For each category, we summarize the contributions and system features and highlight the representative work. Then, we introduce data sources and metrics for system performance evaluation in Section 2.4. Finally, we draw the conclusion of this chapter in Section 2.5.

- **Chapter 3**
  In this chapter, we understand the user mobility in LBSNs from the geographical perspective and capture the geographical influence for POI recommendation. In Section 3.2, we introduce the related work in three aspects: POI recommendation in LBSNs, geographical influence capturing methods, and GA-GMM. In Section 3.3, we demonstrate the two models: GMM and GA-GMM to capture the geographical influence for POI recommendation. In Section 3.4, we compare our proposed methods with state-of-the-art geographical models. Experimental results show that our proposed models better capture the geographical influence and perform better for POI recommendation. In Section 3.5, we summarize this chapter and draw the conclusion.

- **Chapter 4**
  In this chapter, we analyze the user mobility from the temporal perspective and aim to capture the temporal influence for the POI recommendation. In Section 4.2, we review the most relevant work. In Section 4.3, we

introduce our empirical analysis on check-in data and demonstrate the time labeling scheme. Next, in Section 4.4 we present the details of our proposed ATTF model and understand the proposed model from the neural network perspective. Then, in Section 4.5 we report experimental results conducted on two real-world datasets. Finally, in Section 4.6 we summarize this chapter and draw the conclusion.

- **Chapter 5**
  In this chapter, we propose the Geo-Teaser model for POI recommendation to capture the user preference, check-ins' sequential pattern, and the user spatial and temporal mobility pattern. In Section 5.2, we review the related work and summarize the connections of our model and prior work. In Section 5.3, we introduce two real-world datasets and report empirical data analysis that motivates our method. Next, we introduce our proposed Geo-Teaser model and show the learning algorithm in Section 5.4. Then, we evaluate our proposed model in Section 5.5. Finally, we conclude this chapter in Section 5.6.

- **Chapter 6**
  In this chapter, we propose the STELLAR system to resolve the time-aware successive POI recommendation problem. In Section 6.2, we review the most relevant work and summarize the connections of our model and prior work. In Section 6.3, we introduce the empirical analysis on check-in data to show the spatial and temporal properties. Next, we present the details of our model and show the learning procedures in Section 6.4. Then, in Section 6.5 we report

experimental results conducted on two real-world datasets. Finally, we conclude this chapter in Section 6.6.

- **Chapter 7**
  In this chapter, we summarize this thesis and point out the future work direction. In particular, we draw the conclusion of this thesis in Section 7.1. Then, in Section 7.2, we point out the future work direction in three aspects: ranking based model, online recommendation, and deep learning based recommendation.

□ **End of chapter.**

Chapter 2

Literature Review

POI recommendation comes out with the popularity of LBSNs, which suggests users locations where they may feel interested and plan to check-in in the future. In this chapter, we offer a systematic review of POI recommendation, summarizing the contributions of individual efforts and exploring their relations. First, we report the problem description and discuss the new properties and challenges in POI recommendation. Then, we present a comprehensive review in two aspects: influential factors for POI recommendation and methodologies employed for POI recommendation. Specifically, we propose two taxonomies to classify POI recommendation systems: 1) We categorize the systems by the influential factors for check-in activities, including the geographical information, social relationship, temporal influence, and content indications. 2) We categorize the systems by the methodology, including systems modeled by fused methods and joint learning methods. For each category, we summarize the contributions and system features and highlight the representative work. Moreover, we discuss the available datasets and the popular metrics. Finally, we conclude this literature review.

2.1 Problem Description

POI recommendation aims to mine users' check-in records and recommend POIs for users in LBSNs. Formally, we define two important terms, i.e., check-in and check-in sequence, as follows.

Definition 1 (Check-in) *A check-in is denoted as a triple* $\langle u, l, t\rangle$ *that depicts a user* u *visiting POI* l *at time* t*.*

Definition 2 (Check-in sequence) *A check-in sequence is a set of check-ins of user* u*, denoted as* $S_u = \{\langle l_1, t_1\rangle, \cdots, \langle l_n, t_n\rangle\}$*, where* t_i *is the check-in timestamp. For simplicity, we denote* $S_u = \{l_1, \cdots, l_n\}$*.*

POI recommendation recommends a user a list of POIs via mining the check-in records. Given Definition 1 and Definition 2, the problem of POI recommendation can be defined as follows.

Definition 3 (POI recommendation) *Given all users' check-in sequences* S*, POI recommendation aims to recommend a POI list* S_N *to each user* u*. Here* S *is a collected check-in sequence set, contain all sequences* S_u *for all users.*

POI recommendation is a branch of recommendation systems, which encourages to address this task through borrowing ideas from conventional recommendation systems such as movie recommendation. Hence, we suffice to make use of conventional recommendation system techniques to recommend POIs, e.g., collaborative filtering (CF) methods. However, the special scenario in LBSNs that the location bridges the physical world and the online networking services, arouses new challenges to the traditional recommendation system techniques. Take Foursquare as an example, Figure 2.1 demonstrates how the

Figure 2.1: Demonstration of check-in information in Foursquare

check-in information is recorded, including user name, POI, check-in timestamp, and geographical information in the map. After introducing the location, some new challenges appear, which can be summarized as follows [123].

1. Physical constraints. The check-in activity is limited by physical constraints, compared with shopping online from Amazon and watching a movie on Netflix. For one thing, users in LBSNs check-in at geographically constrained areas. As observed in [11, 10], users usually check-in at POIs around their homes and offices, and there are a few check-ins out of their cities. For another, shops regularly provide services at some limited time. For instance, most of coffee shops open during day time but close at night. Such physical constraints make the check-in activity in LBSNs exhibit significantly spatial and temporal properties [2, 10, 18, 19, 76, 104, 110, 125, 124].

2. Extreme sparseness. A typical user in LBSNs such as

Foursquare contains hundreds of check-ins each year. But there are millions of POIs in the LBSNs. Compared with traditional movie recommendation, POI check-in data are much sparser. So it is difficult to suggest top N (usually five or ten) POIs from millions of candidates.

3. Complex relations. The location sharing activities in the online social media alter original social relations since people are apt to make new friends with geographical neighbors [83, 84]. Moreover, for online social media services such as Twitter and Facebook, the location for geo-tagging yields new relations between locations and locations [112], and as well between users and locations [17, 82, 108].

4. Heterogeneous information. LNSNs consist of different kinds of information, including not only check-in records, the geographical information of locations, and venue descriptions but also users' social relation information and media information (e.g., user comments and tweets). The heterogeneous information depicts the user activity from a variety of perspectives [98, 97, 115], inspiring POI recommendation systems of different kinds [47, 54, 46, 65, 81, 96, 114].

2.2 Taxonomy by Influential Factors

We categorize the studies in POI recommendation according to several influential factors upon the user check-in activity. Because of the spatial and temporal properties resulted from the physical constraints and heterogeneous information such as locations' geographical information and users' comments, the check-in activity is a synthesized decision from a variety of factors. Figure 2.2 shows four main factors in POI recommendations: temporal dynamics, geographical influence, social relations, and content indications. In the following, we demonstrate how each factor affects the check-in activity and how to model each influential factor for POI recommendation.

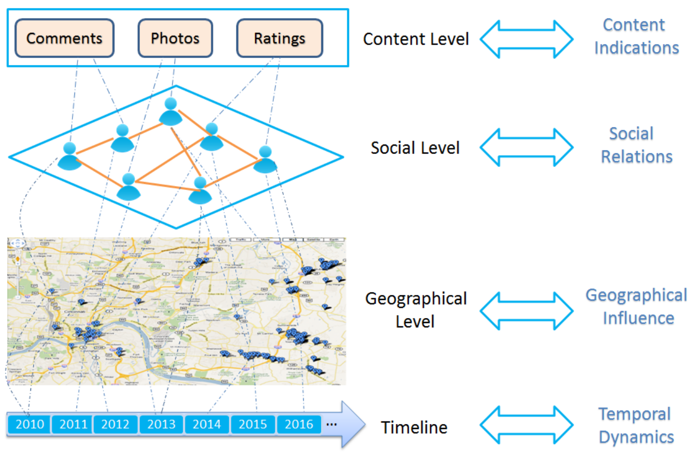


Figure 2.2: Influential factors in LBSNs

2.2.1 Geographical Influence

Geographical influence is an important factor that distinguishes the POI recommendation from traditional item recommendation because the check-in behavior depends on locations' geographical features. Analysis on users' check-in data shows that a user acts in geographically constrained areas and prefers to visiting POIs nearby where the user has checked-in. The feature of geographical constraints can shrink the POI candidate set and alleviate the effect of data sparsity. Several studies [7, 44, 54, 107, 113, 116, 117, 122] attempt to employ the geographical influence to improve POI recommendation systems. In particular, three representative models, i.e., power law distribution model, Gaussian distribution model, and kernel density estimation model, are proposed to capture the geographical influence in POI recommendation.

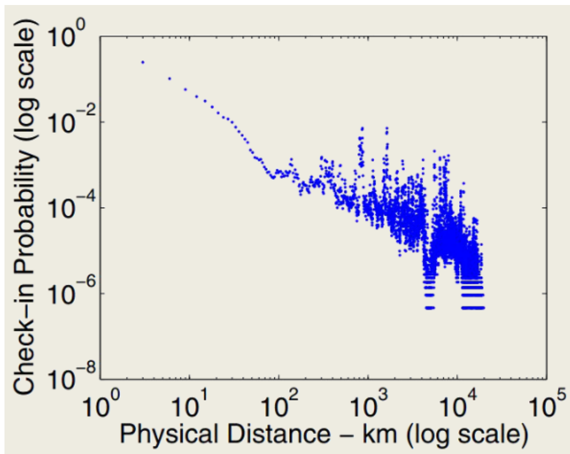


Figure 2.3: Power law distribution pattern [107]

In [107], Ye et al. employ a power law distribution model to capture the geographical influence. Power law distribution pattern has been observed in human mobility such as withdraw activities in ATMs and travel in different cities [3, 21, 76]. Also, Ye et al. discover a similar pattern of users' check-in activity in LBSNs [106, 107]. Figure 2.3 demonstrates two POIs' co-occurrence probability distribution over the distance between two POIs. Because of the power law distribution in Figure 2.3, we are able to model the geographical influence as follows. The co-occurrence probability y of two POIs by the same user can be formulated as follows,

$$y = a * x^b, \tag{2.1}$$

where x denotes the distance between two POIs, a and b are parameters of the power-law distribution. Here, a and b

should be learned from the observed check-in data, depicting the geographical feature of the check-in activity. A standard way to learn the parameters, a and b, is to transform Eq. (2.1) to a linear equation via a logarithmic operation, and learn the parameters by fitting a linear regression problem.

On the basis of the geographical influence model depicted through the power law distribution, new POIs can be suggested according to the following formula. Given a checked-in POI set L_i, the probability $Pr(l_j|L_i)$ of visiting POI l_j for user u_i, is formulated as,

$$Pr(l_j|L_i) = \frac{Pr(l_j \cup L_i)}{Pr(L_i)} = \prod_{l_y \in L_i} Pr(d(l_j, l_y)), \tag{2.2}$$

where $d(l_j, l_y)$ denotes the distance between POI l_j and l_y, and $Pr(d(l_j, l_y)) = a * d(l_j, l_y)^b$. In [106, 107], Ye et al. leverage the power law distribution to model the geographical influence and combine it with collaborative filtering techniques [77] to recommend POIs. In addition, Yuan et al. [113] also adopt the power law distribution model, but learn the parameter using a Bayesian rule instead.

The second type to model the geographical influence is a series of Gaussian distribution based methods. Cho et al. [11] observe that users in LBSNs always act around some activity centers, e.g., home and office, as shown in Figure 2.4. Further, Cheng et al. [7] propose a Multi-center Gaussian Model (MGM) to capture the geographical influence for POI recommendation. Given the multi-center set C_u, the probability of visiting POI l by user u is defined by

$$P(l|C_u) = \sum_{c_u=1}^{|C_u|} P(l \in c_u) \frac{f_{c_u}^{\alpha}}{\sum_{i \in C_u} f_i^{\alpha}} \frac{N(l|\mu_{C_u}, \sum_{C_u})}{\sum_{i \in C_u} N(l|\mu_i, \sum_i)}, \tag{2.3}$$

Figure 2.4: Check-in distribution in multi-centers [11]

where $P(l \in c_u) \propto \frac{1}{d(l,c_u)}$ is the probability of the POI l belonging to the center c_u, $\frac{f_{c_u}^{\alpha}}{\sum_{i \in C_u} f_i^{\alpha}}$ denotes the normalized effect of the check-in frequency on the center c_u and parameter α maintains the frequency aversion property, $N(l|\mu_{C_u}, \sum_{C_u})$ is the probability density function of Gaussian distribution with mean μ_{C_u} and covariance matrix $\sum_{C_u}$. Specifically, the MGM employs a greedy clustering algorithm on the check-in data to find the user activity centers, which may result in the unbalanced assignment of POIs to different activity centers. Hence, Zhao et al. [122] propose a genetic-based Gaussian mixture model to capture the geographical influence, which outperforms the MGM in POI recommendation.

The third type of geographical model is the kernel density estimation (KDE) model. In order to mine the personalized geographical influence, Zhang et al. [116] argue that the geographical influence on each individual user should be personalized rather than modeling through a common distribution, e.g., power law distribution [107] and MGM [7]. As shown

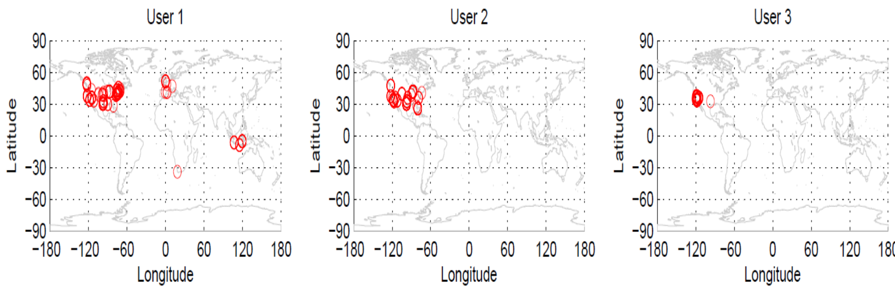


Figure 2.5: Distributions of personal check-in locations [116]

in Figure 2.5, it is hard to model different users using the same distribution. To this end, they leverage kernel density estimation [85] to model the geographical influence using a personalized distance distribution for each user. Specifically, the kernel density estimation model consists of two steps: distance sample collection and distance distribution estimation. The step of distance sample collection generates a sample X_u for a user by computing the distance between every pair of locations visited by the user. Then, the distance distribution can be estimated through the probability density function f over distance d,

$$f(d) = \frac{1}{|X_u|\sigma} \sum_{d' \in X_u} K(\frac{d-d'}{\sigma}), \tag{2.4}$$

where σ is a smoothing parameter, called the bandwidth. $K(\cdot)$ is the Gaussian kernel

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \tag{2.5}$$

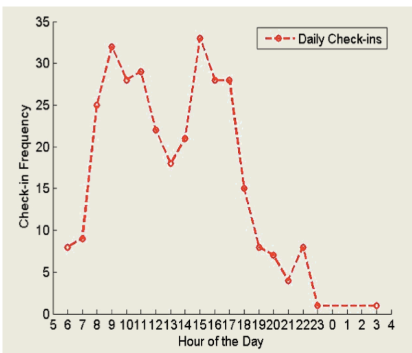
Denote $L_u = \{l_1, l_2, \ldots, l_n\}$ as the visited locations of user u. The probability of user u visiting a new POI l_j given the checked-in POI set L_u is defined as,

$$p(l_j|L_u) = \frac{1}{|L_u|} \sum_{l_i \in L_u} f(d_{ij}), \tag{2.6}$$

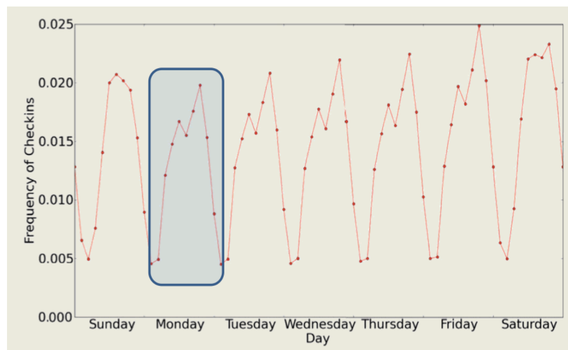
where d_{ij} is the distance between l_i and l_j, $f(\cdot)$ is the distance distribution function in Eq. (2.4).

2.2.2 Temporal Influence

Temporal influence is of vital importance for POI recommendation because physical constraints on the check-in activity result in specific patterns. Temporal influence in a POI recommendation system performs in three aspects: periodicity, consecutiveness, and non-uniformness.



(a) Day pattern

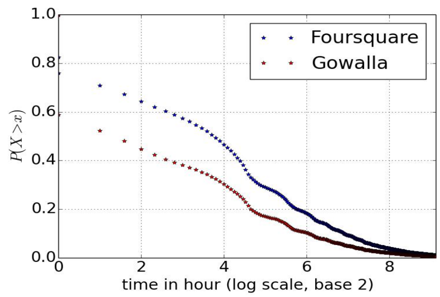


(b) Week pattern

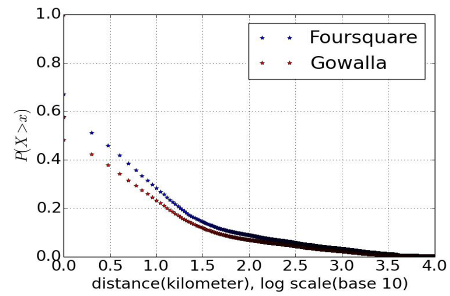
Figure 2.6: Periodic pattern [10]

Users' check-in behaviors in LBSNs exhibit the periodic pattern. For instance, users always visit restaurants at noon and have fun in nightclubs at night. Also, users visit places around the office on weekdays and spend time in shopping malls on weekends. Figure 2.6 shows the periodic pattern in a day and a week, respectively. The check-in activity exhibits this kind periodic pattern—visiting the same or similar POIs at the same time slot. This observation inspires the studies exploiting this periodic pattern for POI recommendation [11, 15, 113, 118].

Consecutiveness performs in the check-in sequences, especially in the successive check-ins. Successive check-ins are usually



(a) CCDF of intervals in successive check-ins



(b) CCDF of distances in successive check-ins

Figure 2.7: Consecutive pattern [128]

correlated. For instance, users may have fun in a nightclub after dining in a restaurant. This frequent check-in pattern implies that the nightclub and the restaurant are geographically adjacent and correlated from the perspective of venue function. Data analysis on Foursquare and Gowalla in [128] explores the spatial and temporal property of successive check-ins in Figure 2.7, namely the complementary cumulative distributive function (CCDF) of intervals and distances between successive check-ins. It is observed that many successive check-ins are highly correlated: over 40% and 60% successive check-in behaviors happen in less than 4 hours in Foursquare and Gowalla respectively; about 90% successive check-ins happen in less than 32 kilometers (half an hour driving distance) in Foursquare and Gowalla. Researchers exploit the Markov chain model to capture the sequential pattern [9, 14, 25, 119]. Studies in [9, 14] assume that two successive checked-in POIs in a short term are highly correlated and employ the factorized personalized Markov chain (FPMC) model [74] to recommend successive POIs. Zhang et al. [119] propose an additive Markov model to learn the transitive probability between two successive check-ins. Zhao et al. [128] exploit a spatial temporal latent ranking model for POI recommendation, which captures the consecutiveness by a POI-POI latent interaction similar to FPMC model.

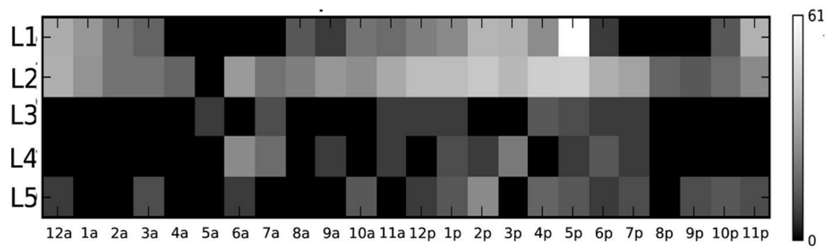


Figure 2.8: Demonstration of non-uniformness [18]

The non-uniformness feature depicts a user's check-in preference variance at different hours of a day, or at different months of a year, or at different days of a week [15]. As shown in Figure 2.8, the study in [15] demonstrates an example of a random user's aggregated check-in activities on the user's top five most visited POIs. It is observed that a user's check-in preference changes at different hours of a day—the most frequent checked-in POI alters at different hours. Similar temporal characteristics also appear at different months of a year, and different days of a week as well. This non-uniformness feature can be explained from the user's daily life customs: 1) A user may check-in at POIs around the user's home in the morning hours, visit places around the office in the day hours, and have fun in bars during night hours. 2) A user may visit more locations around the user's home or office on weekdays. On weekends, the user may check-in more at shopping malls or vacation places. 3) At different months, a user may have different hobbies for food and entertainment. For instance, a user would visit ice cream shops in the months of summer while visit hot pot restaurants in the months of winter.

2.2.3 Social Influence

Inspired by the assumption that friends in LBSNs share more common interests than non-friends, social influence is explored to enhance POI recommendation [7, 17, 18, 20, 106, 103, 117, 119]. In fact, employing social influence to enhance recommendation systems has been explored in traditional recommendation systems, both in memory-based methods [31, 58] and model-based methods [32, 56, 57]. Researchers borrow the ideas from traditional recommendation systems to POI recommendation. In the following, we demonstrate representative studies capturing social influence in two aspects: memory-based and model-based.

Ye et al. [106] propose a memory-based model, friend-based collaborative filtering (FCF) approach for POI recommendation. FCF model constrains the user-based collaborative filtering to find top similar users in friends rather than all users of LBSNs. Hence, the preference r_{ij} of user u_i at l_j is calculated as follows,

$$r_{ij} = \frac{\sum_{u_k \in F_i} r_{kj} w_{ik}}{\sum_{u_k \in F_i} w_{ik}}, \tag{2.7}$$

where F_i is the set of friends with top-n similarity, w_{ik} is similarity weight between u_i and u_k. FCF enhances the efficiency by reducing the computation cost of finding top similar users. However, it overlooks the non-friends who share many common check-ins with the target user. Experimental results show that FCF brings very limited improvements over user-based POI recommendation in terms of precision.

Cheng et al. [7] apply the probabilistic matrix factorization with social regularization (PMFSR) [57] in POI recommendation, which integrates social influence into PMF [80]. Denote $\mathcal{U}$ and

$\mathcal{L}$ are the set of users and POIs, respectively. PMFSR learns the latent features of users and POIs by minimizing the following objective function,

$$\arg\min_{U,L} \sum_{i=1}^{|\mathcal{U}|} \sum_{j=1}^{|\mathcal{L}|} I_{ij}(g(c_{ij}) - g(U_i^T L_j))^2 + \lambda_1 ||U||_F^2 + \lambda_2 ||L||_F^2 + \beta \sum_{i=1}^{|\mathcal{U}|} \sum_{u_f \in F_i} sim(i,f) ||U_i - U_f||_F^2, \tag{2.8}$$

where c_{ij} is the check-in frequency, U_i, U_f, and L_j are the latent features of user u_i, u_f, and POI l_j respectively, I_{ij} is an indicator denoting user u_i has checked-in at POI l_j, F_i is the set of user u_i's friends, $sim(i,f)$ denotes the social weight between user u_i and u_f, and $g(\cdot)$ is the sigmoid function to mapping the target value into the range of [0,1]. In this framework, the social influence is incorporated by the social constraints that ensure latent features of friends keep in close at the latent subspace. Due to its validity, Yang et al. [103] also employ the same framework to their sentiment-aware POI recommendation.

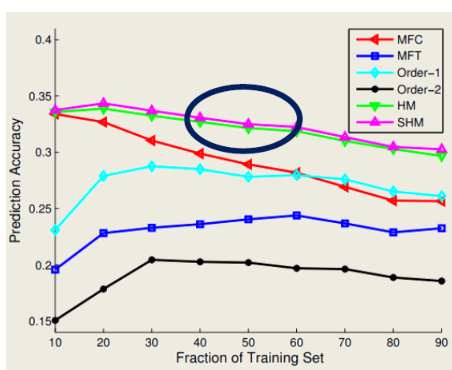


Figure 2.9: The significance of social influence on POI recommendation [18]

Although social influence improves traditional recommendation system significantly [32, 56, 57], the social influence on POI recommendation shows limited improvements [7, 18, 106]. Figure 2.9 shows the limited improvement achieved from social influence in [18]. Why this happens can be explained as follows. Users in LBSNs make friends online without any limitation; on the contrary, the check-in activity requires physical interactions between users and POIs. Hence, friends in LBSNs may share the common interest but may not visit common locations. For instance, friends in favor of Italian food from different cities will visit their own local Italian food restaurants. This phenomenon differs from the online movie and music recommendation scenarios in Netflix and Spotify.

2.2.4 Content Indications

In LBSNs, users generate contents including tips and photos about the POIs. Although contents do not accompany each check-in record, the available contents such as the user comments and photos, can be used to enhance the POI recommendation [16, 29, 43, 103, 109, 95]. On the one hand, user comments provide extra information from the shared tips beyond the check-in behavior, e.g., the preference on a location. For instance, the check-in at an Italian restaurant does not necessarily mean the user likes this restaurant. Probably the user just likes Italian food but not this restaurant, even dislikes the taste of this restaurant. Compared with the check-in activity, the comments usually provide explicit preference information, which is a kind of complementary explanations for the check-in behavior. As a result, the comments are able to be used to deeply understand the users' check-in behavior and improve POI recommendation [16, 29, 103]. On the other hand, photos about POIs also reveal users' check-in preference. For example, a user who posts many architecture photos is more likely to visit famous landmarks; while a user posts lots of images about food has more incentive to visit restaurants. Thus, images have potentials to improve the performance of POI recommendation. In the following, we report two representative studies that exploit comments and photos to enhance the POI recommendation respectively.

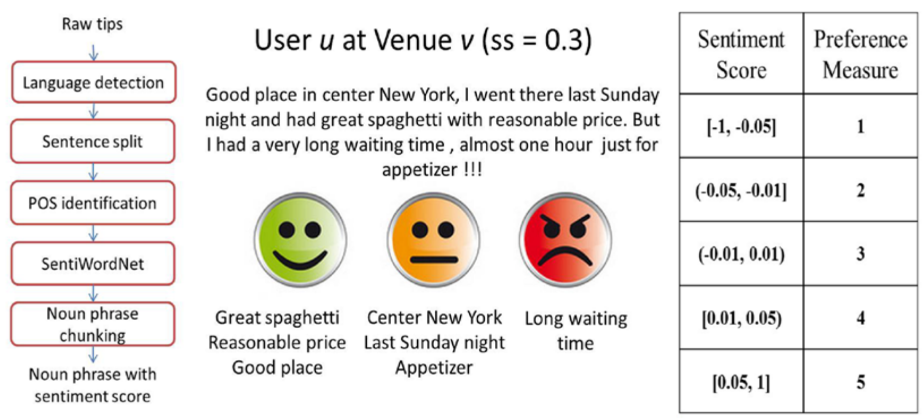


Figure 2.10: Sentiment-preference transforming rule [103]

The research in [103] is the first and representative work exploiting the comments to strengthen the POI recommendation. Yang et al. [103] propose a sentiment-enhanced location recommendation method, which utilizes the user comments to adjust the check-in preference estimation. As shown in Figure 2.10, the raw tips in LBSNs are collected and analyzed using natural language processing techniques, including language detection, sentence split, POS identification, processed by SentiWordNet, and Noun phrase chunking. Then, each comment is given a sentiment score. According to the estimated sentiment, a preference score of one user at a POI is generated. Figure 2.10 also shows how to handle a comment example: transforming it to several noun phrases such as "Reasonable price", "Good place", and "Long waiting time", generating a sentiment score of 0.3, and mapping this value to the preference measure of 5. Moreover, through combining the preference measure from sentiment analysis and the check-in frequency, the proposed model in [103] generates a modified rating $\hat{C}_{i,j}$ measuring the preference of user u_i at a POI l_j. Accordingly, the traditional matrix factorization method can be employed to recommend POIs through the following objective,

$$\arg\min_{U,L} \sum_{(i,j)\in\Omega} (\hat{C}_{i,j} - U_i L_j^T)^2 + \alpha||U||_F^2 + \beta||L||_F^2, \tag{2.9}$$

where U_i and L_j are latent features of user u_i and l_j respectively, $\hat{C}_{i,j}$ is the combined rating value, α and β are regularizations.
The research in [95] is the first and representative work exploiting the photos to strengthen the POI recommendation. Let $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{P}$ be the set of users, POIs, and photos, respectively. Furthermore, $\mathbf{X} \in R^{|\mathcal{U}|\times|\mathcal{L}|}$ denotes the user-POI check-in matrix, where each entry means the check-in frequency. Next, $\mathbf{G} \in R^{|\mathcal{U}|\times|\mathcal{L}|}$ denotes the normalized version of $\mathbf{X}$ with $\mathbf{G}_{ij} = g(\mathbf{X}_{ij})$ and $g(\cdot)$ is the sigmoid function. $\mathcal{P}_{u_i}$ denotes the set of photos uploaded by user u_i and $\mathcal{P}_{l_j}$ denotes the set of phones tagged to POI l_j. Hence, the image enhanced POI recommendation aims to recommend each user top k unvisited POIs, given the check-in matrix $\mathbf{G}$, user images $\mathcal{P}_u$ for all users, and POI images $\mathcal{P}_l$ for all POIs. This visual feature enhanced POI recommendation system can be learned by maximizing the likelihood defined on $\mathbf{G}$, $\mathcal{P}_u$, and $\mathcal{P}_l$.
To learn the maximum likelihood on $\mathcal{P}_u$, $P(f_{is} = 1|u_i, p_s)$ is defined to measure the probability that a photo p_s belongs to a user u_i. Considering the image p_s posted by u_i, it is natural to

assume that p_s contains certain visual contents that meet u_i's preferences; while for an arbitrary image p_w posted by other users, i.e., $p_w \notin \mathcal{P}_{u_i}$, p_w is less likely to contain visual contents that meet u_i's preferences. Meantime, u_i's preferences are captured by the latent feature $\mathbf{u}_i \in R^K$. Then, the probability $P(f_{is} = 1|u_i, p_s)$ is defined via a softmax function,

$$P(f_{is} = 1|u_i, p_s) = \frac{\exp(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_s))}{\sum_{p_k \in \mathcal{P}} \exp(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_k))} \tag{2.10}$$

where f_{is} denotes if p_s is posted by u_i, $CNN(p_s)$ is the extracted feature from the image p_s via CNN (implemented by VGG-16 [86]), $\mathbf{P} \in R^{K \times d}$ is the interaction matrix between the visual contents and latent user features, and d is the dimension of the visual contents.

Similarly, the probability $P(g_{jt} = 1|l_j, p_t)$ is defined to measure whether the image p_t is tagged to the POI l_j for learning the maximum likelihood on $\mathcal{P}_l$. Considering an image p_t associated with POI l_j, $CNN(P_t)$ denotes the visual feature of image p_t, and $\mathbf{l}_j \in R^K$ denotes the latent feature of POI l_j. Hence, the probability of $P(g_{jt} = 1|l_j, p_t)$ is defined as follows,

$$P(g_{jt} = 1|l_j, p_t) = \frac{\exp(\mathbf{l}_j^T \cdot \mathbf{Q} \cdot CNN(p_t))}{\sum_{p_k \in \mathcal{P}} \exp(\mathbf{l}_j^T \cdot \mathbf{Q} \cdot CNN(p_k))}, \tag{2.11}$$

where g_{jt} denotes if photo p_t is tagged to POI l_j, $\mathbf{Q} \in R^{K \times d}$ is the interaction matrix between the visual contents and latent POI features, and d is the dimension of the visual contents.

Given the check-in matrix $\mathbf{G}$, user images $\mathcal{P}_u$, and POI images $\mathcal{P}_l$, the visual feature enhanced POI recommendation framework is defined via maximizing the following logarithmic posterior distribution,

$$\max_{\mathbf{U},\mathbf{L},\mathbf{P},\mathbf{Q}} \log P(\mathbf{U}, \mathbf{L}, \mathbf{P}, \mathbf{Q}|\mathbf{G}, \mathcal{P}_u, \mathcal{P}_l), \tag{2.12}$$

where $\mathbf{U}, \mathbf{L}, \mathbf{P}, \mathbf{Q}$ are user latent feature matrix, POI latent feature matrix, user-photo feature interaction matrix, and POI-photo feature interaction matrix, respectively. Furthermore, based on $P(f_{is} = 1|u_i, p_s)$ and $P(g_{jt} = 1|l_j, p_t)$ this posterior distribution can be learned through the maximum likelihood over $\mathbf{G}$, $\mathcal{P}_u$, and $\mathcal{P}_l$ with regularizations,

$$\max_{\mathbf{U},\mathbf{L},\mathbf{P},\mathbf{Q}} \alpha(\sum_{i=1}^{|\mathcal{U}|} \sum_{p_k \in \mathcal{P}_{u_i}} \log P(f_{ik} = 1|u_i, p_k) + \sum_{j=1}^{|\mathcal{L}|} \sum_{p_k \in \mathcal{P}_{l_j}} \log P(g_{jk} = 1|l_j, p_k))$$
$$-||\mathbf{Y} \odot (\mathbf{G} - \mathbf{U}^T\mathbf{L}||_F^2 - \lambda_1(||\mathbf{U}||_F^2 + ||\mathbf{L}||_F^2) - \lambda_2(||\mathbf{P}||_F^2 + ||\mathbf{Q}||_F^2), \tag{2.13}$$

where $\mathbf{Y}$ is the indicator matrix that constrains the calculation only valid for the non-zero entries in $\mathbf{G}$, $\odot$ is the Hadamard product, α is the hyperparamter to constrain the effect of visual modeling, and λ_1 and λ_2 are regularizations to avoid overfitting. After learning the objective function Eq. (2.13), the top k POIs can be selected according to the value of $\mathbf{U}^T\mathbf{L}$ that measures the user check-in preference on POIs.

2.2.5 Summary

In this section, we show how the four influential factors, geographical influence, temporal influence, social influence, and content indications contribute to the POI recommendation and how the deliveries incorporate them. In addition, the check-in activity implies the user preference, which can be modeled using the collaborative filtering methods. In the following, we summarize the existing POI recommendation systems and demonstrate how the influential factors contribute to POI recommendation in Table 2.1.

Table 2.1: Summary of POI recommendation systems modeling for influential factors

	User Preference	**Geographical Influence**	**Temporal Influence**	**Social Influence**	**Content Indications**
[7]	√	√		√	
[107]	√	√		√	
[103]	√			√	√
[44]	√	√			
[128]	√	√	√		
[126]	√		√		
[113]	√	√	√		
[116]	√	√		√	
[15]	√		√		
[16]	√				√
[14]	√	√			
[121]	√	√	√		
[95]	√				√

2.3 Taxonomy by Methodology

In this section, we categorize the POI recommendation systems by the methodologies. There are two ways to construct a POI recommendation system: the fused model and the joint model, which are categorized by the way of modeling the influential factors discussed in Section 2.2. The fused model fuses recommended results from collaborative filtering method and recommended results from models capturing geographical influence, social influence, and temporal influence. The joint model establishes a joint model to learn the user preference and the influential factors together.

2.3.1 Fused Model

The fused model establishes a model for each influential factor and combines their recommended results with suggestions from the collaborative filtering model [77] that captures user preference on POIs. Since social influence provides limited improvements in POI recommendation and user comments are usually missing in users' check-ins, geographical influence and temporal influence constitute two important factors for POI recommendation. Hence, a typically fused model [7, 107, 117] recommends POIs through combining the traditional collaborative filtering methods and influential factors, especially including geographical influence or temporal influence. Using collaborative filtering methods to capture the user preference can be categorized into two types: memory-based method (e.g., user-based) and model-based method (matrix factorization). In the following, we introduce the representative work for fused model in two aspects: memory-based and model-based.

Representative Work for Memory-based Method

In [107], Ye at al. propose a fused framework for POI recommendation, which captures the user preference, social influence, and geographical influence. Specifically, Ye et al. [107] use the user-based collaborative filtering to model the user preference, friend-based collaborative filtering for social influence, power law distribution model for geographical model. Let $S_{i,j}$ denote the check-in probability score of user u_i at POI l_j. $S^u_{i,j}$, $S^s_{i,j}$, and $S^g_{i,j}$ denote the check-in probability scores of user u_i at POI l_j, corresponding to recommendation results based on user preference, social influence, and geographical influence, respectively. Then, the fused recommendation result is formulated as,

$$S_{i,j} = (1 - \alpha - \beta) S^u_{i,j} + \alpha S^s_{i,j} + \beta S^g_{i,j}, \tag{2.14}$$

where the two weighting parameters α and β ($0 \leq \alpha + \beta \leq 1$) denote the relative importance of social influence and geographical influence comparing to user preference.

Specifically, $S^u_{i,j}$, $S^s_{i,j}$, and $S^g_{i,j}$ are obtained from the check-in probability $p^u_{i,j}$, $p^s_{i,j}$, and $p^g_{i,j}$ for a user u_i to visit a POI l_j. $p^s_{i,j}$, and $p^g_{i,j}$ can be calculated using Eq. (2.7) described in Section 2.2.3 and Eq. (2.2) described in Section 2.2.1, respectively. $p^u_{i,j}$ is calculated through the user-based CF model,

$$p^u_{i,j} = \frac{\sum_{u_k} w_{i,k} \cdot c_{k,j}}{\sum_{u_k} w_{i,k}}, \tag{2.15}$$

where $c_{k,j}$ denotes the check-in frequency of user u_k at POI l_j, $w_{i,k}$ is the similarity weight between user u_i and u_k calculated via the Pearson Correlation Covariance [77]. After we get the check-in probability estimation, we obtain the corresponding scores,

$$S^u_{i,j} = \frac{p^u_{i,j}}{Z^u_i}, \ \ S^s_{i,j} = \frac{p^s_{i,j}}{Z^s_i}, \ \ S^g_{i,j} = \frac{p^g_{i,j}}{Z^g_i}, \tag{2.16}$$

where Z_i^u, Z_i^s, Z_i^g are normalization terms. $Z_i^u = \max_{l_j \in L \setminus L_i}\{p_{i,j}^u\}$, $Z_i^s = \max_{l_j \in L \setminus L_i}\{p_{i,j}^s\}$, $Z_i^g = \max_{l_j \in L \setminus L_i}\{p_{i,j}^g\}$, where $L \setminus L_i$ denotes the POIs user u_i has not visited.

Representative Work for Model-based Method

In [7], Cheng et al. employ probabilistic matrix factorization (PMF) [80] and probabilistic factor model (PFM) [55] to learn user preference for recommending POIs. Suppose $\mathcal{U}$ denote the set of users and $\mathcal{L}$ denote the set of POIs. U_i and L_j denote the latent feature of user u_i and POI l_j. PMF-based method assumes Gaussian distribution on observed check-in data and Gaussian priors on the user latent feature matrix U and POI latent feature matrix L. Then, the objective function to learn the model is as follows,

$$\min_{U,L} \sum_{i=1}^{|\mathcal{U}|} \sum_{j=1}^{|\mathcal{L}|} I_{ij}(g(c_{ij}) - g(U_i^T L_j))^2 + \lambda_1 ||U||_F^2 + \lambda_2 ||L||_F^2, \quad (2.17)$$

where $g(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, c_{ij} is the checked-in frequency of user u_i at POI l_j. I_{ij} is the indicator function to record the check-in state of u_i at l_j. Namely, I_{ij} equals one when the i-th user has checked-in at j-th POI; otherwise zero. After learning the user and POI latent features, the preference score of u_i over l_j is measured by the following score function,

$$P(F_{ul}) = g(U_i^T L_j), \quad (2.18)$$

where $g(\cdot)$ is the sigmoid function.

In addition, the geographical influence can be modeled through MGM, shown in Eq. (2.3) of Section 2.2.1. Then, a fused model is proposed to combine user preference learned from Eq. (2.17) and geographical influence modeled in Eq. (2.3). The proposed model determines the probability P_{ul} of a user u visiting a location l via the product of the preference score estimation and the probability of whether a user will visit that place in terms of geographical influence ,

$$P_{ul} = P(F_{ul}) \cdot P(l|C_u), \tag{2.19}$$

where $P(l|C_u)$ is calculated via the MGM and $P(F_{ul})$ encodes a user's preference on a location.

2.3.2 Joint Model

Different from the fused model, the joint model learns several influential factors together and then recommends POIs from the jointly learned model. Compared with the fused model, a joint model connects different influential factors into the same final training target—the check-in behavior. The joint model depicts the check-in behavior as a synthesized decision influenced by several factors together, which better reflects the real scenario than the fused model. This advantage over the fused model makes the joint model attract more attention. Recently a number of joint models [15, 16, 29, 35, 44, 45, 54, 103, 111] have been proposed for POI recommendation. We categorize the joint models into three types: 1) MF-based joint model that incorporates factors such as geographical influence and temporal influence into traditional collaborative filtering model like matrix factorization and tensor factorization, e.g., [15, 16, 44, 54, 103]; 2) generative graphical model that establishes a generative model according to the check-ins and extra influences like geographical information, e.g., [29, 45, 35, 111]; 3) neural network model that jointly models the influential factors in a neural network, e.g., [52, 49].

Representative Work for MF-based Joint Model

In this section, we report two representative studies about the MF-based joint model, which incorporate temporal effect and geographical effect into a matrix factorization framework, respectively.

In [15], Gao et al. propose a Location Recommendation framework with Temporal effects (LRT), which incorporates temporal influence into a matrix factorization model. The LRT model contains two assumptions on temporal effect: 1) non-uniformness, users' check-in preferences change at different hours of one day; 2) consecutiveness, users' check-in preferences are similar in consecutive time slots. To model the non-uniformness, LRT separates a day into T slots, and defines time-dependent user latent feature $U_t \in R^{m \times d}$, where m is the number of users, d is the latent feature dimension, and $t \in [1, T]$ indexes time slots. Suppose that $C_t \in R^{m \times n}$ denotes a matrix depicting the check-in frequency at temporal state t. U and L denote the latent feature matrix for user and POI, respectively. Using the non-negative matrix factorization to model the POI recommendation system, the time-dependent objective function is as follows,

$$\min_{U_t \geq 0, L \geq 0} \sum_{t=1}^{T} ||Y_t \odot (C_t - U_t L^T)||_F^2 + \alpha \sum_{t=1}^{T} ||U_t||_F^2 + \beta ||L||_F^2, \quad (2.20)$$

where Y_t is the corresponding indicator matrix, α and β are the regularizations. Furthermore, the temporal consecutiveness inspires to minimize the following term,

$$\min \sum_{t=1}^{T} \sum_{i=1}^{m} \phi_i(t, t-1) ||U_t(i,:) - U_{t-1}(i,:)||_2^2, \quad (2.21)$$

where $\phi_i(t, t-1) \in [0, 1]$ is defined as a temporal coefficient that measures user preference similarity between temporal state t and $t-1$. The temporal coefficient could be calculated via cosine similarity according to users' check-ins at state t and $t-1$. To represent the Eq. (2.21) in matrix form, we get

$$\min \sum_{t=1}^{T} Tr((U_t - U_{t-1})^T \Sigma_t (U_t - U_{t-1}), \tag{2.22}$$

where $\Sigma_t \in R^{m \times m}$ is the diagonal temporal coefficient matrix among m users. Combining the two minimization targets, the objective function of the LRT model is gained as follows,

$$\begin{aligned} \min_{U_t \geq 0, L \geq 0} \sum_{t=1}^{T} ||Y_t \odot (C_t - U_t L^T)||_F^2 + \alpha \sum_{t=1}^{T} ||U_t||_F^2 + \beta ||L||_F^2 \\ + \lambda \sum_{t=1}^{T} Tr((U_t - U_{t-1})^T \Sigma_t (U_t - U_{t-1}), \end{aligned} \tag{2.23}$$

where λ is a non-negative parameter to control the temporal regularization. User and location latent representations can be learned by solving the above optimization problem. Then, the user check-in preference $\hat{C}_t(i,j)$ at each temporal state can be estimated by the product of user latent feature and location feature $(U_t(i,:)L(j,:)^T)$. Recommending POIs for users is to find POIs with the higher value of $\hat{C}(i,j)$. To aggregate different temporal states' contributions, $\hat{C}(i,j)$ is estimated through

$$\hat{C}(i,j) = f(\hat{C}_1(i,j), \hat{C}_2(i,j), \ldots, \hat{C}_T(i,j)), \tag{2.24}$$

where $f(\cdot)$ is an aggregation function, e.g., sum, mean, maximum, and voting operation.

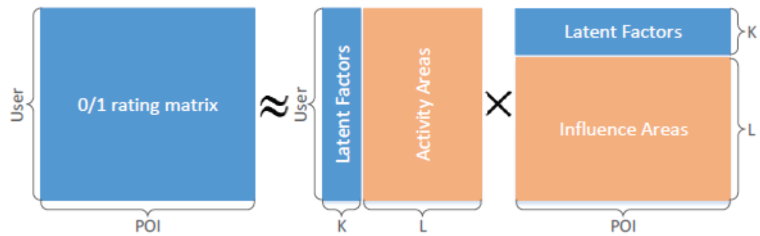


Figure 2.11: Demonstration of GeoMF model [44]

In [44], Lian et al. propose the Geo matrix factorization (GeoMF) model to incorporate geographical influence into a weighted regularized matrix factorization model (WRMF) [30, 66]. WRMF is a popular model for one-class collaborative filtering problem, learning implicit feedback for recommendations. GeoMF treats the user check-in as implicit feedback and leverages a 0/1 rating matrix to represent the user check-ins. Furthermore, GeoMF employs an augmented matrix to recover the rating matrix, as shown in Figure 2.11. Each entry in the rating matrix is the combination of two interactions: user feature and POI feature, users' activity area representation and POIs' influence area representation. Suppose there are m users and n POIs. The latent feature dimension is d for user and POI representations, and the latent feature dimension is l for users' activity area and POIs' influence area representations. Then the estimated rating matrix can be formulated as,

$$\tilde{R} = PQ^T + XY^T, \tag{2.25}$$

where $\tilde{R} \in R^{m \times n}$ is the estimated matrix, $P \in R^{m \times d}$ and $Q \in R^{n \times d}$ are the user latent matrix and POI latent matrix, respectively. In addition, $X \in R^{m \times l}$ and $Y \in R^{n \times l}$ are user activity area representation matrix and POI activity area representation matrix, respectively. Define W as the binary weighted matrix whose entry w_{ui} is set as follows,

$$w_{ui} = \begin{cases} \alpha(c_{ui}) + 1 & \text{if } c_{ui} > 0 \\ 1 & \text{otherwise,} \end{cases} \tag{2.26}$$

where c_{ui} is user u's check-in frequency at POI l_i, $\alpha(c_{ui}) > 0$ is a monotonically increasing function with respect to c_{ui}. Following the scheme of WRMF model, the objective function of GeoMF is formulated as,

$$\arg\min_{P,Q,X} ||W \odot (R - PQ^T - XY^T)||_F^2 + \gamma(||P||_F^2 + ||Q||_F^2) + \lambda ||X||_1, \tag{2.27}$$

where Y is POIs' influence area matrix generated from a Gaussian kernel function, P, Q, and X are parameters that need to learn, and γ and λ are regularizations. After learning the latent features from Eq. (2.27), the proposed model estimates the check-in possibility according to Eq. (2.25), and then recommends the POIs with higher values for each user.

Representative Work for Generative Graphical Model

In this section, we present the representative research about the generative graphical model, which incorporates geographical influence into a generative graphical model.

In [45], Liu et al. propose a geographical probabilistic factor analysis framework that takes various factors into consideration, including user preferences, the geographical influence, and the user mobility pattern. The proposed model mimics the user check-in decision process to learn geographical user preferences for effective POI recommendations. Figure 2.12 demonstrates the graphical representation of the proposed model. Specifically, the proposed model assumes that the geographical locations

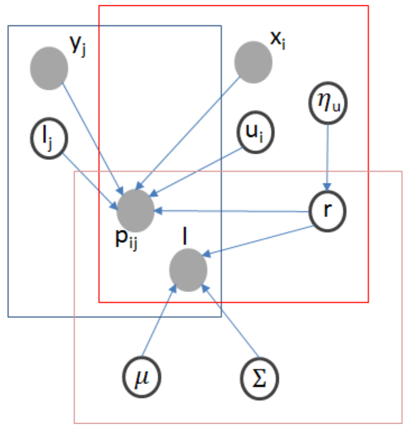


Figure 2.12: A graphical representation of the model [45]

have been clustered into several latent regions denoted as R. A multinomial distribution is applied to model user mobility over the regions R, $r \sim p(r|\eta_u)$, where η_u is a user dependent distribution over latent regions for user u_i. Then, each region $r \in R$ is assumed to be a Gaussian geographical distribution and the POI l_j is characterized by $l \sim \mathcal{N}(\mu_r, \sum_r)$ with μ_r and $\sum_r$ being the mean vector and covariance matrix of the region. In addition, the user check-in process is affected by the following factors: (1) each user u_i is associated with an interest $\alpha(i,j)$ with respect to POI l_j; (2) each POI l_j has popularity ρ_j; and (3) the distance between the user and the POI $d(u_i, l_j)$. Then, the probability of user u_i visiting POI l_j can be formulated as,

$$p(u_i, l_j) \propto \alpha(i,j)\rho_j(d_0 + d(u_i, l_j))^{-\tau}, \tag{2.28}$$

where a power-law like the parametric term $(d_0 + d(u_i, l_j))^{-\tau}$ is used to model the distance factor. Moreover, the user preference for POI can be represented as a linear combination of a latent factor $\mathbf{u}_i^T \mathbf{l}_j$ and a function of user and item observable properties $x_i^T W y_j$, namely

$$\alpha(i,j) = \mathbf{u}_i^T \mathbf{l}_j + x_i^T W y_j. \tag{2.29}$$

ALGORITHM 1: Model generative process

1: Draw a region $r \sim \text{Multinomial}(\eta_u)$
2: Draw a location $l \sim \mathcal{N}(\mu_r, \sum_r)$
3: Draw a user preference
4: Generate user latent factor $\mathbf{u}_i \sim P(u_i; \Phi_{\mathbf{u}})$
5: Generate POI latent factor $\mathbf{l}_j \sim P(\mathbf{l}_j; \Phi_{\mathbf{l}_j})$
6: User-item preference $\alpha(i,j) = \mathbf{u}_i^T \mathbf{l}_i + x_i^T W y_j$
7: Generate $p_{ij} \sim P(f_{ij})$, where $p_{ij} = (\mathbf{u}_i^T \mathbf{l}_j + x_i^T W y_j)\rho_j(d_0 + d(u_i, l_j))^{-\tau}$

The proposed model uses implicit user check-in data to model user preferences and the distribution of check-in counts are usually skewed, so a Bayesian probabilistic non-negative latent factor model is employed: $p_{ij} \sim P(f_{ij})$ where $f_{ij} = \alpha(i,j)\rho_j(d_0 + d(u_i, l_j))^{-\tau}$. The proposed model shown in Figure 2.12 can be generated according to Algorithm 1.

After the parameters are learned, the proposed model predicts the number of check-ins of a user for a given POI as $\mathbb{E}(p_{ij}|u_i, l_j) = (\mathbf{u}_i^T \mathbf{l}_j + x_i^T W y_j)\rho_j(d_0 + d(u_i, l_j))^{-\tau}$. Moreover, POI recommendations are based on the predicted check-in times. The larger the predicted value is, the more likely the user will choose this POI.

Representative Work for Neural Network Model

In this section, we report the representative research about neural network model, which extends the recurrent neural network (RNN) [22] with spatial and temporal information for next POI recommendation.

In [49], Liu et al. propose the Spatial Temporal Recurrent Neural Network (ST-RNN) model to predict the next POI. Formally, let P be a set of users and Q be a set of locations, $\mathbf{p}_u, \mathbf{q}_l \in R^d$ indicate the latent vectors of user u and location l.

For each user u, the history of where he has been is given as $Q^u = q^u_{t_1}, q^u_{t_2}, \ldots$, where $q^u_{t_i}$ denotes where user u is at time t_i. And the history of all users is denoted as $Q^U = \{Q^{u1}, Q^{u2}, ...\}$. Given historical records of a users, the task is to predict where a user will go next at a specific time t. The possibility of user u visit location l at time t can be estimated by the following function,

$$o_{u,t,l} = (\mathbf{h}^u_{t,q_l} + \mathbf{p}_u)^T \mathbf{q}_l, \tag{2.30}$$

where $\mathbf{p}_u, \mathbf{q}_l \in R^d$ indicate the latent vectors of user u and location l, and $\mathbf{h}^u_{t,q_l}$ captures the user's dynamic interests under spatial and temporal contexts learned from a RNN model.

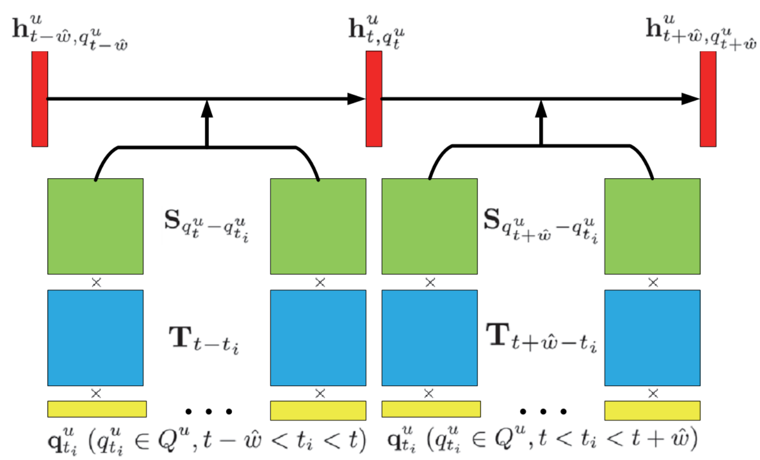


Figure 2.13: Overview of ST-RNN [49]

Specifically, the hidden representation for capturing user interest's dynamics is learned by the ST-RNN model, shown in Figure 2.13. $\mathbf{h}^u_{t,q^u_t}$ denotes the dynamic interest representation of user u at time t, $\mathbf{q}^u_{t_i}$ is the latent vector of the location the user visits at time t_i, and $\hat{w}$ is window size. Therefore, $\mathbf{h}^u_{t,q^u_t}$ can be formulated from the visited POIs in the watching window,

$$\mathbf{h}^u_{t,q^u_t} = f\Big(\sum_{q^u_{t_i} \in Q^u, t-\hat{w} < t_i < t} \mathbf{M}\mathbf{q}^u_{t_i} + \mathbf{C}\mathbf{h}_{t-\hat{w},q^u_{t-\hat{w}}} \Big), \tag{2.31}$$

where $f(x)$ is sigmoid function, $\mathbf{M}$ denotes the transition matrix for input elements to capture the current behavior of the user, and $\mathbf{C}$ is the recurrent connection of the previous status propagating sequential signals. Furthermore, $\mathbf{M}$ is used to capture the spatial and temporal contexts, defined as $\mathbf{M} = \mathbf{S}_{q_t^u - q_{t_i}^u}\mathbf{T}_{t-t_i}$, where $\mathbf{S}_{q_t^u - q_{t_i}^u}$ is the distance-specific transition matrix for the geographical distance between q_t^u and $q_{t_i}^u$, and $\mathbf{T}_{t-t_i}$ denotes the time-specific transition matrix for the time interval $t - t_i$. To learn the parameters $\mathbf{S}_{q_t^u - q_{t_i}^u}$, $\mathbf{T}_{t-t_i}$, $\mathbf{q}_{t_i}^u$, $\mathbf{C}$, $\mathbf{h}_{t,q_t^u}^u$, $\mathbf{p}_u$, and $\mathbf{q}_l$, [49] uses the Bayesian Personalized Ranking (BPR) [72] and Back Propagation Through Time (BPTT) [79] to infer the model.

2.3.3 Summary

In this section, we categorize the POI recommendation systems according to the used methodologies. As shown in Section 2.2, POI recommendations are influenced by several factors. Two different means are used to model the influential factors together, 1) recommend POIs separately according to each factor and then ensemble results, and 2) establish a joint model incorporating different influential factors. Table 2.2 show the summary of POI recommendation systems categorized by the methodology.

2.4 Performance Evaluation

In this section, we report two important aspects for evaluating a POI recommendation system: data sources and metrics. We first summarize several popular LBSN datasets. Then, we describe the metrics used to verify the effectiveness of the recommendation results.

Table 2.2: Summary of POI recommendation systems categorized by methodology

	Fused Model		**Joint Model**		
	Memory-based Model	**Model-based Model**	**MF-based Model**	**Generative Model**	**Neural Model**
[7]		√			
[107]	√				
[103]			√		
[128]			√		
[44]			√		
[126]			√		
[113]	√				
[116]	√				
[15]			√		
[16]			√		
[14]			√		
[121]			√		
[45]				√	
[49]					√
[35]				√	
[111]				√	
[52]					√
[95]			√		

2.4.1 Data Sources

Gowalla, Brightkite, and Foursquare are famous benchmark datasets available for evaluating a POI recommendation model. In this section, we briefly introduce these datasets and describe the statistics, shown in Table 2.3. In particular, the Brightkite and Gowalla data in [11] contains the user friendships and user check-in sequences. The check-in data also contains the timestamp and geographical information, shown in Figure 2.14. Since the user check-in data is recorded chronologically, this dataset can also be used for sequential modeling. The Foursquare data in [17] and [18] collect the data using the similar format, including the user friendship and check-in sequential records. In addition, the Foursquare data in [17] and [18] also includes the user comments. But the user comments are very limited and many check-in records do not contain comments. The Gowalla data in [7] also contain the user sequential check-in records(including the user comments) and user friendships. But the data in [7] store the geographical information(i.e., latitude and longitude) separately. The data in [11, 7, 17, 18] are collected globally and contain the similar information, including the sequential check-in records and user friendship. On the contrary, the Foursquare data in [1] is collected in city level, including crawled check-in data in Los Angeles and New York. This dataset not only contains the user friendships and user comments, but also the user profiles, venue category information, and photos for the venue. Yet this dataset does not contain the sequential information or timestamps for each check-in.

user id	check-in time	latitude	longitude	POI id

Figure 2.14: Demonstration of check-in meta record

Table 2.3: LBSN datasets for POI recommendation

Name	Statistics
Brightkite [11]	4,491,143 check-ins from 58,228 users
Gowalla 1 [11]	6,442,890 check-ins from 196,591 users
Gowalla 2 [7]	4,128,714 check-ins from 53,944 users
Foursquare 1 [17]	2,073,740 check-ins from 18,107 users
Foursquare 2 [18]	1,385,223 check-ins from 11,326 users
Foursquare 3 [1]	325,606 check-ins from 80,606 users

2.4.2 Metrics

Most of the POI recommendation systems utilize metrics of *precision* and *recall*, which are two general metrics to evaluate the model performance in information retrieval [12, 23]. To see the balance of precision and recall, *F-score* is also introduced in some work. Since the precision and recall are low for POI recommendation, some studies [45, 106] introduce one relative metric, which measures the model comparative performance over random selection.

The precision and recall in the top-N recommendation system are denoted as P@N and R@N, respectively. P@N measures the ratio of recovered POIs to the N recommended POIs, and R@N means the ratio of recovered POIs to the set of POIs in the testing data. For each user $u \in \mathcal{U}$, $\mathcal{L}_u^T$ denotes the set of correspondingly visited POIs in the test data, and $\mathcal{L}_u^R$ denotes the set of recommended POIs. Then, the definitions of P@N and R@N are formulated as follows,

$$P@N = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathcal{L}_u^R \cap \mathcal{L}_u^T|}{N}, \tag{2.32}$$

$$R@N = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathcal{L}_u^R \cap \mathcal{L}_u^T|}{|\mathcal{L}_u^T|}. \tag{2.33}$$

Further, *F-score* is the harmonic mean of precision and recall. Therefore, the *F-score* is defined as,

$$F\text{-}score@N = \frac{2 * P@N * R@N}{P@N + R@N}. \tag{2.34}$$

In order to better compare the results, a relative metric is introduced. Relative precision@N and recall@N are denoted as r-P@N and r-R@N, respectively. Let $\mathcal{L}_u^C$ denote the candidate POIs for each user u, namely POIs the user has not checked-in, then precision and recall of a random recommendation system is $\frac{|\mathcal{L}_u^T|}{|\mathcal{L}_u^C|}$ and $\frac{|N|}{|\mathcal{L}_u^C|}$, respectively. Then, the relative precision@N and recall@N are defined as,

$$r - P@N = \frac{P@N}{|\mathcal{L}_u^T|/|\mathcal{L}_u^C|}, \tag{2.35}$$

$$r - R@N = \frac{R@N}{|N|/|\mathcal{L}_u^C|}. \tag{2.36}$$

2.5 Conclusion

Due to the prevalence of LBSNs and the importance of POI recommendation systems in LBNSs, we provide a systematic survey of the related recent studies. We review over 60 papers published in related top conferences and journals, including but not limited to AAAI, IJCAI, SIGIR, KDD, WWW, RecSys, UbiComp, ACM SIGSPATIAL, ACM TIST, and IEEE TKDE. we categorize the systems by the influential factors and the methodology. Particularly we also report the representative work in each category. This chapter presents a panorama of this research with a balanced depth and scope.

□ **End of chapter.**

Chapter 3

Understanding Human Mobility from Geographical Perspective

POI recommendation is a significant service for LBSNs. It recommends new places such as clubs, restaurants, and coffee bars to users. Whether recommended locations meet users' interests depends on three factors: user preference, social influence, and geographical influence. Especially, capturing the geographical influence plays the most important role for POI recommendations. Previous studies observe that checked-in locations disperse around several centers and employ Gaussian distribution based models to approximate users' check-in behaviors. Yet centers discovering methods are not satisfactory in prior work. In this chapter, we propose two models—Gaussian mixture model (GMM) and genetic algorithm based Gaussian mixture model (GA-GMM) to capture geographical influence. More specifically, we exploit GMM to automatically learn users' activity centers; further we utilize GA-GMM to improve GMM by eliminating outliers. Experimental results on a real-world LBSN dataset show that GMM beats several popular geographical capturing models in terms of POI recommendation, while GA-GMM excludes the effect of outliers and enhances GMM.

3.1 Introduction

POI recommendation is a significant service for LBSNs. With the development of mobile devices and Web 2.0 technologies, many LBSNs like Foursquare and Gowalla emerge and attract many users. These LBSNs allow users to check-in at POIs, make friends, and share location-related information. In order to help users discover new interesting places in LBSNs, the POI recommendation arises.

User preference, social influence, and geographical influence are three aspects responsible for users' check-in activities [106, 107]. Generally we derive user preference from user-based collaborative filtering, explore social influence based on users' social relationships, and model geographical influence from check-in locations' spatial features. And then we construct a POI recommendation system in the way of combining those three kinds of influence. The representative work is as follows. Ye et al. [107] propose a linear fused framework to combine them and Cheng et al. [7] propose a fused model to recommend POIs.

For POI recommendation in LBSNs, research about geographical influence is new and requires more attention, comparing with user preference and social influence. It is well-defined on how to derive user preference and social influence in a recommendation system [34, 78]. Note that users' evaluations for items reflect their preferences and friends are inclined to share preferences. We derive user preference from user-based collaborative filtering and introduce social influence by containing similarity among friends. For POI recommendation system, we use collaborative filtering method to get user preference through treating locations as items and check-in frequencies as rating values, and we capture social influence by including friends' similarity in check-in locations [7, 106, 107]. In 2010, Ye et al. [106] first propose POI recommendation for LBSNs and utilize power law principle to model users' geographical influence. In the meantime, Cho et al. [11] study the user mobility in LBSNs inspired by Gonzalez's discovery [21]. The study focuses on those users who frequently check-in, since Gonzalez's discovery bases on call logs data that have strong periodic property. They propose a periodic mobility model (PMM) to capture user's geographical influence for location prediction in LBSNs. Next, Cheng et al. [7] propose Multi-center Gaussian Model (MGM) to capture geographical influence. This model assumes a user's visited locations disperse around several centers and utilizes a greedy method to discover centers. It defines a district by a fixed distance and thus ignores discrepancy between users. In summary, Gaussian distribution based models perform well for POI recommendation, but we still encounter challenges in how to discover the activity centers accurately and how to eliminate the effect of outliers.

In order to find activity centers more accurately and eliminate outliers, we propose two models—Gaussian mixture model (GMM) and genetic algorithm based Gaussian mixture model (GA-GMM) to capture geographical influence. From geographical perspective, people prefer places that are nearer to their activity centers. Those frequent checked-in places naturally form a user's activity district. According to locations' spatial clustering feature, we apply GMM to find a user's activity district centers. However, outliers exist in the observed data that do harm to learn the model. How to eliminate the impact of outliers? Thang et al. [90] propose a genetic algorithm based EM algorithm to implement the trimmed likelihood estimate (TLE) method [64] to eliminate the outliers in mixture models. We exploit this genetic based EM algorithm to train GMM. The genetic algorithm based GMM (GA-GMM) improves GMM and models the geographical influence better.

Our contributions are as follows. First, we propose GMM to automatically learn users' activity centers via exploring their check-in history records. Moreover, we enhance GMM by GA-GMM to eliminate outliers. Finally, we conduct experiments on a real-world LBSN dataset and demonstrate that the proposed models capture the geographical information better and improve the accuracy of POI recommendation.

3.2 Related Work

In this section, we introduce related work in three aspects: POI recommendation in LBSNs, geographical influence capturing methods, and GA-GMM.

POI recommendation in LBSNs is a new research topic. POI recommendation is widely used in GPS-based mobile devices at first [28, 33]. In 2010, Ye et al. [106] first propose POI recommendation in LBSNs. Further, Ye et al. [107] point out that user preference, social influence, and geographical influence are three aspects responsible for recommending POIs and geographical influence is the most important among the three factors. The representative work is as follows. Ye et al. [107] recommend POIs through a linear fused framework combining user preference, social influence, and geographical influence. Cheng et al. [7] propose a fused model to combine them to recommend POIs.

Study of geographical influence capturing methods is new for POI recommendation. In 2010, Ye et al. [106] first propose POI recommendation for LBSNs and arise a power law principle to capture geographical influence for POI recommendation. Earlier related work about geographical influence appears in the study of user movement pattern. Gonzalez et al. [21] build a model using call logs and discover that activities of an individual usually center around a small number of frequently visited locations. Based on this, Cho et al. [11] study the specific users frequently checking in and propose a periodic mobility model (PMM) to capture geographical influence for location prediction in LBSNs. Cheng et al. [7] employ a multi-center Gaussian model (MGM) to capture the geographical feature of locations in the proposed fused POI recommendation model.

Genetic algorithm based GMM (GA-GMM) is a method to eliminate outliers when learning GMM. Trimmed likelihood estimate (TLE) method is adopted to eliminate outliers in some studies of mixture model analysis [64]. Thang et al. [90] first propose a genetic algorithm based method to implement the trimmed likelihood estimating method to train mixture models and demonstrate the performance through a genetic algorithm based GMM (GA-GMM). Wang et al. utilize the GA-GMM to process EEG signal and apply it on brain-computer interface [93].

3.3 Model

3.3.1 Gaussian Mixture Model

Gaussian mixture model (GMM) [63] is the most widely used mixture model. We can formulize it as follows:

$$p(\mathrm{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}_i | \mu_k, \textstyle\sum_k),$$

where $p(\mathrm{x}_i)$ denotes probability dense distribution of data x_i, μ_k indicates mean value, $\sum_k$ indicates covariance matrix for a base distribution, K denotes the number of base components, and π_k is the mixing coefficient.

We exploit GMM to capture geographical influence in POI recommendation. Each Gaussian distribution component represents an activity district and the mean value denotes the longitude and latitude of the district center. Centers may be his home, office, or some specific entertainment place. We assume places nearer to some center are geographically easier to arrive and people prefer those places.

In the following, we show how to recommend POIs through GMM. For a user, a location's geographical information ([longitude, latitude]) in his check-in history records represents data x_i. We recommend POIs through the following steps:

1. Learn the parameters of GMM,
2. Calculate candidate locations' probabilities fitting the trained model, and
3. Sort the candidate locations and recommend the top K locations.

3.3.2 Genetic Algorithm Based Gaussian Mixture Model

In order to eliminate the effect of outliers, we introduce a genetic algorithm based Gaussian mixture model (GA-GMM). Generally we could use maximum likelihood EM algorithm to learn GMM [63]. If we use Θ to denote the parameters, likelihood function could be represented as

$$p(X|\Theta)_{ML} = \prod_{i=1}^{n} p(\mathrm{x}_i|\Theta).$$

Further, if we use the logarithm form, we can denote the objective of maximum likelihood EM algorithm as follows:

$$\hat{\Theta}_{ML} = \arg\max \log p(X|\Theta)_{ML} = \arg\max \sum_{i=1}^{n} \log p(\mathrm{x}_i|\Theta). \quad (3.1)$$

This formula includes all observed data. Trimmed likelihood estimate (TLE)—that aims to to select the subset of data with maximum sum of likelihood values—is used to eliminate the outliers [64]. We can use a genetic algorithm to find the optimal subset and exploit maximum likelihood EM algorithm to learn the parameters of GMM, as illustrated in Algorithm 2 [90]. In this case, the objective function could be represented as

$$\log p_{TLE}(X|\Theta) = \sum_{i=1}^{n} w_i \log p(\mathrm{x}_i|\Theta), \quad (3.2)$$

where $\forall i = 1, 2, ..., n, w_i \in \{0, 1\}$ and $\sum_{i=1}^{n} w_i = m$, m represents the number of valid data. When $w_i = 1$, it indicates that the corresponding data is chosen into the subset. Otherwise, the data is an outlier and should be discarded. Hence, the result is a subset of size m out of n original samples, which fits GMM most in terms of likelihood contribution.

As a genetic algorithm, GA-GMM contains properties of genetic algorithm—it includes encoding scheme, fitness function, and operators like crossover, mutation, and selection. We use the standard way to implement crossover and selection [59]. Encoding scheme, fitness function, and a self-defined mutation (Guided Mutation) are defined as follows.

Definition 4 *Encoding scheme. The chromosome is encoded into a binary string and each bit represents the existence of corresponding observed data. Each chromosome and its corresponding mixture model will be a possible solution to our problem.*

Definition 5 *Fitness function. The fitness score function is set as the trimmed logarithm likelihood of the corresponding GMM of a chromosome*—$\log p_{TLE}(X|\Theta)$.

Definition 6 *Guided Mutation. Guided Mutation ensures the chromosome in a population to mutate toward maximizing fitness score. It means we choose chromosome that has higher value fitting trained GMM.*

3.4 Experiment

3.4.1 Setup and Metrics

We prepare the data by cleaning and splitting. We filter locations of less than 10 visits. And then we split the dataset into three non-overlapping sets in sequence: a redundant set, a training set, and a test set. The test set keeps 10% of the whole data set. We test different cases in which the proportion of training data is 90% and 50% respectively. When training

ALGORITHM 2: Genetic-based Expectation Maximization Algorithm

1. t=0;
2. Initialize $P_0(t)$;
3. **for** $t = 1 : G$ **do**
4. $\quad P_1(t) \leftarrow$ perform several cycles of EM on $P_0(t)$;
5. $\quad P_2(t) \leftarrow$ Guided Mutation in $P_1(t)$;
6. $\quad fScore_2 \leftarrow$ evaluate $P_2(t)$;
7. $\quad P_0(t)' \leftarrow$ selection and crossover to generate offspring from $P_2(t)$;
8. $\quad P_1(t)' \leftarrow$ perform several cycles of EM on $P_0(t)'$;
9. $\quad P_2(t)' \leftarrow$ Guided Mutation in $P_1(t)'$;
10. $\quad fScore_2' \leftarrow$ evaluate $P_2(t)'$;
11. $\quad P_3(t) \leftarrow$ selection from $[P_2(t), P_2(t)']$;
12. $\quad iBest \leftarrow$ best individual from $P_3(t)$;
13. $\quad$ **if** $iBest$ satisfies convergence condition **then** break;
14. $\quad P_0(t+1) \leftarrow P_3(t)$;
15. $\quad t = t + 1$;
16. Perform EM on $iBest$ until convergence;

data set is 90%, there is no redundant data. When the training data set is 50%, redundant data is the former 40% data that will be discarded.

We evaluate the performance of different models in capturing geographical influence by the accuracy of POI recommendation that is measured by Precision and Recall. POI recommendation is to recommend the top-N highest ranked locations. However, the system should not recommend locations the user has checked in. To evaluate the performance of POI recommendation, we use the Precision@N and Recall@N as the metrics that are standard metrics to measure the performance of POI recommendation [107]. Precision@N defines the ratio of recovered POIs to the N recommended POIs and Recall@N defines the ratio of recovered POIs to the size of test set.

Table 3.1: Data statistics

Min. C.	**Max. C.**	**Avg. C.**	**Min. T.**	**Max. T.**	**Avg. T.**
1,001	50,243	2,505	366	968	593

3.4.2 Dataset

We use the Gowalla data records from February 2009 to September 2011. We select 3836 active users' records to experiment. We define active users as users whose check-ins are more than 1000 times and experience of using Gowalla is more than 1 year. After removing locations with less than 10 visits, all check-ins of active users include 183,667 different locations. We illustrate statistics of the data in Table 3.1, where "C." represents the check-in times of a user and "T." represents the time span (unit is day) from first check-in to last check-in.

3.4.3 Results

We compare the POI recommendation performance of GMM and GA-GMM with Gaussian model (GM) and Multi-center of Gaussian model (MGM) [7] when training data set is 90% and 50% respectively.

Gaussian model (GM) [21] is a baseline model used in [11]. It models human movement as a stochastic process centered around a single point.

Multi-center Gaussian model (MGM) [7] is a latest model. It uses a fixed distance to define a district. When check-ins in a district are more than a threshold, the mean of all check-ins is the center. It utilizes a greedy method to find the district and requires no overlapping between two districts.

We illustrate experimental results in Figure 3.1. GMM outperforms GM and MGM; further GA-GMM improves GMM. Hence,

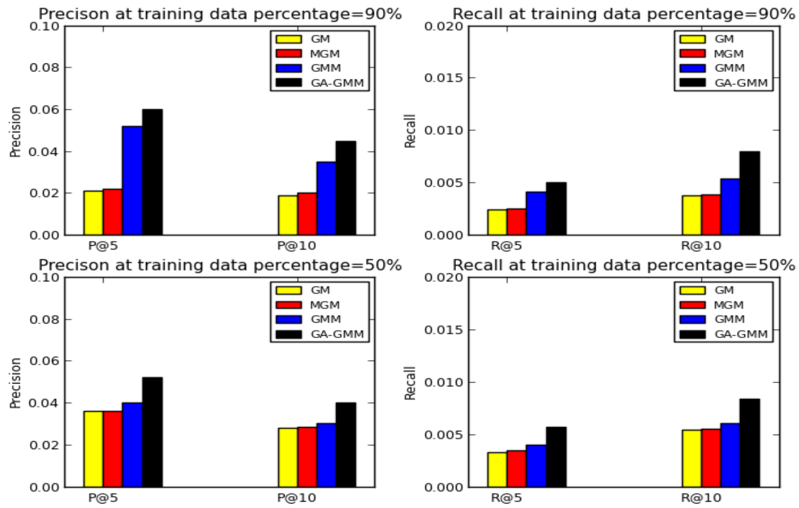


Figure 3.1: Comparison of different models

GA-GMM could better capture the geographical influence. In the experiment, we set the number of centers in GMM and GA-GMM as 2 for simplicity, since Cho et al. propose that the check-in behavior comprises two states in [11]. We set the radius of a region in MGM as 1 kilometer and the threshold as 10% (that means the ratio of check-ins in one district is at least 10% of all his check-ins).

3.5 Conclusion

We apply GMM and GA-GMM to capture geographical influence in POI recommendation. We exploit GMM to automatically learn users' activity centers; further we utilize GA-GMM to improve GMM by eliminating outliers. According to experimental results, we draw conclusions as follows. 1) GMM outperforms the baseline model GM and the latest model MGM. 2) GA-GMM eliminates the outliers of data and improves GMM.

It discovers the activity centers more precisely, which increases the accuracy of POI recommendation.

□ **End of chapter.**

Chapter 4

Understanding Human Mobility from Temporal Perspective

Understanding user mobility from the temporal perspective is the key to POI recommendation that mines user check-in sequences to suggest interesting locations for users. Because user mobility in LBSNs exhibits strong temporal patterns—for instance, users would like to check-in at restaurants at noon and visit bars at night. Hence, capturing the temporal influence is necessary to ensure the high performance in a POI recommendation system. We summarize the temporal characteristics of user mobility in LBSNs in three aspects: periodicity, consecutiveness, and non-uniformness. More importantly, we observe that the temporal characteristics exist at different time scales. To this end, we propose an Aggregated Temporal Tensor Factorization (ATTF) model for POI recommendation to capture the three temporal features together, as well as at different time scales. Specifically, we employ a temporal tensor factorization method to model the check-in activity, subsuming the three temporal features together. Next, we exploit a linear combination operator to aggregate temporal latent features' contributions at different time scales. Experiments on two real-world datasets show that the ATTF model achieves better performance than the state-of-the-art temporal models for POI recommendation.

4.1 Introduction

Understanding the user mobility from temporal perspective is important to establish a practical POI recommendation system. Previous studies show that the user mobility in LBSNs exhibits significant temporal features [11, 15, 113]. For example, users always stay in the office in the Monday afternoon, and enjoy entertainments in bars at night. In summary, the temporal features in users' check-in data can be abstracted in three aspects.

- **Periodicity.** Users share the same periodic pattern, visiting the same or similar POIs at the same time slot [11, 113]. For instance, a user always visits restaurants at noon, so do other users. Hence, the periodicity inspires the time-aware collaborative filtering method to recommend POIs [113].

- **Consecutiveness.** A user's current check-in is largely correlated with the recent check-in [9, 15]. Gao et al. [15] model this property by assuming that user preferences are similar in two consecutive hours. Cheng et al. [9] assume that two checked-in POIs in a short term are highly correlated in latent feature space.

- **Non-uniformness.** A user's check-in preference changes at different hours of a day [15]. For example, at noon a user may visit restaurants while at night the user may have fun in bars.

By capturing the observed temporal features, a variety of systems are proposed to enhance POI recommendation performance [9, 15, 113], which gain better performance than general collaborative filtering (CF) methods [107]. Nevertheless, previous work [9, 15, 113] cannot model the three features together. Moreover, an important fact is ignored in prior work that the temporal influence exists at different time scales. For example, in day level, you may check-in at POIs around your home in the earning morning, visit places around your office in the day time, and have fun at nightclubs in the evening. In week level, you may stay in the city for work on weekdays and go out for vocation on weekends. Hence, to better model the temporal influence, capturing the temporal features at different time scales is necessary.

In this chapter, we propose an Aggregated Temporal Tensor Factorization (ATTF) model for POI recommendation to capture the three temporal features together, as well as at different time scales. We construct a user-time-POI tensor to represent the check-ins as shown in Figure 4.1, and then employ the interaction tensor factorization [75] to model the temporal effect. Different from prior work that represents the temporal influence at a single scale, we index the temporal information at different scales, i.e., hour, week day, and month, to learn the latent representation. Furthermore, we employ a linear combination operator to aggregate different temporal latent features' contributions, which capture the temporal influence at

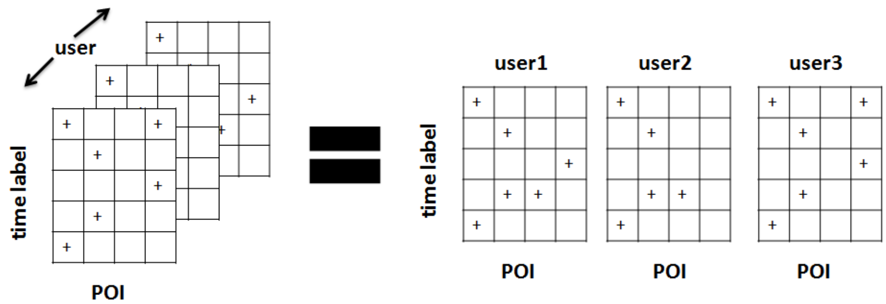


Figure 4.1: Tensor illustration for check-ins

different scales. Specifically, our ATTF model learns the three temporal properties as follows: (1) periodicity is learned from the temporal CF mechanism; (2) consecutiveness is manifested in two aspects—time in a slot brings the same effect through sharing the same time factor, and the relation between two consecutive time slots can be learned from the tensor model; (3) non-uniformness is depicted by different time factors representing different time slots from each time scale perspective. Moreover, an aggregate operator is introduced to combine the temporal influence at different scales, i.e., hour, week day, and month, and represent the temporal effect in a whole.

To sum up, we propose the ATTF model to seek a better way to capture the temporal influence for POI recommendation. Moreover, we establish an embedding neural network to represent the ATTF model, which gives new insights to understand the proposed model from the neural network perspective. Specifically, this chapter makes the following contributions.

- To the best of our knowledge, this is the first temporal tensor factorization method for POI recommendation, subsuming all the three temporal properties: periodicity, consecutiveness, and non-uniformness.

- We propose a novel model to capture temporal effect in POI recommendation at different time scales. Experimental results show that our model outperforms prior temporal model more than 20%.
- The ATTF model is a general framework to capture the temporal features at different scales, which outperforms single temporal factor model and gains 10% improvement in the top-5 POI recommendation task on Gowalla data.
- We understand the ATTF model from the embedding neural network perspective, verifying the effectiveness of the embedding neural network that is a general framework for latent factor models, including rating estimation models (e.g., MF [34]) and ranking models (e.g., our ATTF model).

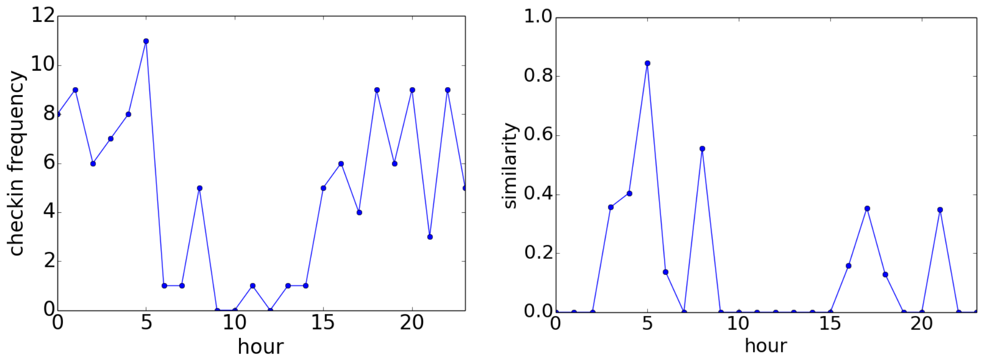
4.2 Related Work

In this section, we first review the literature of POI recommendation. Then, we summarize the progress of modeling temporal effect for POI recommendation. Finally, we review the literature of embedding learning and its applications, which inspires us to understand the proposed ATTF model from the neural network perspective.

POI Recommendation. Most of POI recommendation systems base on the collaborative filtering (CF) techniques, which can be reported in two aspects, memory-based and model-based. On the one hand, Ye et al. [107] propose the POI recommendation problem in LBSNs solved by user-based CF method, and further improve the system by linearly combining the geographical influence, social influence, and preference similarity. In order to enhance the performance, more advanced techniques are then applied, e.g., incorporating temporal influence [113], and utilizing a personalized geographical model via kernel density estimator [116, 117]. On the other hand, model-based CF is proposed to tackle the POI recommendation problem that benefits from its scalability. Cheng et al. [7] propose a multi-center Gaussian model to capture user geographical influence and combine it with social matrix factorization (MF) model [57] to recommend POIs. Gao et al. [15] propose an MF-based model, Location Recommendation framework with Temporal effects (LRT), utilizing similarity between time-adjacent check-ins to improve performance. Lian et al. [44] and Liu et al. [54] enhance the POI recommendation by incorporating geographical information in a weighted regularized matrix factorization model [30]. In addition, some researchers subsume users' comments to improve the recommendation performance [16, 43, 111]. Other researchers model the consecutive check-ins' correlations to enhance the system [9, 51, 128, 127].

Temporal Effect Modeling. In 2011, Cho et al. [11] propose the periodicity of check-in data in LBSNs. People always visit restaurants at noon, so we suffice to recommend users restaurants meeting their tastes at noon. The CF technique helps us to recommend similar POIs at the same time slot. However, experiments in [11] depend on dense check-in data, not fitting most of the users. In 2013, Yuan et al. [113] combine the temporal similarity and non-temporal similarity and propose a new similarity metric to enhance the user-based CF model. At the same year, Gao et al. [15] observe the non-uniformness property (a user's check-in preference changes at different hours of a day), and consecutiveness (a user's preference at time t is similar with time $t-1$). Further, Gao et al. propose LRT model based on MF technique to model the non-uniformness and consecutiveness. Meantime, Cheng et al. [9] propose the Factorized Personalized Markov Chain model [74] with Local Region constraint (FPMC-LR) to capture the consecutiveness, supposing the strong correlation between two consecutive checked-in POIs. However, previous work does not model the three features together nor modeling the temporal influence at different scales.

Embedding Neural Network. The embedding neural network, e.g., word2vec framework [61], has turned out to be a successful semi-supervised learning method. It is used in natural language processing [48, 53]. For the efficacy of the framework in capturing the correlations of items, the embedding neural network is employed to the network embedding [69, 89, 24], and as well as in recommendation systems [102, 88]. Moreover, recent studies [39, 42] show that the neural word embeddings can be treated as a kind of matrix factorization method [34]. This equivalence between neural embeddings and the latent factor models inspires us to understand our ATTF model from the embedding neural network perspective. Our interpretation of ATTF model from neural network perspective verifies that the embedding neural network can be treated as a general framework for latent factor models, including rating estimation models [34] and ranking models (e.g., our proposed ATTF model).



(a) A user's check-in pattern in a day (b) Consecutive hour pair similarity

Figure 4.2: Sparsity demonstration

4.3 Preliminaries

In this section, we first analyze the temporal features of user check-in data. Then, we introduce the time labeling scheme that is the prerequisite of our ATTF model. We analyze user check-in data in Foursquare and Gowalla, which demonstrates the similar check-in pattern. In the following, we show the empirical data analysis result based on a randomly selected user in Foursquare.

4.3.1 Empirical Data Analysis

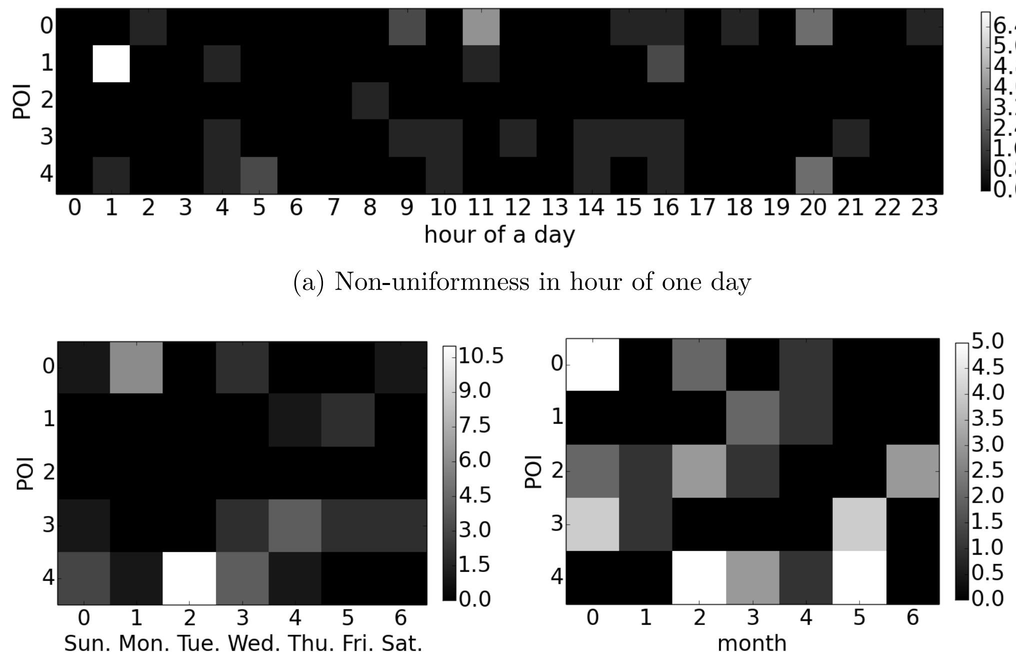
We leverage empirical data analysis to explore the three temporal properties of check-in data. Our analysis verifies previous discoveries, for instance, the non-uniformness—user check-in preference changes at different time of a day [15]. Moreover, we observe some new properties not covered in prior work, e.g., the non-uniformness exists at different time scales.

Data sparsity is a big concern in previous temporal models. Figure 4.2(a) demonstrates a user's check-in pattern in a day.

We observe that the user always has many check-ins in the morning and evening, which verifies the periodicity. The check-in activity repeats in the morning and evening. Figure 4.2(b) shows the consecutive hour pair similarity[1], i.e., the check-in similarity between time t and $t-1$ (t means the hour $1, 2, ..., 24$). We observe that the user check-in preference has high similarity at some consecutive hours, e.g., between 5 o'clock and 4 o'clock, 8 o'clock and 7 o'clock. However, we also find that at some time (e.g., 9:00), the user has few check-ins and the similarity is zero. Therefore, the sparse data make it too hard to model the periodicity and consecutiveness via a counting method (e.g., Pearson correlation or cosine similarity). As shown in Figure 4.2(b), most of the similarities are zero. So the consecutiveness cannot be modeled at most of the time. The dilemma of counting methods in the face of sparse data motivates us to exploit a latent factor learning model. In our model, we use a time latent factor to represent the temporal effect of a time slot, not modeling the temporal effect from the user or POI perspective. Further, the temporal factor is learned from all users' check-ins at the time slot. Therefore, it overcomes the sparsity problem in counting methods.

We observe that the non-uniformness (e.g., the check-in change characteristics) exists at different time scales in users' check-in data. Following [15], we demonstrate an example of a random user's aggregated check-in activities on his/her top 5 most visited POIs in Figure 4.3. Figure 4.3(a) verifies the non-uniformness: a user's check-in preference changes at different hours of a day [15]. As shown in Figure 4.3(a), the most visited

[1] We use cosine similarity here; other measures like Pearson correlation are also applicable.



(a) Non-uniformness in hour of one day

(b) Non-uniformness in day of week

(c) Non-uniformness in month

Figure 4.3: Demonstration of non-uniformness at different time scales

POI changes at different hours. For example, the most visited POI is POI 1 at 1:00 while the most visited POI is POI 4 at 5:00. Besides, we discover there are other change characteristics. As shown in Figure 4.3(b) and Figure 4.3(c), a user's check-in preference changes at different months of a year, and among different days of a week as well. The change of check-ins at different time scales depicts the user preference from different perspectives: 1) A user may check-in at POIs around his/her home in the morning, visit places around the office in the day time, and have fun in bars at night. 2) A user may visit more locations around his/her home or office on weekdays. On weekends, he/she may check-in more at some shopping malls or vocation places. 3) At different months, a user may have different customs. For instance, he/she would visit ice cream shops in the months of summer and hot pot restaurants in the months of winter. Hence, only modeling the non-uniformness at a single scale, we cannot capture all temporal features, which need to be formulated at different scales.

4.3.2 Time Labeling Scheme

Time labeling is a prerequisite of our ATTF model. We use a time latent factor to represent the temporal effect at a specific time, and then learn from a latent factor model. Time labeling scheme determines how to assign a latent factor to specific time. Before diving to the model, we describe the time labeling scheme first.

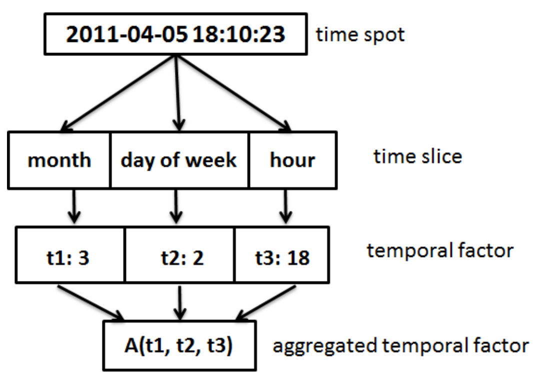


Figure 4.4: Time labeling scheme demonstration

Figure 4.4 demonstrates the time labeling scheme. In order to capture temporal features at different time scales, we represent a time spot with several parts and then aggregate their contributions together. According to the empirical data analysis, we consider temporal features in three time scales: month of a year, day of a week, and hour of a day. Now the temporal effect is formulated by three latent time factors. As shown in Figure 4.4, we leverage three slices to denote a time spot: month of year, day of week, and hour of day. Further they are depicted by three kinds of different temporal latent vectors respectively. So a time spot t is labeled by a tuple (t_1, t_2, t_3), which satisfies that $t_1 \in \{0, 1, \cdots, 12\}$, $t_2 \in \{0, 1, \cdots, 6\}$, $t_3 \in \{0, 1, \cdots, 23\}$ (we have 12 months in a year, 7 days in a week, and 24 hours in a day). Furthermore, we define $T_1 \subset R^{12\times d}$, $T_2 \subset R^{7\times d}$, and $T_3 \subset R^{24\times d}$ to denote the corresponding temporal latent factor matrices, where d is the latent vector dimension.

To aggregate several temporal factors, we define an operator $A(\cdot) : R^d \times R^d \times R^d \to R^d$ to combine different temporal features. Take "2011-04-05 18:10:23" as an example, its label ids at month, day of week, and hour are 3 (April), 2 (Tuesday), 18 (after 18:00). Hence its temporal latent factor is formulated as $A(T_{1,3}, T_{2,2}, T_{3,18})$. It is important to note that our scheme is flexible: we are able to ignore one feature by taking away a slice, or introduce a new feature by adding a new slice.

Memory Reducing Trick. We reduce the input data size through a binary coding trick. We employ one label id instead of three to represent the three slices. In detail, we use 4 bits to represent the month, 3 bits to represent the day of week, and 5 bits to represent the hour slot. So the time label id can be represented by an integer of 16 bits. For instance, "2011-04-05 18:10:23" could be coded as "0011 010 10010", and its label id is 850. When we learn the model, we transform the label id into binary representation and find the corresponding label to each slice. After labeling the time, we are able to model the temporal effect from a user-time-POI latent factorization model; see details in Section 4.4.

4.4 Method

In this section, we first demonstrate the ATTF model. Then we give the detailed model inference and learning procedure. Finally, we summarize the model discussion.

4.4.1 Aggregated Temporal Tensor Factorization Model

Denote that $\mathcal{U}$ is the set of users and $\mathcal{L}$ is the set of POIs. In addition, $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ are the set of months, days of week, and hours respectively. Further we define $\mathcal{T}$ as the set of time label tuples, consisting of elements $t := (t_1, t_2, t_3)$, namely the temporal representation at different scales. The ATTF model estimates the preference of a user u at a POI l given a specific time label t through a score function $f(u, t, l)$, where $u \in \mathcal{U}$ is user id, $t \in \mathcal{T}$ is a time label tuple, and $l \in \mathcal{L}$ is POI id.
We are typically given N training examples $(u_i, t_i, l_i) \in \{1, ..., |\mathcal{U}|\} \times \{1, ..., |\mathcal{T}|\} \times \{1, ..., |\mathcal{L}|\}, i = 1, 2, ..., N$, and correspondingly outputs $y_i \in R, i = 1, 2, ..., N$. Here, (u_i, t_i, l_i) is the index of a particular element of a user-time-POI tensor, and y_i is the preference score of the user at the POI given the time label. One could simply collate the training data to build a suitable tensor, so the training task turns to fill in the blank entries of the tensor.
We exploit the Pairwise Interaction Tensor Factorization (PITF) [75] model to decompose the user-time-POI tensor. PITF model that turns out to be successful in the ECML/PKDD Discovery Challenge, runs much faster than other tensor factorization methods and has better performance in a large scale prediction task [75]. Thus the score of a POI l given user u and time

t is factorized into three interactions: user-time, user-POI, and time-POI, where each interaction is modeled through the latent vector product. Further, we infer the model via Bayesian Personalized Ranking (BPR) criteria [73] that is a general framework to train a recommendation system from implicit feedback. Because prior work [44, 128, 127] indicates that treating the check-ins as implicit feedback is better than explicit ways for POI recommendation. Since we recommend POIs for users at specific time, any candidate POI has the same user-time interaction. As a result, the preference score is independent of the user-time interaction. Then the score function for a given time label t, user u, and a target POI l could be formulated as :

$$f(u,t,l) = \langle U_u^{(L)}, L_l^{(U)} \rangle + \langle A(T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)}), L_l^{(T)} \rangle, \quad (4.1)$$

where $\langle \cdot \rangle$ denotes the vector inner product, $A(\cdot)$ is the aggregate operator. Suppose that d is the latent vector dimension, $U_u^{(L)} \in R^d$ is user u's latent vector for POI interaction, $L_l^{(U)}, L_l^{(T)} \in R^d$ are POI l's latent vectors for user interaction and time interaction, $T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)} \in R^d$ are time t's latent vector representations in three aspects: month, day of week, and hour. Aggregate operator combines the several temporal features together. We propose a linear convex combination operator. It is formulated as follows,

$$A(\cdot) = \alpha_1 \cdot T_{1,t_1}^{(L)} + \alpha_2 \cdot T_{2,t_2}^{(L)} + \alpha_3 \cdot T_{3,t_3}^{(L)}, \quad (4.2)$$

where α_1, α_2, and α_3 denote the weights of each temporal factor, which satisfy $\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\alpha_1, \alpha_2, \alpha_3 >= 0$.

4.4.2 Learning

We infer the model via BPR criteria [73], which treats the check-in activity as a kind of implicit feedback. Namely, we assume the user prefers the visited POIs than the unvisited. We treat the visited POIs as positive and the unvisited as negative. Then, we suppose that the score of $f(u,t,l)$ at positive observations is higher than the negative POIs, given u and t. Further, we formulate the relation that user u prefers a positive POI l_i than a negative one l_j at time t as follows

$$l_i >_{u,t} l_j. \tag{4.3}$$

Based on the pairwise preference defined above, we suffice to extract the set of preference constraints from the training examples

$$D_S := \{(u,t,l_i,l_j)|l_i >_{u,t} l_j, u \in \mathcal{U}, t \in \mathcal{T}, l_i, l_j \in \mathcal{L}\}. \tag{4.4}$$

For simplicity, we denote $y_{u,t,l} = f(u,t,l)$. Then for any quadruple in D_S, it satisfies $y_{u,t,l_i} > y_{u,t,l_j}$. Using a logistic function to model this relation, we get

$$p(l_i >_{u,t} l_j) := \sigma(y_{u,t,l_i} - y_{u,t,l_j}), \tag{4.5}$$

which measures the probability of l_i is a positive observation and l_j is a negative observation for user u at time t. In Eq. (4.5), σ is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$.
Suppose the quadruples in D_S are independent of each other, then learning the ATTF model is to maximize the likelihood of all the pair orders

$$\arg\max_{\Theta} \prod_{(u,t,l_i,l_j)\in D_S} p(l_i >_{u,t} l_j), \tag{4.6}$$

where Θ is the parameters to learn, namely $U^{(L)}$, $L^{(U)}$, $L^{(T)}$, $T_1^{(L)}$, $T_2^{(L)}$, and $T_3^{(L)}$. The objective function is equivalent to minimizing the negative log likelihood. To avoid the risk of overfitting, we add a Frobenius norm term to regularize the parameters. Then the objective function is

$$\arg\min_{\Theta} \sum_{(u,t,l_i,l_j)\in D_S} - ln(\sigma(y_{u,t,l_i} - y_{u,t,l_j})) + \lambda_\Theta ||\Theta||_F^2, \tag{4.7}$$

where λ_Θ is the regularization parameter.
We leverage the Stochastic Gradient Decent (SGD) algorithm to learn the objective function for efficacy. First, we define $y_{u,t,l_p,l_n} = y_{u,t,l_p} - y_{u,t,l_n}$, which models the pairwise relation in D_S. Further we denote a common part in gradient decent values for all parameters as $\delta = 1 - \sigma(y_{u,t,l_p,l_n})$. As $T^{(L)}_{1,t_1}$, $T^{(L)}_{2,t_2}$, and $T^{(L)}_{3,t_3}$ are symmetric, they have the same gradient form. For simplicity, we use $T^{(L)}_t \in \{T^{(L)}_{1,t_1}, T^{(L)}_{2,t_2}, T^{(L)}_{3,t_3}\}$ to represent any of them, $\alpha \in \{\alpha_1, \alpha_2, \alpha_3\}$ to denote corresponding weight, and $A(\cdot)$ to denote $A(T^{(L)}_{1,t_1}, T^{(L)}_{2,t_2}, T^{(L)}_{3,t_3})$. Then the updating rule for the parameters is as follows,

$$\begin{aligned}
&U^{(L)}_u \leftarrow U^{(L)}_u + \gamma \cdot (\delta \cdot (L^{(U)}_{l_p} - L^{(U)}_{l_n}) - \lambda \cdot U^{(L)}_u),\\
&L^{(U)}_{l_p} \leftarrow L^{(U)}_{l_p} + \gamma \cdot (\delta \cdot U^{(L)}_u - \lambda \cdot L^{(U)}_p),\\
&L^{(T)}_{l_p} \leftarrow L^{(T)}_{l_p} + \gamma \cdot (\delta \cdot A(\cdot) - \lambda \cdot L^{(T)}_{l_p}),\\
&L^{(U)}_{l_n} \leftarrow L^{(U)}_{l_n} - \gamma \cdot (\delta \cdot U^{(L)}_u + \lambda \cdot L^{(U)}_n),\\
&L^{(T)}_{l_n} \leftarrow L^{(T)}_{l_n} - \gamma \cdot (\delta \cdot A(\cdot) + \lambda \cdot L^{(T)}_{l_n}),\\
&T^{(L)}_t \leftarrow T^{(L)}_t + \gamma \cdot (\delta \cdot \alpha \cdot (L^{(T)}_{l_p} - L^{(T)}_{l_n}) - \lambda \cdot T^{(L)}_t),
\end{aligned} \tag{4.8}$$

where γ is the learning rate, λ is the regularization parameter. To train the model, we use the bootstrap skill to draw the quadruple from D_S, following [73]. Algorithm 3 gives the detailed procedure to learn the ATTF model. We aim to learn the latent representations of user, temporal features, and POIs, namely $U^{(L)}, T^{(L)}_1, T^{(L)}_2, T^{(L)}_3, L^{(U)}, L^{(T)}$. Let $|\mathcal{U}|$ denote the number of

ALGORITHM 3: ATTF model learning algorithm

Input: Training tuples $\{(u_i, t_i, l_i)\}_{i=1,...,N}$
Output: $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$
Initialize $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$
Uniformly sample $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$ check-in tuples from D_S to generate D_e for loss calculation
for *iterations* **do**
  //S is the number of sampled check-ins
  for $i \in [1, S]$ **do**
    Draw (u, t, l_p) uniformly from training tuples
    // k is the number of negative samples
    for $n = 1, 2, ..., k$ **do**
      Draw l_n uniformly to form (u, t, l_p, l_n)
      $y_{u,t,l_p,l_n} \leftarrow y_{u,t,l_p} - y_{u,t,l_n}$
      $\delta \leftarrow 1 - \sigma(y_{u,t,l_p,l_n})$
      Update parameters according to Eq. (6.8)
    end
  end
  Estimate the loss defined on D_e
end
Return $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$

users, then we generate about $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$ tuples from D_S to generate a tuple set D_e for the loss estimation, namely the negative log likelihood value. We follow the implementation of BPRMF [73] in MyMediaLite[2] to set the number of samples for loss estimation as $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$. In each iteration, we sample S check-ins and then generate negative samples to learn the model. After that, we calculate the loss value over D_e: $\sum\limits_{(u,t,l_i,l_j) \in D_e} - ln(\sigma(y_{u,t,l_i} - y_{u,t,l_j})) + \lambda_\Theta ||\Theta||_F^2$. The convergent condition is satisfied when the loss value for the fixed sampled tuples does not decrease.

[2]http://www.mymedialite.net/index.html

Complexity. The runtime for predicting a triple (u, t, l) is in $O(d)$, where d is the number of latent vector dimension. The updating procedure is also in $O(d)$. Hence training a quadruple is in $O(d)$, then training an example (u, t, l) is in $O(k \cdot d)$, where k is the number of sampled negative POIs. For each iteration, we sample S training examples. The calculation cost for loss estimation is less than the training procedure. Therefore, training the model costs $O(I \cdot S \cdot k \cdot d)$, where I is the number of iterations. In practical, I is always small for different datasets, in the range of $[5, 30]$.

4.4.3 Model Discussion

The ATTF model can be treated as a linear combination of two matrix factorization models which learn user preference and temporal effect respectively, as shown in Eq. (4.1). The first term depicts the user-POI interaction, which is similar as the low rank matrix factorization for the user-POI matrix through collaborative filtering technique. The second term depicts the time-POI interaction, which acts like leveraging a latent factor model to describe the relations between time labels and POIs. Further, the aggregate operator $A(\cdot)$ combines several temporal factors together.

Two points are important to note for our model: (1) The ATTF model and the time labeling scheme are a general framework to subsume several temporal characteristics together. We take three common ones in this work, but it is easy to add others, e.g., different days in a month, workdays and vocations in a year. (2) Even though the model equation for ATTF in POI recommendation suffices to be expressed by a combination of two MF models, it is different from a simple ensemble of two MF

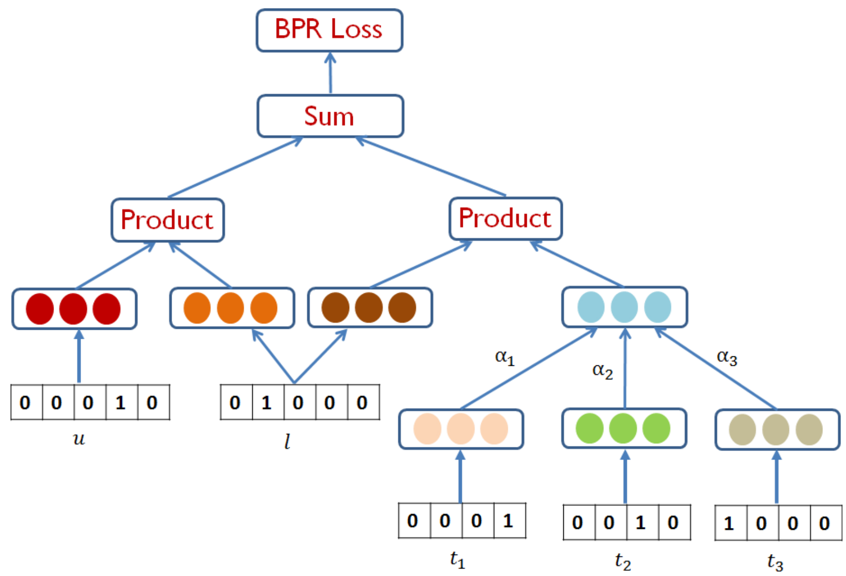


Figure 4.5: Embedding neural network for ATTF model

model recommendation results because in our case the model parameters are learned jointly. Thus the learned parameters jointly represent the user preference and temporal effect. It better reflects the fact that user check-in behavior is a complex decision under many conditions.

The ATTF model can also be interpreted from the embedding neural network perspective. The embedding network, e.g., word2vec framework [61], has turned out to be a successful semi-supervised learning method in natural language processing [68, 36], network embedding [69, 89, 24], and recommendation systems [102, 88]. Moreover, recent studies [39, 42] show that the neural word embeddings can be treated as a kind of matrix factorization method [34]. This equivalence between neural embeddings and the latent factor models inspires us to understand our ATTF model from the embedding neural network perspective. Figure 4.5 demonstrates the equivalent embedding neural network for the ATTF model. The input layer is the one-hot representation for user, POI, and temporal information. The second layer is the embedding layer, which projects the one-hot vector as a continuous latent vector in the Euclidean subspace. Next, we exploit the product and sum operation to represent the check-in preference as $\langle U_u^{(L)}, L_l^{(U)}\rangle + \langle A(T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)}), L_l^{(T)}\rangle$, equivalent to Eq. (4.1). Finally, we construct a BPR loss layer to learn the embedding representations.

4.5 Experiment

We conduct systematical experiments to seek the answers of the following questions: 1) how the proposed ATTF model performs comparing with state-of-the-art models? 2) whether the ATTF model is better than single temporal factor models? 3) how the parameters affect the model performance?

4.5.1 Data Description and Experimental Setting

Two real-world datasets are used in the experiment: one is Foursquare data from January 1, 2011 to July 31, 2011 provided in [18] and the other is Gowalla data from January 1, 2011 to September 31, 2011 in [122]. We filter the POIs checked-in by less than 5 users and then choose users who check-in more than 10 times as our samples. After the preprocessing, the datasets contain the statistical properties as shown in Table 4.1. We randomly choose 80% of each user's check-ins as training data, and the remaining 20% for test data. Moreover, we use each check-in (u, t, l) in training data to learn the latent features of user, time, and POI. Then given the (u, t), we estimate the score value of different candidate POIs, select the top N candidates, and compare them with check-in tuples in test data.

Table 4.1: Statistics of datasets

Source	**Foursquare**	**Gowalla**
#users	10,180	3,318
#POIs	16,561	33,665
#check-ins	867,107	635,600
Avg. #check-ins each user	85.2	191.6
Avg. #POIs each user	24.3	104.1
Avg. #users each POI	14.9	10.3
Density [3]	0.0015	0.003

4.5.2 Performance Metrics

In this work, we leverage three metrics to evaluate the model performance–*precision*, *recall*, and *F-score*. The precision and recall in the top-K recommendation system are denoted as P@K and R@K respectively. P@K measures the ratio of recovered POIs to the K recommended POIs, and R@K means the ratio of recovered POIs to the set of POIs in the testing data. For each user $u \in \mathcal{U}$, $\mathcal{L}^T(u)$ denotes the set of correspondingly visited POIs in the test data, and $\mathcal{L}^R(u)$ denotes the set of recommended POIs. Then the definitions of P@K and R@K are formulated as follows

$$P@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}^R(u) \cap \mathcal{L}^T(u)|}{K}, \tag{4.9}$$

$$R@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}^R(u) \cap \mathcal{L}^T(u)|}{|\mathcal{L}^T(u)|}. \tag{4.10}$$

Further, *F-score* is the harmonic mean of precision and recall. So the *F-score* is defined as

$$F\text{-}score@K = \frac{2 * P@K * R@K}{P@K + R@K}. \tag{4.11}$$

[3]Density means the fraction of checked-in entries over all entries in user-POI matrix.

4.5.3 Baselines

We compare our ATTF model with state-of-the-art collaborative filtering (CF) methods and POI recommendation methods incorporating temporal effect. Prior work [128, 127] indicates that treating the check-ins as implicit feedback is better to recommend POIs. Hence we exploit Weighted Regularized Matrix Factorization (WRMF) [30, 66] and Bayesian Personalized Ranking Matrix Factorization (BPR-MF) [73] as comparative CF methods. To illustrate the efficacy of our ATTF model, we compare it with LRT [15] and FPMC-LR [9] which are state-of-the-art POI recommendation methods incorporating temporal effect.

- **WRMF.** The WRMF model is designed for processing large scale implicit feedback data. We define the weight mapping of user u_i at POI l_j as $w_{i,j} = (1+10 \cdot C_{i,j})^{0.5}$, where $C_{i,j}$ is the check-in counts, following the setting in [54].

- **BPR-MF.** The BPR-MF model is a popular MF-based recommendation method to learn the pairwise relation, in which users prefer the observed items than the unobserved.

- **LRT.** The LRT model is designed to modeling the "non-uniformnes" and "consecutiveness" in a matrix factorization model for POI recommendation.

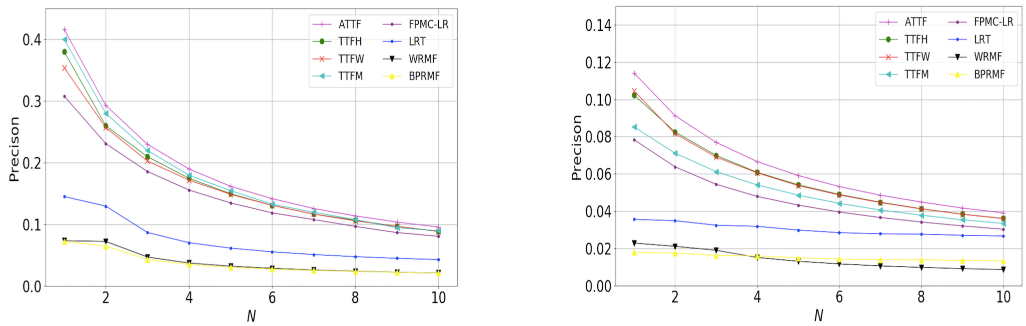
- **FPMC-LR.** The FPMC-LR model adds the Local Region constraint (i.e., geographical information) in the Factorized Personalized Markov Chain (FPMC) model [74]. FPMC-LR incorporates the geographical information and temporal consecutiveness through a local region constraint and the FPMC model respectively.

Moreover, to demonstrate the advantage of ATTF in aggregating several temporal latent factors, we also compare with three single temporal latent factor models: **TTFM**, **TTFW**, and **TTFH**. They are typically PITF model, that correspondingly considering the month, day of week, and hour as a temporal latent factor. Because these three models are the subset of our ATTF model, we attain their results by setting the corresponding weight as 1, and others as 0 in ATTF.

4.5.4 Experimental Results

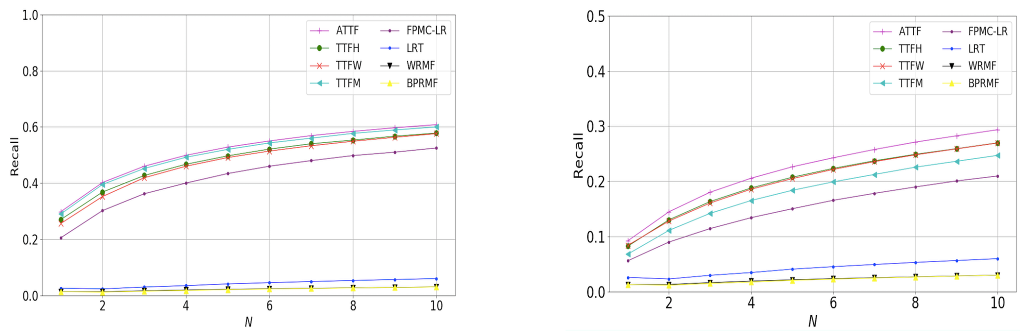
Performance Comparison

In the following, we demonstrate the performance comparison on precision, recall and F-score. We set the latent factor dimension as 60 for all compared models. We leverage grid search method to find the best weights in ATTF model. α_1, α_2, and α_3 are constrained in the range of [0, 1]. In the grid search method, we first change α_1 from zero to one with step size 0.1. Then, for each α_1 value, for instance $\alpha_1 = 0.1$, we change α_2 from zero to $1-\alpha_1$ with step size 0.1. α_3 can be calculated by $1-\alpha_1-\alpha_2$. The grid search method tries all value combinations with step size 0.1 satisfying the constraints $\alpha_1+\alpha_2+\alpha_3 = 1$, and $\alpha_1, \alpha_2, \alpha_3 >= 0$. As a result, the ATTF model on Foursquare data achieves the best result when $\alpha_1 = 0.7$, $\alpha_2 = 0.1$, and $\alpha_3 = 0.2$, while the ATTF model on Gowalla data achieves the best when $\alpha_1 = 0.2$, $\alpha_2 = 0.1$, and $\alpha_3 = 0.7$.
Figure 4.6, Figure 4.7, and Figure 4.8 show the experimental results for Foursquare and Gowalla data on measurement precision, recall, and F-score respectively. We see that 1) Our proposed ATTF model outperforms state-of-the-art CF



(a) Foursquare (b) Gowalla

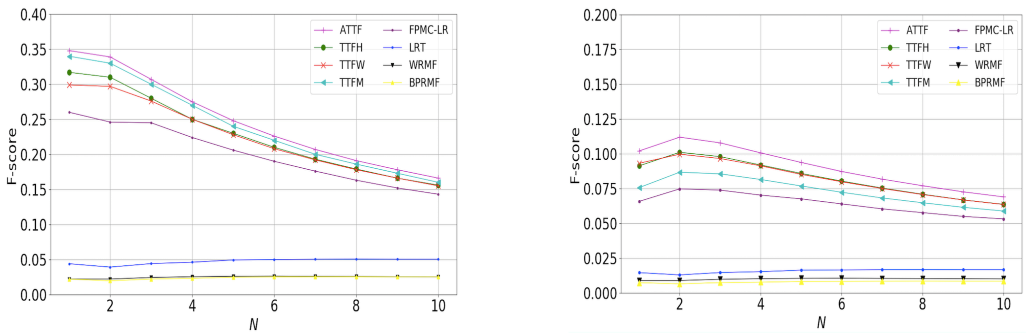
Figure 4.6: Precision on Foursquare and Gowalla



(a) Foursquare (b) Gowalla

Figure 4.7: Recall on Foursquare and Gowalla

methods and POI recommendation models. Compared with the best state-of-the-art competitor in POI recommendation area (e.g., FPMC-LR), the ATTF model gains more than 20% enhancement on Foursquare data, and more than 36% enhancement on Gowalla data for all three measures, Precision@5, Recall@5, and F-score@5. We observe that models perform better on Foursquare data than Gowalla data, even though it is sparser. The reason lies in that Gowalla data contain much more POIs and a large candidate POI set makes the recommendation harder. 2) The ATTF model outperforms single temporal factor models. Compared with best single temporal factor model, the



(a) Foursquare (b) Gowalla

Figure 4.8: F-score Foursquare and Gowalla

ATTF model gains about 3% enhancement on Foursquare data, and about 10% improvement on Gowalla data in a top-5 POI recommender system. So when data are denser, the ATTF model gets advantages. Because the ATTF model uses a tuple to represent a time spot, which gives more precise information. Dense data strengthen this precise labeling scheme. In addition, different weight assignments on both data give us two interesting insights: (i) When data are sparse, the temporal feature on month dominates the POI recommendation performance. Because check-ins on hour or day of week are sparse as shown in Figure 4.3, then the corresponding characteristics are not easily caught. The Foursquare dataset has high weight on month temporal factor. However, when data are denser, check-ins on hour are not so sparse. So the temporal characteristic on hour of day becomes prominent. (ii) We usually pay much attention to temporal characteristics on hour of day and day of week. Our experimental results indicate that the temporal characteristic on month is important, especially for sparse data. 3) Our proposed ATTF model, single temporal factorization models (e.g., TTFM, TTFW, and TTFH), and FPMC-LR perform much better than

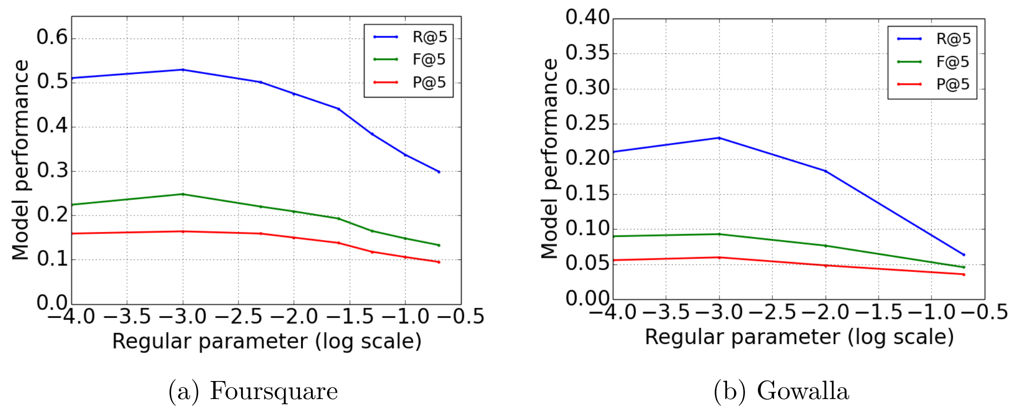


Figure 4.9: The effect of regularization parameter λ

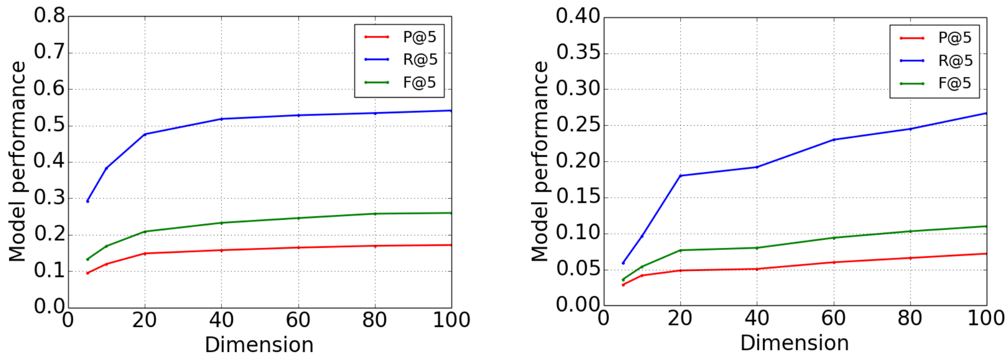
other competitors, especially at recall measure. They try to recommend POIs at more specific situations, which is the key point to improve performance. Our models recommend a user POIs at some specific time, and FPMC-LR recommends POIs given a user's recent checked-in POIs; while, the other three models give general recommendations.

Parameter Effect

The regularization parameter and latent vector dimension are two important factors affecting the model performance. We explore how they affect the proposed model in the condition of other parameters fixed.

Figure 4.9 demonstrates the effect of regularization parameter on model performance. For simplicity, we set the same parameter λ for all latent vectors. The regularization part does not significantly affect the model. The model achieves the best performance at 0.001. With the increasing of λ, the performance decreases.

Figure 4.10 demonstrates how the latent vector dimension



(a) Foursquare (b) Gowalla

Figure 4.10: The effect of latent factor dimension

affects the model. The performance of ATTF steadily rises with the increase of latent vector dimension. For the trade-off of high performance and low computation cost, we suggest setting dimension $d = 60$.

4.6 Conclusion

In this chapter, we propose the ATTF model for POI recommendation. The proposed model introduces time factor to model the temporal effect in POI recommendation, subsuming all the three temporal properties: periodicity, consecutiveness, and non-uniformness. Moreover, the ATTF model captures the temporal influence at different time scales through aggregating several time factors' contributions. Experimental results on two real-world datasets show that the ATTF model outperforms state-of-the-art models. Our model is a general framework to aggregate several temporal characteristics at different scales.

□ **End of chapter.**

Chapter 5

Geo-Teaser: Geo-Temporal Sequential Embedding Rank for POI Recommendation

POI recommendation is an important application for LBSNs, which learns the user preference and mobility pattern from check-in sequences to recommend POIs. Previous studies show that modeling the sequential pattern of user check-ins is necessary for POI recommendation. Markov chain model, recurrent neural network, and the word2vec framework are used to model check-in sequences in previous work. However, all previous sequential models ignore the fact that check-in sequences on different days naturally exhibit the various temporal characteristics, for instance, "work" on `weekday` and "entertainment" on `weekend`. In this chapter, we take this challenge and propose a **Geo-Te**mpor**a**l **s**equential **e**mbedding **r**ank (Geo-Teaser) model for POI recommendation. Inspired by the success of the word2vec framework to model the sequential contexts, we propose a *temporal POI embedding* model to learn POI representations under some particular temporal state. The temporal POI embedding model captures the contextual check-in information in sequences and the various temporal characteristics on different days as well. Furthermore, we propose a new way to incorporate the geographical influence into the pairwise preference ranking method through discriminating the unvisited POIs according to geographical information. Then we develop a geographically hierarchical pairwise preference ranking model. Finally, we propose a unified framework to recommend POIs combining these two models. To verify the effectiveness of our proposed method, we conduct experiments on two real-life datasets. Experimental results show that the Geo-Teaser model outperforms state-of-the-art models.

5.1 Introduction

LBSNs such as Foursquare and Facebook Places have become popular services to attract users sharing their check-in behaviors, making friends, and writing comments on POIs. With the prosperity of LBSNs, POI recommendation comes out to improve the user experience, which mines users' check-in sequences and recommends places where an individual has not been. POI recommendation not only helps users explore new interesting places in a city, but also facilitates business owners to launch advertisements to target customers. Due to the significance for users and businesses, POI recommendation has attracted much academic attention, and thus a bunch of methods has been proposed to enhance the POI recommendation system [7, 25, 107, 111].

Modeling the sequential pattern of user check-ins is necessary for POI recommendation. Because successive check-ins are usually correlated [9, 51, 105, 119]. Markov chain model, recurrent neural network, and the word2vec framework are used to model the check-in sequences in previous work. Studies in [51, 105, 119] exploit the Markov chain model to capture the successive check-ins' transitive pattern. Besides, researchers in [9, 14, 128] use the latent factor model based on the Markov chain property to model the successive check-ins' correlations. Recently, inspired by the success of deep learning, the neural network has been used to model the check-in sequences. Liu et al. [49] employ the recurrent neural network (RNN) to find the sequential correlations. The work in [52] models the check-in sequences through the word2vec framework to capture the sequential contexts. Moreover, we observe that check-in sequences on different days naturally exhibit the various temporal characteristics. For example, users always check-in at POIs around offices on `weekday` while visit shopping malls on `weekend`. However, all previous sequential models ignore the various temporal characteristics, which motivates our model.

Inspired by the success of the word2vec framework to model the sequential contexts [52], we propose a temporal POI embedding model to capture the contextual check-in information and the various temporal characteristics as well. In [52], all POIs are built as the "corpus", each POI is treated as a "word", and a user's all sequential check-ins are treated as a "sentence". Then, the word2vec framework [61] can be used to learn the POI embeddings, which contain the contextual relationships of consecutively visited POIs, showing better performance than Markov chain model. Nevertheless, the learned POI embeddings for capturing the sequential contexts cannot subsume the various temporal characteristics on different days. Moreover, the geographical influence is not considered in [52]. Studies on user

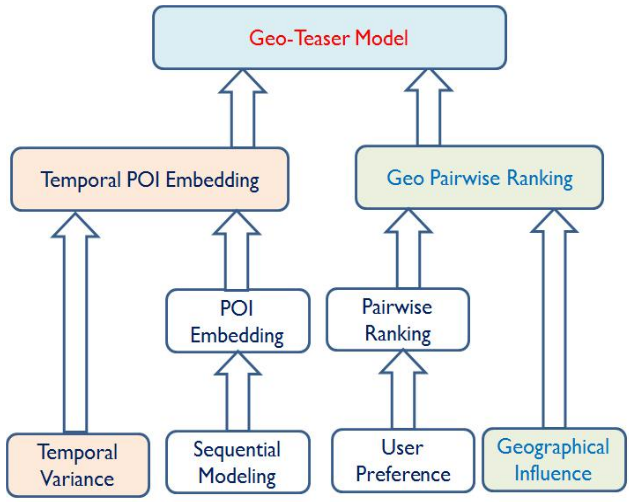


Figure 5.1: Framework of the Geo-Teaser model

mobility data show that the geographical influence is the most significant factor for POI recommendation [107, 117, 122]. Therefore, the geographical influence is expected to be incorporated to improve the POI recommendation.

To sum up, we propose the Geo-Teaser model for POI recommendation, as shown in Figure 5.1. On the one hand, we propose a temporal POI embedding model to capture the contextual check-in information and the various temporal characteristics as well. In particular, we treat one user's check-in sequence in one day as a "sentence". Then we consider each sequence under a specific temporal state and define the *temporal POI*, referring to a POI taking a specific temporal state as context. Further, we propose the temporal POI embedding model to learn POI representations and temporal state representations. On the other hand, we incorporate the geographical influence into a pairwise preference ranking model and develop a geographically hierarchical pairwise preference ranking model. Traditionally, we assume users prefer the visited POIs than the unvisited and establish a pairwise ranking model to learn user preference on POIs [41, 128]. Previous studies [7, 107] indicate that users prefer POIs that are geographically adjacent to their visited POIs. This geographical characteristic inspires us to boost the traditional pairwise ranking model through hierarchical pairwise preference relations that discriminate the unvisited POIs according to POIs' geographical information. Finally, we propose the Geo-Teaser model as a unified framework to recommend POIs combining the temporal POI embedding model and the geographically hierarchical pairwise ranking model. We summarize the contributions as follows:

1. We propose the temporal POI embedding model, which captures the check-ins' sequential contexts and the various temporal characteristics on different days. In particular, we introduce the word2vec framework to project every POI as one object in an embedding space for learning the sequential relations among POIs. Furthermore, we learn the temporal POI representations from the check-in sequence under some specific temporal state.

2. We propose a new way to incorporate the geographical influence into the pairwise preference ranking method through discriminating the unvisited POIs according to geographical information. In particular, we define a hierarchical pairwise preference relation for each user check-in: the user prefers the visited POI than the unvisited neighboring POI, and the user prefers the unvisited neighboring POI than the unvisited non-neighboring POI. Then we learn the hierarchical pairwise preference to capture the geographical influence and user preference.

3. We propose the Geo-Teaser model as a unified framework combining the temporal POI embedding model and the geographically hierarchical pairwise preference ranking model. Experimental results on two real-life datasets show that the Geo-Teaser model outperforms state-of-the-art models. Compared with the best baseline competitor, the Geo-Teaser model improves at least 20% on both datasets for all metrics.

5.2 Related Work

In this section, we first demonstrate the recent progress of POI recommendation. Then we report how the prior work exploits the sequential influence and geographical influence to improve the POI recommendation. Since our proposed method adopts an embedding learning method, the word2vec framework, to model check-in sequences, we also review the literature of the word2vec framework and its applications.

POI Recommendation. POI recommendation has attracted intensive academic attention recently. Most of the proposed methods base on Collaborative Filtering (CF) techniques to learn user preference on POIs. On the one hand, the studies in [107, 113] employ the memory-based CF to recommend POIs. The proposed system first finds some users sharing the similar check-in preference with the target user and then recommends POIs where the "similar" users have checked-in but the target user has not. Furthermore, the researchers attempt to analyze the user check-in behavior and incorporate the spatial and temporal influence to improve the recommendation performance. On the other hand, some other studies in [7, 15, 16, 40] leverage the model-based CF, i.e., the Matrix Factorization (MF) technique. They treat the POI as "item" and the check-in frequency as "rating" and establish a user-POI matrix to recommend POIs using traditional MF models. Moreover, the researchers in [44, 54] observe that it is better to treat the check-ins as implicit feedback than explicit way, namely the check-ins are similar to clicks on Webs rather than the rating on Movies. They utilize the weighted regularized MF [30] to model this kind of implicit feedback. In addition, recent work in [41, 128, 126] employs pairwise ranking models to learn the user check-in as an implicit feedback and shows the advantages of ranking methods.

Sequential Modeling. Modeling the sequential pattern is important for POI recommendation. Most of the studies employ the Markov chain property in consecutive check-ins to capture the sequential pattern. We usually categorize the POI recommendation system as generic POI recommendation and successive POI recommendation by subtle differences in the recommendation task whether to be biased to the recent check-in. The successive POI recommendation is proposed to recommend POIs given the recent check-in, which naturally attempts to model the sequential pattern from successive check-ins [9, 14, 117, 49]. Also, researchers leverage the sequential modeling to improve the generic POI recommendation. The studies in [51, 105] learn the categories' transitive pattern in sequential check-ins. Zhang et al. [119] recommend POIs by learning the transitive probability through an additive Markov chain. Recently, inspired by the success of deep learning, the neural network has been used to model the check-in sequences. Liu et al. [49] employ the recurrent neural network (RNN) to find the sequential correlations among POIs. In the meantime, the work [52] models the check-in sequences through the word2vec framework [61] to capture the sequential contexts. The success in the prior work [51, 105, 119, 49, 52] motivates us to capture the sequential pattern in user check-ins to improve the generic POI recommendation. However, all previous sequential models ignore the various temporal characteristics. Hence, we propose a temporal POI embedding method to capture the sequential POIs' correlations under different temporal states.

Geographical Influence. Geographical influence plays an important role in POI recommendation. Compared with watching movies on Netflix and online shopping in Amazon, the check-in activity is limited to the physical constraint. Hence, the check-ins usually occur in the POIs nearby the user's home and working place. This observation motivates researchers to capture the geographical influence to improve the POI recommendation. On the one hand, researchers attempt to establish geographical models to recommend POIs. First, researchers in [107, 113] discover that the distances for each pair of visited POIs in the LBSN follow the power law distribution after analyzing the geographical relations among visited POIs. Then, they propose a power law distribution model to fit the spatial relations among POIs and recommend POIs according to this kind of geographical influence [107, 113]. Moreover, researchers in [11, 7, 122] analyze each user's check-ins rather than all visited POIs and propose the Gaussian distribution based models to capture the geographical influence. Recently, Zhang et al. [117, 120] have observed that each user occupies a group of special parameters in the Gaussian mixture model. Then, they leverage the kernel density estimation to model each user's check-ins for personalization. On the other hand, instead of independently modeling the geographical influence, more researchers attempt to jointly model the geographical influence and other factors such as user preference and temporal influence together. The studies in [44, 54] incorporate the geographical influence into a weighted regularized MF model [30, 66] to learn the geographical influence and user preference together. Similar to [44, 54], we model the check-ins as a kind of implicit feedback. But we learn it through a Bayesian pairwise ranking method [73] due to its success in [128]. Furthermore, we propose a geographically hierarchical pairwise ranking model, which captures the geographical influence via discriminating the unvisited POIs according to their geographical information.

Embedding Learning. The word2vec framework [61] is an effective neural language model to learn embedding representations in word sequences. The key idea is to learn the sentence as the bag of words and represent the relations among words in the embedding subspace, such as "male"-"female"+"queen" = "king". The embedding learning technique in the word2vec framework attempts to capture the words' contextual correlations in sentences, showing better performance than the perspectives of word transitivity in sentences and word similarity. As a result, the embedding learning technique has been widely used in natural language processing recently [60, 62]. Afterwards, paragraph2vector [36] and other variants [48, 53] are proposed to enhance the word2vec framework for specific purposes. Since the efficacy of the framework in capturing the contextual correlations of items, the embedding technique based on the word2vec framework is employed to network embedding [69], as well as in user modeling [88] and item modeling [87]. To take the power of embedding learning for POI recommendation, Liu et al. [52]

Table 5.1: Data statistics

	Foursquare	**Gowalla**
#users	10,034	3,240
#POIs	16,561	33,578
#check-ins	865,647	556,453
Avg. #check-ins of each user	86.3	171.7
Avg. #POIs for each user	24.6	95.4
Avg. #users for each POI	14.9	9.2
Density	0.0015	0.0028

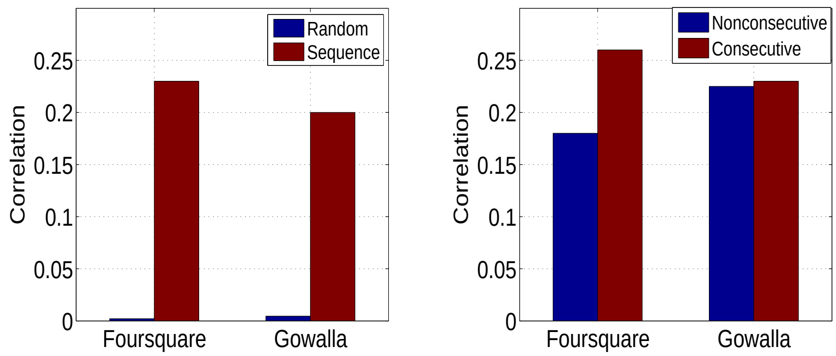
model the sequential contexts through a Skip-Gram model and achieves better performance than Markov chain model. Xie et al. [102] use similar embedding technique to recommend POIs. However, the previous work [52, 102] ignores two significant factors accounting for the check-in activity, the various temporal characteristics and geographical influence. To incorporate these two factors, we propose the Geo-Teaser model.

5.3 Data Description and Analysis

In this section, we first introduce two real-world LBSN datasets and then conduct the empirical analysis to explore the properties of check-in sequences in one day.

5.3.1 Data Description

We use two check-in datasets crawled from real-world LBSNs for data analysis. One is collected from Foursquare provided in [18] and the other is Gowalla data provided in [122]. We preprocess the data by filtering the POIs checked-in less than five users and users whose check-ins are less than ten times. Then we keep the remaining users' check-in records from January 1, 2011



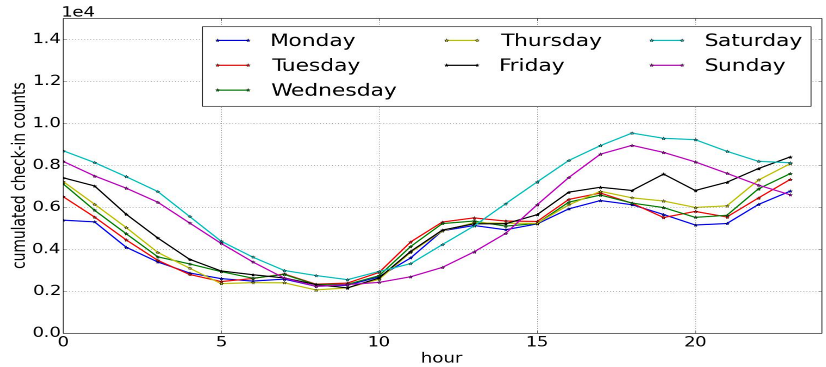
(a) Sequence vs. Random (b) Consecutive vs. Nonconsecutive

Figure 5.2: POI correlation in sequences

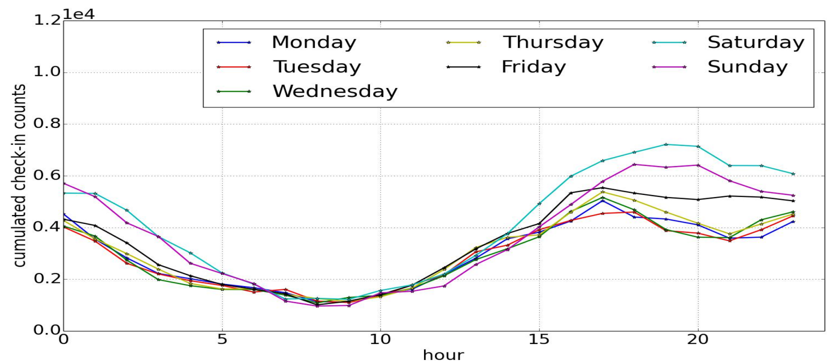
to July 31, 2011. After the preprocessing, the datasets contain the statistical properties as shown in Table 5.1.

5.3.2 Empirical Analysis

We conduct data analysis to answer the following two questions: 1) how POIs in sequences of one day correlate each other? 2) how check-in sequences perform on different days?
We investigate the correlations of POIs in sequences of one day, as shown in Figure 5.2. To calculate the correlation between two POIs, we construct the user-POI matrix according to the check-in records. Then, we measure the correlation of a POI pair regarding the Jaccard similarity of those users who have checked-in at the two POIs. In Figure 5.2(a), we calculate the average correlation value of POI pairs in sequences for all users and compare it with the average correlation value of 5,000 random POI pairs. We observe that the correlation of POIs in sequences is much higher than random pairs by about 100 times for Foursquare and 50 times for Gowalla, which motivates the sequential modeling. In Figure 5.2(b), we compare the



(a) Foursquare



(b) Gowalla

Figure 5.3: Check-in pattern at different hours over day of week

correlation of consecutive pairs with nonconsecutive pairs in sequences. Take a sequence of (l_1, l_2, l_3) as an example, (l_1, l_2) and (l_2, l_3) are consecutive pairs, and (l_1, l_3) is a nonconsecutive pair. We also calculate the average value of all sequences for all users to make the comparison. We observe that the nonconsecutive pairs contain comparable correlation to the consecutive pairs. Hence, not only consecutive POIs are highly correlated [9, 128], all POIs in a sequence are highly correlated with a contextual property. Accordingly, it is not satisfactory to only model the consecutive check-ins' transitive probability by Markov chain model or the consecutive check-ins' correlation by tensor factorization. This observation motivates us to model

the whole sequence through the word2vec framework.
We explore how the various temporal characteristics on different days affect the user's check-in behavior. Previous work [126, 128] shows that user check-ins exhibit different patterns on different days, especially for working days and weekends. Figure 5.3 demonstrates the number of cumulated check-ins for all users at different hours on different days of a week, from Monday to Sunday. From the statistics of cumulated check-ins in Figure 5.3, we observe the day of week check-in pattern at different hours: users take more check-ins in the late afternoon and the evening from 16:00 p.m. to 3:00 a.m. on weekends than the weekdays. Hence, Saturday and Sunday take the similar pattern, while the days from Monday to Friday take the similar pattern that is different from the weekends. We may infer that `weekday` and `weekend` exert two types of effects on the user's check-in behavior. Therefore, modeling the sequence pattern should contain this temporal feature.

5.4 Method

In this section, we first propose the temporal POI embedding model to capture the various temporal characteristics for sequential modeling. Next, we demonstrate the geographically hierarchical pairwise preference ranking model. Then, we propose the Geo-Teaser model as a unified framework to recommend POIs combining the temporal POI embedding model and the geographically hierarchical pairwise preference ranking model. Finally, we show the learning procedures for the Geo-Teaser model.

5.4.1 Temporal POI Embedding

We propose a temporal POI embedding method to learn the sequential pattern, which captures POIs' contextual information from user check-in sequences and as well as the various temporal characteristics. Different from the work [52] that treats a user's all check-ins as a "sentence", we treat a user's check-ins of one day as a "sentence". Because consecutive check-ins on different days may span a long time and be not highly correlated. Further, we assume that check-in sequences on different days exhibit various temporal characteristics. Then, we learn POI embeddings in a sequence with some specific temporal state.
To better describe the model, we present some basic concepts as follows.

Definition 7 (Check-in) *A check-in is a triple $\langle u, l, t\rangle$ that depicts a user u visiting POI l at time t.*

Definition 8 (Check-in sequence) *A check-in sequence is a set of check-ins of user u in one day, denoted as $S_u = \{\langle l_1, t_1\rangle, \ldots, \langle l_n, t_n\rangle\}$, where t_1 to t_n belong to the same day. For simplicity, we denote $S_u = \{l_1, \ldots, l_n\}$.*

Definition 9 (Target POI and context POI) *In a sequence S_u, the chosen l_i is the target POI and other POIs in S_u are context POIs.*

We propose the temporal POI embedding model based on the Skip-Gram model [61]. As shown in Figure 5.4, we learn the representations of context POIs from l_{i-k} to l_{i+k} given a target POI l_i and the sequence temporal state t_s. Here k is a parameter to control the context window size. In addition, the temporal state t_s is composed of two options, `weekday`

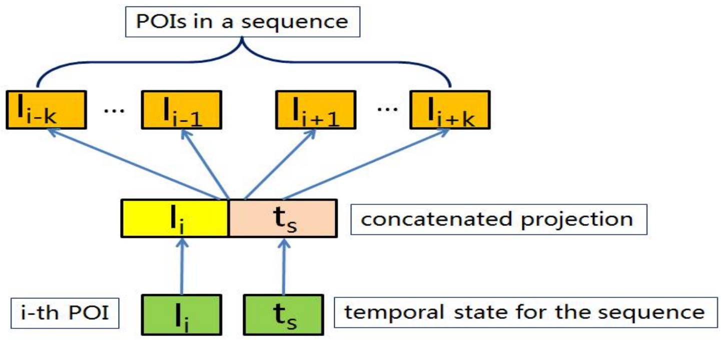


Figure 5.4: Temporal POI embedding model

and `weekend`. Because we want to discriminate `weekday` and `weekend`, which depict the various temporal characteristics on day level as shown in Figure 5.3. Formally, given a sequence S_u and its temporal state t_s, our model attempts to learn the temporal POI embeddings through maximizing the following function,

$$\mathcal{L}_{TPE} = \sum_{S_u \in S} \frac{1}{|S_u|} \sum_{l_i \in S_u} \sum_{-k \leq c \leq k, c \neq 0} \big(\log \Pr(l_{i+c} | l_i, t_s) \big), \tag{5.1}$$

where S is a set containing all sequences S_u for all users. $\mathcal{L}_{TPE}$ aims to maximize the context POI's conditional occurrence likelihood for all sequences.

Furthermore, we formulate the probability $\Pr(l_{i+c}|l_i, t_s)$ using a softmax function. For better description, we introduce two symbols, defined as follows: $\hat{\mathbf{l}}'_c = \mathbf{l}'_c \oplus \mathbf{l}'_c$, $\mathbf{l}^t_i = \mathbf{l}_i \oplus \mathbf{t}_s$, where $\oplus$ is the concatenation operator, and $\mathbf{l}'_c$, $\mathbf{l}_i$, and $\mathbf{t}_s$ are latent vectors of output layer context POI, target POI, and temporal state, respectively. Thus, we get $\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i = \mathbf{l}'_c \cdot \mathbf{l}_i + \mathbf{l}'_c \cdot \mathbf{t}_s$. Therefore, the probability $\Pr(l_{i+c}|l_i, t_s)$ can be formulated as,

$$\Pr(l_{i+c}|l_i, t_s) = \frac{\exp(\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i)}{\sum_{l_i \in L} \exp(\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i)}. \tag{5.2}$$

As the size of set L in Eq. (5.2) is large, we exploit the negative sampling technique [61] to learn the model efficiently. Then, the objective function can be formulated in a new form easier to optimize,

$$\mathcal{L}_{TPE} = \sum_{S_u \in S} \frac{1}{|S_u|} \sum_{l_i \in S_u} \sum_{-k \leq c \leq k, c \neq 0} \Big(\log \sigma(\hat{\mathbf{l}}_c' \cdot \mathbf{l}_i^t) + \sum_{h} E_{k'} \log \sigma(-\hat{\mathbf{l}}_{k'}' \cdot \mathbf{l}_i^t) \Big), \tag{5.3}$$

where $l_{k'}$ is the sampled negative POI, h is the number of negative samples, $\sigma(\cdot)$ is the sigmoid function, and $E(\cdot)$ means to calculate the expectation value for all generated negative samples. Here we adopt the same strategy in [61], namely using a unigram distribution, to draw the negative samples.

5.4.2 Geographically Hierarchical Pairwise Ranking

We propose the geographically hierarchical pairwise preference ranking model, which incorporates the geographical influence into a pairwise ranking model. The check-in activity is observed as a kind of implicit feedback similar to the web clicks [44, 54]. To learn this implicit feedback, we leverage the Bayesian personalized ranking (BPR) criteria [73] to learn the user preference on POIs. BPR is a pairwise ranking model, which learns the pairwise user preference based on the assumption that users prefer the visited POIs than the unvisited. In our geographically hierarchical pairwise ranking model, we discriminate the unvisited POIs using POIs' geographical information. Previous studies [7, 113, 122] observe that users prefer the POIs nearby the visited than POIs far away, we can discriminate the unvisited POIs and define *neighboring POI* and *non-neighboring POI* as follows.

Definition 10 (Neighboring POI and non-neighboring POI) *For each check-in $\langle u, l_i \rangle$, the neighboring POI is the POI whose distance from l_i is less than or equal to a threshold s, while the non-neighboring POI is the POI whose distance is more than s.*

Furthermore, for each check-in $\langle u, l_i \rangle$, we define a hierarchical pairwise preference relation: the user prefers the visited POI l_i than the unvisited neighboring POI l_{ne}, and prefers the unvisited neighboring POI l_{ne} than the unvisited non-neighboring POI l_{nn}. Denote $d(l_i, l_j)$ as the distance of two POIs l_i and l_j, we represent the hierarchical pairwise preference relation for check-in $\langle u, l_i \rangle$ as follows,

$$l_i >_{u,d(l_i,l_{ne})\leq s} l_{ne} \ \vee \ l_{ne} >_{u,d(l_i,l_{nn})>s} l_{nn}. \tag{5.4}$$

Suppose L is the set of POIs, and L_u is the visited POIs of user u, the hierarchical pairwise preference relation set for a sequence S_u satisfying Eq. (5.4) is defined as follows,

$$\begin{aligned} D_{S_u} = \{(u, l_i, l_{ne}) \vee (u, l_{ne}, l_{nn}) | l_i \in S_u, d(l_i, l_{ne}) \leq s, \\ d(l_i, l_{nn}) > s, l_{ne}, l_{nn} \in L \setminus L_u\}. \end{aligned} \tag{5.5}$$

Now learning the geographically hierarchical pairwise ranking model is equivalent to model the preference relations in D_{S_u}. Here we employ the MF model to formulate the preference score function. We use $\mathbf{l}_i^t = \mathbf{l}_i \oplus \mathbf{t}_s$ to represent the temporal POI latent vector, which is consistent with the temporal POI embedding model. In addition, we define $\hat{\mathbf{u}} = \mathbf{u} \oplus \mathbf{u}$, then the score function can be formulated as,

$$f(u, t_s, l_i) = \hat{\mathbf{u}} \cdot \mathbf{l}_i^t. \tag{5.6}$$

Next, we use the sigmoid function to formulate the pairwise preference probability. Suppose $\Pr(l_i >_u l_n)$ denotes the probability of user u prefers POI l_i than l_n, and $\sigma(\cdot)$ is the sigmoid function. Then, each pair in the preference set can be formulated as,

$$\Pr(l_i >_u l_n) = \sigma(f(u, t_s, l_i) - f(u, t_s, l_n)) = \sigma(\mathbf{u} \cdot (\mathbf{l}_i - \mathbf{l}_n)). \quad (5.7)$$

Thus, learning the geographically hierarchical pairwise ranking model is equivalent to maximize the following function,

$$\mathcal{L}_{GPR} = \sum_{S_u \in S} \sum_{(u, l_i, l_n) \in D_{S_u}} \log \sigma(\mathbf{u} \cdot (\mathbf{l}_i - \mathbf{l}_n)), \quad (5.8)$$

where S is a set containing all sequences S_u for all users and D_{S_u} is hierarchical pairwise preference relations on sequence S_u.

5.4.3 Geo-Teaser Model

We propose the Geo-Teaser model as a unified framework to recommend POIs combining the temporal embedding model and the pairwise ranking model. Learning the Geo-Teaser model is equivalent to maximize $\mathcal{L}_{TPE}$ and $\mathcal{L}_{GPR}$ together,

$$\mathcal{O} = \arg\max_{\mathbf{U}, \mathbf{L}, \mathbf{T}} \alpha \cdot \mathcal{L}_{TPE} + \beta \cdot \mathcal{L}_{GPR}, \quad (5.9)$$

where α and β are the hyperparameters to trade-off the sequential modeling and the preference learning modules. We expect to obtain the user, POI, and temporal state representations through learning the temporal POI embeddings and geographically pairwise preference relations in the Geo-Teaser model.
Substituting $\mathcal{L}_{TPE}$ and $\mathcal{L}_{GPR}$ with Eq. (5.3) and Eq. (5.8) respectively, then we can learn the Geo-Teaser model through

the following objective function,

$$\begin{aligned}\arg\max_{\mathbf{U},\mathbf{L},\mathbf{T}} \sum_{S_u \in S} \sum_{l_i \in S_u} \Big(\sum_{-k \le c \le k, c \neq 0} & \alpha \log \sigma(\mathbf{l}'_c \cdot \mathbf{l}_i) + \\ & \sum_h \alpha E_{k'} \log \sigma(-\mathbf{l}'_{k'} \cdot \mathbf{l}_i) + \\ & \sum_{D_{S_u}} \beta \log(\sigma(\mathbf{u} \cdot (\mathbf{l}_i - \mathbf{l}_n))) \Big).\end{aligned} \tag{5.10}$$

5.4.4 Learning

We use an alternate iterative update procedure and employ stochastic gradient descent (SGD) to learn the objective function. To learn the model, for each sampled training instance, we separately calculate the derivatives for $\mathcal{L}_{TPE}$ and $\mathcal{L}_{GPR}$, and then update the corresponding parameters along the ascending gradient direction,

$$\Theta^{t+1} = \Theta^t + \eta \times \frac{\partial \mathcal{O}(\Theta)}{\partial \Theta}, \tag{5.11}$$

where Θ is the training parameter and η is the learning rate.
Specifically, for a check-in $\langle u, l_i \rangle$, we calculate the stochastic gradient decent for $\mathcal{L}_{TPE}$. First, we get the updating rule for the context POI l_c,

$$\begin{aligned}\mathbf{l}_i &\leftarrow \mathbf{l}_i + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i))\mathbf{l}'_c \\ \mathbf{t}_i &\leftarrow \mathbf{t}_i + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i))\mathbf{l}'_c \\ \mathbf{l}'_c &\leftarrow \mathbf{l}'_c + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}'_c \cdot \mathbf{l}^t_i))(\mathbf{l}_i + \mathbf{t}_i).\end{aligned} \tag{5.12}$$

Then, we update the negative sample l'_k as follows,

$$\begin{aligned}\mathbf{l}_i &\leftarrow \mathbf{l}_i - \alpha\eta\sigma(\hat{\mathbf{l}}'_{k'} \cdot \mathbf{l}^t_i)\mathbf{l}'_{k'} \\ \mathbf{t}_i &\leftarrow \mathbf{t}_i - \alpha\eta\sigma(\hat{\mathbf{l}}'_{k'} \cdot \mathbf{l}^t_i)\mathbf{l}'_{k'} \\ \mathbf{l}'_{k'} &\leftarrow \mathbf{l}'_{k'} - \alpha\eta\sigma(\hat{\mathbf{l}}'_{k'} \cdot \mathbf{l}^t_i)(\mathbf{l}_i + \mathbf{t}_i).\end{aligned} \tag{5.13}$$

To update $\mathcal{L}_{GPR}$, we calculate the stochastic gradient decent for each preference pair (u, l_i, l_n) in D_{S_u}[1]. Denote $\delta = 1 - \sigma(\mathbf{u} \cdot \mathbf{l}_i - \mathbf{u} \cdot \mathbf{l}_n)$, we update the parameters as follows,

$$\begin{aligned}
\mathbf{u} &\leftarrow \mathbf{u} + \beta\eta\delta(\mathbf{l}_i - \mathbf{l}_n) \\
\mathbf{l}_i &\leftarrow \mathbf{l}_i + \beta\eta\delta\mathbf{u} \\
\mathbf{l}_n &\leftarrow \mathbf{l}_n - \beta\eta\delta\mathbf{u}.
\end{aligned} \tag{5.14}$$

Algorithm 4 shows the details of learning the Geo-Teaser model. S is the set of all sequences, and S_u is a sequence of user u. $\mathbf{U}$, $\mathbf{L}$, and $\mathbf{T}$ are feature matrices of the user, POI, and temporal state. $\mathbf{L}'$, an assistant learning parameter, is the output layer POI matrix in Skip-Gram model. We use the standard way [61] to learn the POI representations in the sequences, as shown from line 5 to line 14 in Algorithm 4. Next, we exploit the Bootstrap sampling to generate m unvisited POIs and then classify the unvisited POIs as neighboring POIs and non-neighboring POIs according to their distances from the visited POI l_i. Then, we establish the pairwise preference set D_m for each check-in $\langle u, l_i \rangle$. Here $D_m = \{(u, l_i, l_{ne}) \vee (u, l_{ne}, l_{nn}) | d(l_i, l_{ne}) \leq s, d(l_i, l_{nn}) > s, l_{ne}, l_{nn} \in L \setminus L_u\}$. Then we learn the parameters for each instance in D_m, shown from line 15 to line 25 in Algorithm 4.
After learning the Geo-Teaser model, we get the latent feature representations of users, POIs, and temporal states. Then, we can estimate the check-in possibility of user u over a candidate POI l at temporal state t_s according to the preference score function. Furthermore, we use the Eq. (5.6) for score estimation. Finally, we rank the candidate POIs and select the top N POIs with the highest estimated possibility values for each user.

[1]The pair of (u, l_i, l_n) happens in two cases: (u, l_i, l_{ne}) and (u, l_{ne}, l_{nn}) as shown in Alg. 4.

ALGORITHM 4: Learning algorithm for the Geo-Teaser model

Input: S
Output: $\mathbf{U}$, $\mathbf{L}$, $\mathbf{T}$

Initialize $\mathbf{U}$, $\mathbf{L}$, $\mathbf{L}'$, and $\mathbf{T}$ (uniformly at random)
for *iterations* **do**
    for $S_u \in S$ **do**
        for $\langle u, l_i \rangle \in S_u$ **do**
            for *each context POI* l_c **do**
                $\mathbf{l}_i \leftarrow \mathbf{l}_i + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}_c^{'} \cdot \mathbf{l}_i^t))\mathbf{l}'_c$
                $\mathbf{t}_i \leftarrow \mathbf{t}_i + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}_c^{'} \cdot \mathbf{l}_i^t))\mathbf{l}'_c$
                $\mathbf{l}'_c \leftarrow \mathbf{l}'_c + \alpha\eta(1 - \sigma(\hat{\mathbf{l}}_c^{'} \cdot \mathbf{l}_i^t))(\mathbf{l}_i + \mathbf{t}_i)$
                for $\{k' \sim P_{nc_c}\}$ **do**
                    $\mathbf{l}_i \leftarrow \mathbf{l}_i - \alpha\eta\sigma(\hat{\mathbf{l}}_{k'}^{'} \cdot \mathbf{l}_i^t)\mathbf{l}'_{k'}$
                    $\mathbf{t}_i \leftarrow \mathbf{t}_i - \alpha\eta\sigma(\hat{\mathbf{l}}_{k'}^{'} \cdot \mathbf{l}_i^t)\mathbf{l}'_{k'}$
                    $\mathbf{l}'_{k'} \leftarrow \mathbf{l}'_{k'} - \alpha\eta\sigma(\hat{\mathbf{l}}_{k'}^{'} \cdot \mathbf{l}_i^t)(\mathbf{l}_i + \mathbf{t}_i)$
                end
            end
            Uniformly sample m unvisited POIs
            for $(u, l_i, l_{ne}) \in D_m$ **do**
                $\delta = 1 - \sigma(\mathbf{u} \cdot \mathbf{l}_i - \mathbf{u} \cdot \mathbf{l}_{ne})$
                $\mathbf{u} \leftarrow \mathbf{u} + \beta\eta\delta(\mathbf{l}_i - \mathbf{l}_{ne})$
                $\mathbf{l}_i \leftarrow \mathbf{l}_i + \beta\eta\delta\mathbf{u}$; $\mathbf{l}_{ne} \leftarrow \mathbf{l}_{ne} - \beta\eta\delta\mathbf{u}$
            end
            for $(u, l_{ne}, l_{nn}) \in D_m$ **do**
                $\delta = (1 - \sigma(\mathbf{u} \cdot \mathbf{l}_{ne} - \mathbf{u} \cdot \mathbf{l}_{nn}))$
                $\mathbf{u} \leftarrow \mathbf{u} + \beta\eta\delta(\mathbf{l}_{ne} - \mathbf{l}_{nn})$
                $\mathbf{l}_{ne} \leftarrow \mathbf{l}_{ne} + \beta\eta\delta\mathbf{u}$; $\mathbf{l}_{nn} \leftarrow \mathbf{l}_{nn} - \beta\eta\delta\mathbf{u}$
            end
        end
    end
end

Scalability. For one check-in, learning the temporal embedding model costs $O(k \cdot h \cdot d)$, where k, h, and d denote the context window size, the number of negative samples, and the latent vector dimension, respectively. For the pairwise preference learning from line 15 to 25 in Algorithm 4, we sample m unvisited POIs, which can generate maximum $O(m^2)$ pairwise preference tuples. For each check-in, the learning procedures cost $O(m^2 \cdot d)$. Therefore, the complexity of our model is $O\big((k \cdot h + m^2) \cdot d \cdot |C|\big)$, where C is the set of all check-ins. For k, h, m, and d are fixed hyperparameters, the proposed model can be treated as linear in $O(|C|)$. Furthermore, in order to make our model more efficient, we turn to the asynchronous stochastic gradient descent (ASGD) [71] and parallelly run the algorithm in an unlock way. As the check-in frequency distribution of POIs in LBSNs follows a power law [107], this results in a long tail of infrequent POIs, which guarantees to employ the ASGD to parallel the parameter updates.

5.5 Experimental Evaluation

We conduct experiments to seek the answers to the following questions: 1) how the Geo-Teaser model performs comparing with state-of-the-art recommendation methods? 2) how each component (i.e., the various temporal characteristics and geographical influence) affects the model performance? 3) how the parameters affect the model performance?

5.5.1 Experimental Setting

Two real-world datasets are used in the experiment: one is from Foursquare provided in [18] and the other is from Gowalla in [122]. Table 5.1 demonstrates the statistical information of the datasets. In order to make our model satisfactory to the scenario of recommending for future check-ins, we choose the first 80% of each user's check-ins as training data, the remaining 20% for test data, following [9, 119].

5.5.2 Performance Metrics

In this work, we compare the model performance through *precision* and *recall*, which are generally used to evaluate a POI recommendation system [15, 41]. To evaluate a top-N recommendation system, we denote the precision and recall as P@N and R@N, respectively. In our POI recommendation task, P@N measures the ratio of recovered POIs to the N recommended POIs, and R@N means the ratio of recovered POIs to the set of POIs in the test data. Then we calculate the average precision and recall over all users for evaluation. Supposing $L_{visited}$ denotes the set of correspondingly visited POIs in the test data, and $L_{N,rec}$ denotes the set of recommended POIs, the definitions of P@N and R@N are formulated as follows,

$$P@N = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{visited} \cap L_{N,rec}|}{N}, \tag{5.15}$$

$$R@N = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{visited} \cap L_{N,rec}|}{|L_{visited}|}. \tag{5.16}$$

5.5.3 Model Comparison

Prior work [44, 54] observes that treating the check-ins as implicit feedback is better to model the user preference. Hence we compare our model with WRMF [30, 66] and BPRMF [73], which are state-of-the-art collaborative filtering models designed for capturing the implicit feedback. To illustrate the effectiveness of our model, we compare it with four state-of-the-art

POI recommendation methods: LRT [15], LORE [119], Rank-GeoFM [41], and SG-CWARP [52].

- ***BPRMF*** [73]: ***B***ayesian ***P***ersonalized ***R***anking ***M***atrix ***F***ac-torization (*BPRMF*) is a popular pairwise ranking method that models the implicit feedback data to recommend top-N items.

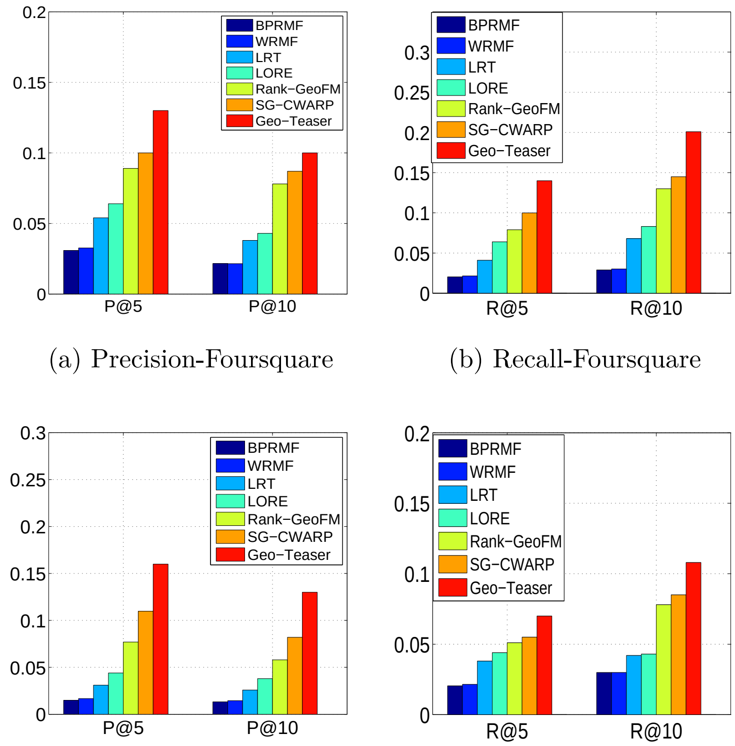
- ***WRMF*** [30, 66]: ***W***eighted ***R***egularized ***M***atrix ***F***actorization (*WRMF*) model is designed for implicit feedback ranking problem. We set the weight mapping function of user u_i at POI l_j as $w_{i,j} = (1 + 10 \cdot C_{i,j})^{0.5}$, where $C_{i,j}$ is the number of check-ins, following the setting in [54].

- ***LRT*** [15]: ***L***ocation ***R***ecommendation framework with ***T***emporal effects model (*LRT*) is a state-of-the-art POI recommendation method, which captures the temporal effect in POI recommendation.

- ***LORE*** [119]: *LORE* is state-of-the-art model that exploits the sequential influence for location recommendation. Compared with other work [9, 105], *LORE* employs the whole sequence's contribution, not only the successive check-ins sequential influence.

- ***Rank-GeoFM*** [41]: *Rank-GeoFM* is a ranking based geographical factorization method, which incorporates the geographical and temporal influence in a latent ranking model.

- ***SG-CWARP*** [52]. *SG-CWARP* is the latest work, which leverages the word2vec framework to model the check-ins for sequential contexts.



(a) Precision-Foursquare (b) Recall-Foursquare

(c) Precision-Gowalla (d) Recall-Gowalla

Figure 5.5: Model comparison

5.5.4 Experimental Results

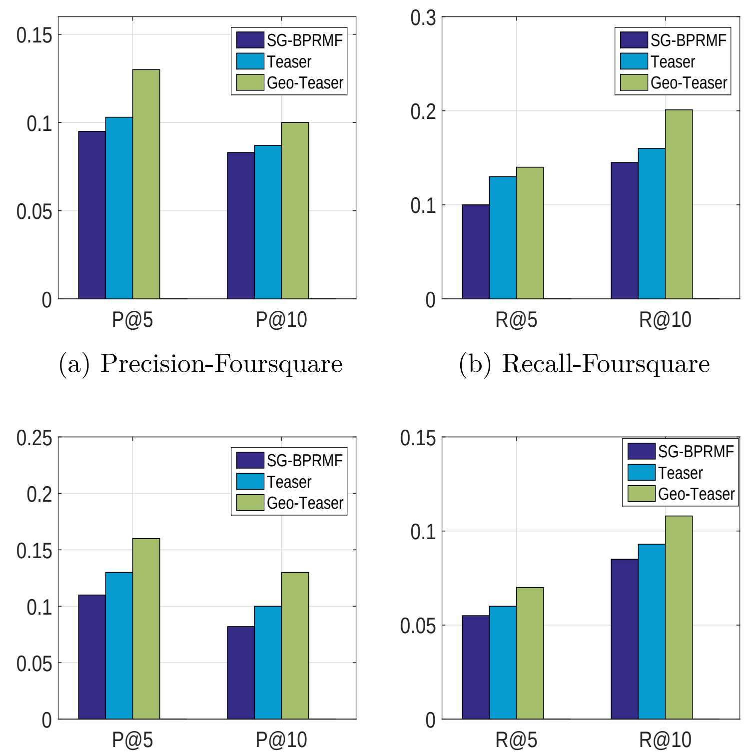
In the following, we demonstrate the experimental results on precision and recall, denoted as P@N and R@N, for the top N POI recommendation task. Since the model comparison results are consistent with different values of N, e.g., 1, 5, 10, and 20, we show representative results at 5 and 10 following [15, 16]. All models achieve the best performances at appropriate parameter settings.

Performance Comparison

Figure 5.5 illustrates the experimental results of different models. We discover that the proposed Geo-Teaser model achieves better performance than all the baselines. Compared with Rank-GeoFM that is a state-of-the-art model incorporating the geographical influence and temporal influence, Geo-Teaser achieves improvements at least 28% on both datasets for all metrics. This verifies the effectiveness of our sequential modeling and as well as the validity of means for incorporating various temporal characteristics and geographical influence. SG-CWARP is the best baseline competitor, which verifies the advantage of modeling the sequential pattern through Skip-Gram model over Markov chain model, namely the LORE model. Our Geo-Teaser model outperforms the SG-CWARP at least 20% on both datasets for all metrics, which verifies our strategy of incorporating various temporal characteristics and geographical influence to improve POI recommendation. In addition, we observe that models perform better on Gowalla than Foursquare for *precision*, but worse for *recall*. The reason lies in that each user's test data size in Gowalla is bigger than Foursquare. As shown in Table 5.1, the average check-ins for each user in Gowalla is about two times of Foursquare. According to the metrics in Eq. (5.15) and Eq. (5.16), the result is reasonable.

Model Discussion

In this section, we explore how each component, i.e., the various temporal characteristics and geographical influence, affects the model performance. The Geo-Teaser model improves the SG-CWARP in two aspects, capturing the various temporal



(a) Precision-Foursquare (b) Recall-Foursquare

(c) Precision-Gowalla (d) Recall-Gowalla

Figure 5.6: Demonstration of model component functions

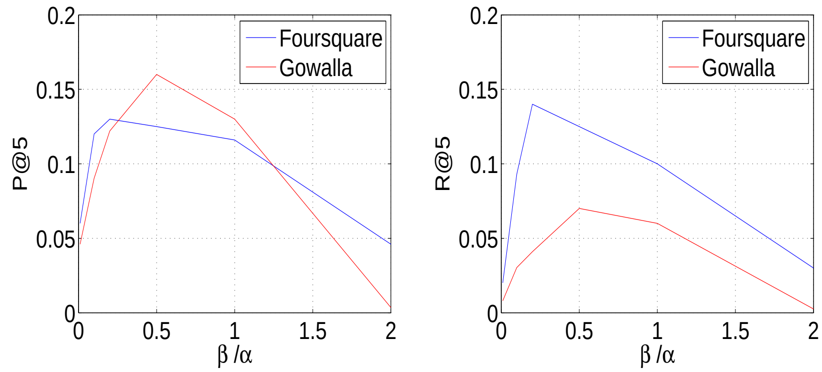
characteristics and geographical influence. Ignoring the various temporal characteristics and geographical influence, we propose the **SG-BPRMF** model as the basic version of our proposed Geo-Teaser model. The SG-BPRMF uses the Skip-Gram model to model the sequence and BPRMF to capture the user preference, which is equivalent to SG-CWARP. Furthermore, we incorporate the various temporal characteristics into SG-BPRMF and propose the **Teaser** model. In the following, we compare the SG-BPRMF, Teaser, and Geo-Teaser to show how the various temporal characteristics and geographical influence affect the model.

Figure 5.6 shows the model performances. We observe that

Teaser model improves SG-BPRMF at least about 10% on both datasets for all metrics, which indicates that incorporating the various temporal characteristics improves the model performance. Moreover, the Geo-Teaser model improves the Teaser model at least about 15% on both datasets. It means our strategy of incorporating geographical influence by discriminating the unvisited POIs is valid.

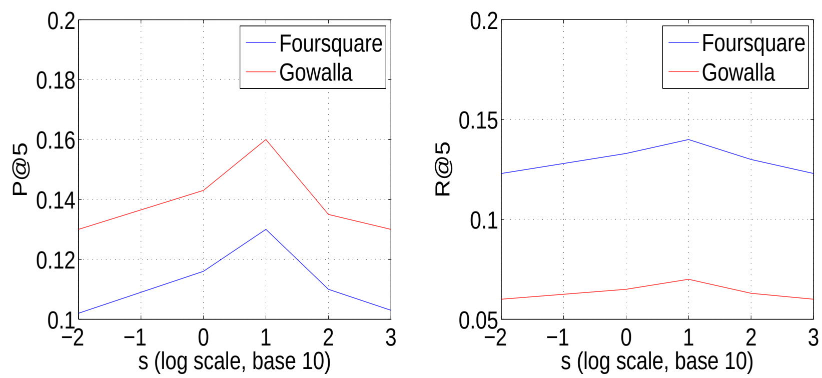
Parameter Effect

In this section, we show how the three important hyperparameters, α, β, and s affect the model performance. α and β balance the sequential influence and the user preference. s shows the sensitivity of our geographical model.
We tune α and β to see how to trade-off the sequential modeling and user preference learning, shown in Figure 5.7. Both α and β appear together with the learning rate η in the parameter update procedures. It is not necessary to separately tune the three parameters. We are able to absorb the learning rate η into α and β. In other words, we set $\alpha \leftarrow \alpha \cdot \eta, \beta \leftarrow \beta \cdot \eta$. We avoid to tune the learning rate η, but turn to control the update step size through tuning α and β. Hence α and β should be small enough to guarantee convergence. Assuming the same value for α and β, we tune α to change the learning rate. The model gets the best performance when $\alpha = 0.05$. Then we set $\alpha = 0.05$, and change β to see how the model performance varies with $\frac{\beta}{\alpha}$. Geo-Teaser attains the best performance if $\frac{\beta}{\alpha} \in [0.25, 0.5]$.
In the Geo-Teaser model, we classify the unvisited POIs as neighboring POIs and non-neighboring POIs to constitute a new preference set according to a threshold distance s. Here we choose different values of s to see how this parameter affects the



(a) P@5 on Geo-Teaser (b) R@5 on Geo-Teaser

Figure 5.7: Parameter effect on α and β



(a) P@5 (b) R@5

Figure 5.8: Parameter effect on distance threshold s

model performance, as shown in Figure 5.8. Here s is calculated in the kilometer. We observe that the Geo-Teaser model achieves the best performance at $s = 10$.

5.6 Conclusion

We propose the Geo-Teaser model for POI recommendation in this chapter. In particular, we propose the temporal POI embedding model to capture the check-ins' sequential contexts and the various temporal characteristics on different days. Moreover, we propose the geographically hierarchical pairwise ranking model to improve the recommendation performance through incorporating geographical influence. Finally, we propose a unified framework combining the two parts to recommend POIs. Experimental results on two datasets, Foursquare and Gowalla, show that our model outperforms state-of-the-art models. The proposed Geo-Teaser model improves at least 20% on both datasets for all metrics compared with SG-CWARP model.

□ **End of chapter.**

Chapter 6

STELLAR: Spatial-Temporal Latent Ranking Model for Successive POI Recommendation

Successive POI recommendation in LBSNs becomes a significant task since it helps users to navigate a large number of candidate POIs and provide the best POI recommendations based on users' most recent check-in knowledge. However, all existing methods for successive POI recommendation only focus on modeling the correlation between POIs based on users' check-in sequences, but ignore an important fact that successive POI recommendation is a time-subtle recommendation task. In fact, even with the same previous check-in information, users would prefer different successive POIs at different time. To capture the impact of time on successive POI recommendation, in this chapter, we propose a **s**patial-**te**mpora**l la**tent **r**anking (STELLAR) method to explicitly model the interactions among user, POI, and time. In particular, the proposed STELLAR model is built upon a ranking-based pairwise tensor factorization framework with a fine-grained modeling of user-POI, POI-time, and POI-POI interactions for successive POI recommendation. In addition, we design a novel three-slice indexing scheme to represent the timestamps, which captures the user check-ins' specific characteristics: preference variance and periodicity. Moreover, we propose a new interval-aware weight utility function to differentiate successive check-ins' correlations, which breaks the time interval constraint in prior work. Evaluations on two real-world datasets demonstrate that the STELLAR model outperforms state-of-the-art successive POI recommendation model about 20% in Precision@5 and Recall@5.

6.1 Introduction

LBSNs such as Foursquare, Gowalla, Facebook Place, and GeoLife, become increasingly popular and provide users a new way to share their locations and experience about POIs via check-in behaviors. To help users navigate a huge number of POIs and suggest the most suitable POIs to meet their personal preferences, POI recommendation methods are developed and play an important role in LBSN services. POI recommendation learns users' preferences based on user check-in records and then predicts users' preferred POIs for recommendation. To this end, a bunch of methods has been proposed for POI recommendation recently [106, 107, 7, 15, 44].

Successive POI recommendation, as a natural extension of general POI recommendation, is proposed and has attracted great research interest recently. Different from general POI recommendation that focuses only on estimating users' preferences on POIs, successive POI recommendation provides satisfied recommendations promptly based on users' most recent checked-in location, which requires not only the preference modeling from users but also the accurate correlation analysis between POIs. Cheng et al. [9] first propose the problem of successive POI recommendation and utilize a personalized Markov chain and region localization to solve the problem. In addition, Feng et al. [14] propose a personalized metric embedding method to model the check-in sequences. However, all previous methods ignore to investigate the impact of time on successive POI recommendation.

Successive POI recommendation is a time-subtle recommendation task since at different time users would prefer different successive POIs. It is easy to imagine that a user may go to a restaurant after leaving from office at noon, while the user may be more likely to go to a gym when the user leaves office at night. However, previous successive POI recommendation methods only highlight the modeling of correlations between POIs within users' check-in sequences, but neglect to model such a time-sensitive property.

In this chapter, we try to understand the underlying mechanism of how time influences successive POI recommendation performance. To motivate this work, we first conduct an empirical analysis on two real-world LBSN datasets to verify that time is an important factor to affect users' successive POI check-in behaviors. Based on the analysis, we propose the STELLAR model to recommend a user most possible successive POIs based on the most recent check-in and the querying timestamp. The proposed STELLAR model is built upon a ranking-based pairwise tensor factorization framework with a fine-grained modeling of user-POI, POI-time, and POI-POI interactions for successive POI

recommendation. To overcome the weaknesses of prior latent ranking models [6, 91, 75] that suffer from coupled interaction on POI feature, we represent each POI by three different latent feature vectors and model the three kinds of interactions separately. Moreover, the proposed STELLAR method contains two specific characteristics making it more suitable for successive POI recommendation: 1) we design a three-slice time indexing scheme to capture the temporal features of check-in behavior–preference variance and periodicity; 2) we introduce an interval-aware weight utility function to differentiate the correlations of successive check-ins, which breaks the time interval constraint in prior work [7].
The contributions of this chapter are summarized as follows:

- We propose a time-aware successive POI recommendation method–the STELLAR model, by considering the time information. In this model, we employ a new POI latent feature representation means to resolve the problem of coupled interaction. Experimental results demonstrate our STELLAR model outperforms state-of-the-art successive POI recommendation method.

- We design a three-slice time indexing scheme to represent the timestamps, which captures the user check-ins specific characteristics: preference variance and periodicity. Experimental results show that our model better captures the temporal effect than state-of-the-art temporal models for POI recommendation.

- We introduce a new interval-aware weight utility function to differentiate successive check-ins' correlations, which improves the successive POI recommendation accuracy.

6.2 Related Work

In this section, we first review the literature of latent ranking model. Then, we show the progress of POI recommendation. Finally, we present the connection of our proposed STELLAR model and the prior work.

Latent ranking model. Latent ranking model is a popular solution for recommendation tasks and ranking tasks [67, 99]. In a recommendation task, latent ranking model represents user and item feature into latent vectors, and find their relations in latent subspace. In particular, Singular Value Decomposition (SVD) [67] and Non-negative Matrix Factorization (NMF) [37] are two standard methods that exploit the latent ranking model for collaborative filtering task. Recently, [100] proposes the latent collaborative retrieval (LCR) model, which combines the retrieval and recommendation task, leading the direction of recommendations sensitive to some query condition.

POI recommendation. POI recommendation is an important task in LBSNs. Ye et al. firstly discuss how to use memory-based methods to recommend POIs [106, 107]. In order to improve the memory-based models, advanced techniques are then leveraged to capture more information, including social and geographical influence [94, 116, 117], temporal effect [113, 118], and sequential check-ins' influence [119, 117]. On the other hand, model-based methods are proposed for the seek of scalability, most of which base on the latent ranking techniques. [7] proposes a multi-center Gaussian model to capture user geographical influence and combines it with matrix factorization (MF) model [34] to recommend POIs. [15] proposes an MF-based model which captures the temporal effect to improve performance. [103], [29], and [16] leverage user comments to improve the POI recommendation system. [44] and [54] improve POI recommendation by incorporating geographical information in a weighted regularized matrix factorization model. Instead of estimating the user preference score on POIs, [9] and [41] establish ranking models to learn the recommender system. Other techniques for POI recommendation include generative graphical models, metric learning techniques, and graph-based method. Readers may refer the papers and references therein [14, 114, 45, 35, 111].

Connection to prior work. We focus on successive POI recommendation, which recommends POIs on the basis of a user's most recent check-in. [9] utilizes the latent ranking model to solve the problem, while [14] employs the metric learning. Our work is most related to [9]. However, prior work does not consider the time effect on successive POI recommendation, which motivates us to propose the STELLAR model. Moreover, we propose a three-slice time indexing scheme to represent the timestamps and introduce an interval-aware weight utility function to differentiate the correlations of successive check-ins.

6.3 Data Description and Successive Check-in Analysis

Before we introduce the proposed method, in this section, we first introduce two real-world LBSN datasets and then conduct some empirical analysis on them to explore the spatial and temporal properties of users' successive check-in behaviors.

6.3.1 Data Description

We use two check-in datasets crawled from real-world LBSNs: one is Foursquare data provided in [18] and the other is Gowalla data [122]. Both contain users' check-in history from January 1, 2011 to July 31, 2011. We filter the POIs checked-in by less than five users and then choose users who check-in more than 10 times as our samples. After the preprocessing, the datasets contain the statistical properties as shown in Table 6.1.

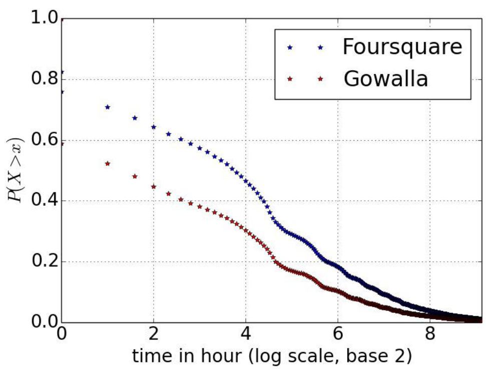
Table 6.1: Statistics of datasets

	Foursquare	**Gowalla**
#users	10,034	3,240
#POIs	16,561	33,578
#check-ins	865,647	556,453
Avg. #check-ins each user	86.3	171.7
Avg. #POIs each user	24.6	95.4
Avg. #users each POI	14.9	9.2
Density	0.0015	0.0028

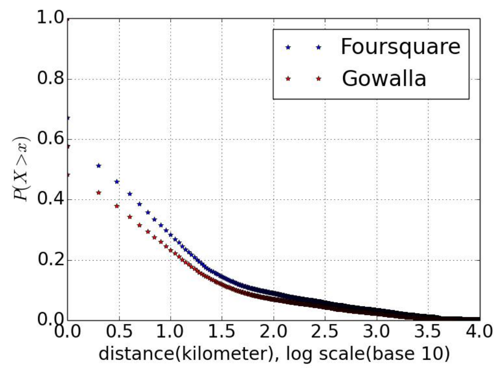
6.3.2 Successive Check-in Analysis

Now we conduct some empirical analysis to demonstrate the spatial and temporal properties of users' successive check-in behaviors.

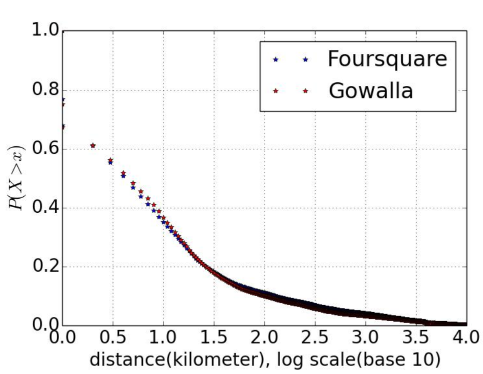
Spatial and temporal analysis. Successive check-ins demonstrate significant spatial and temporal property, shown in Figure 6.1. Figure 6.1(a) and Figure 6.1(b) show the complementary cumulative distribution function (CCDF) of intervals and distances in successive check-ins. We verify the observation in [9] that many successive check-ins are highly correlated especially in



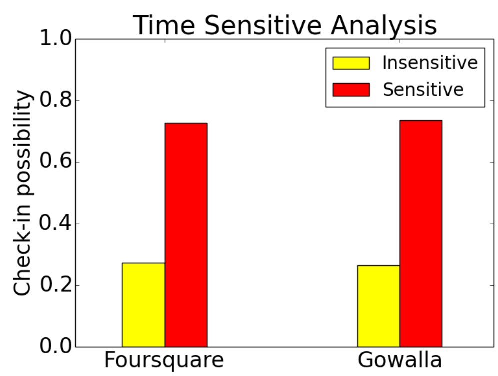
(a) CCDF of intervals in successive check-ins



(b) CCDF of distances in successive check-ins



(c) CCDF of distances in successive check-ins beyond 4 hours



(d) Time sensitive analysis of successive POI check-ins

Figure 6.1: Successive check-ins' spatial-temporal property

spatial relation: over 40% and 60% successive check-in behaviors happen in less than 4 hours since last check-in in Foursquare and Gowalla respectively; about 90% successive check-ins happen in less than 32 kilometers (half an hour driving distance) in Foursquare and Gowalla. Further, we check the CCDF of distances in successive check-ins that happen beyond 4 hours, shown in Figure 6.1(c). We observe although being weaker the spatial correlations still exist: about 80% successive checked-in POIs happen in less than 32 kilometers. It is not hard to explain the phenomenon: a user always acts around his/her home or office, so the successive check-in, even independent with the last check-in, still possibly happens in the same activity area. Hence, successive checked-in POIs are spatially correlated, while successive check-ins in shorter interval contain stronger correlation.

Time sensitive analysis. Besides the spatial and temporal contiguity, we observe that users' successive check-ins are time-sensitive behaviors. We count in all users and calculate (1) the average probability of a previous check-in leading to the same successive POI at different time (time-insensitive) and (2) the average probability of a check-in followed by different successive POIs at different timestamps (time-sensitive). Figure 6.1(d) shows the analytical results. We can obviously find that with different time, given the same previous POI check-in, users' successive POI check-ins would be different. This observation triggers us to incorporate time impact into successive POI recommendation.

6.4 STELLAR Model

In this section, we will detail the STELLAR model for successive POI recommendation. We first demonstrate how to index timestamps in our model. Then we introduce the formulation our STELLAR model. Finally, we demonstrate how to make the model inference and learn the system.

6.4.1 Time Indexing Scheme

To capture the check-in behavior's specific temporal characteristics, we design a novel time indexing scheme to smoothly encode a standard timestamp to a particular time id. The check-in behavior' temporal characteristics contain two aspects: (1) Periodicity [11, 113]. For example, users always visit restaurants at noon and bars at night; users check-in POIs around the office in weekdays but visit malls for shopping on weekends. (2) Preference variance [15]. Users' check-in preferences change with time. In addition, the preference variance exists in three scales: hours of a day, different days of a week, and different months of a year, which is observed in [15] but not modeled. Our proposed scheme captures the two properties in three scales as follows. First, a timestamp is divided into three slices in terms of month, weekday type, and hour slot. Next, we split a week into weekday and weekend and a day into the following four sessions: the morning session from 6:00 a.m. to 10:59 a.m., the afternoon and night session from 0:00 a.m. to 2:59 a.m. and 3:00 p.m. to 11:59 p.m., two transitive sessions that range from 3:00 a.m. to 5:59 a.m. and 11:00 a.m. to 2:59 p.m.. Further, we use 4 bits to represent the month information, 1 bit to denote weekday or weekend, and 2 bits to show the hour session. Finally, we convert the binary code into a unique decimal digit as the time ID, where the ID is in the range of 0 to 95. Figure 6.2 demonstrates the procedure of encoding an exemplary timestamp, “2011-04-05 18:10:23”.

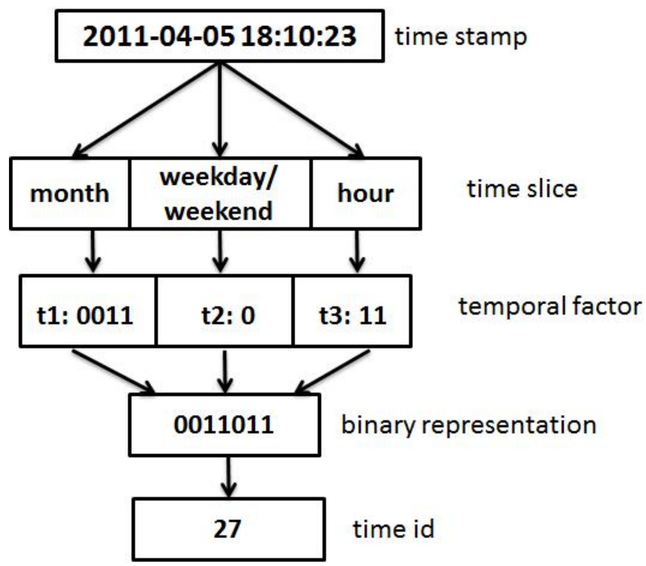


Figure 6.2: Time encoding demonstration

6.4.2 Model Formulation

The STELLAR system aims to provide time-aware successive POI recommendations. The task needs to learn a score function for a given user u to a candidate POI l^c at the timestamp t given his/her last check-in as a query POI l^q, which is defined as follows:

$$f(u, l^q, t, l^c), \tag{6.1}$$

where $f : \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{L} \to \mathbb{R}$ maps a four-tuple tensor to real values. $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{T}$ denote the set of users, the set of POIs, and the set of smoothed time ids, respectively. The score value

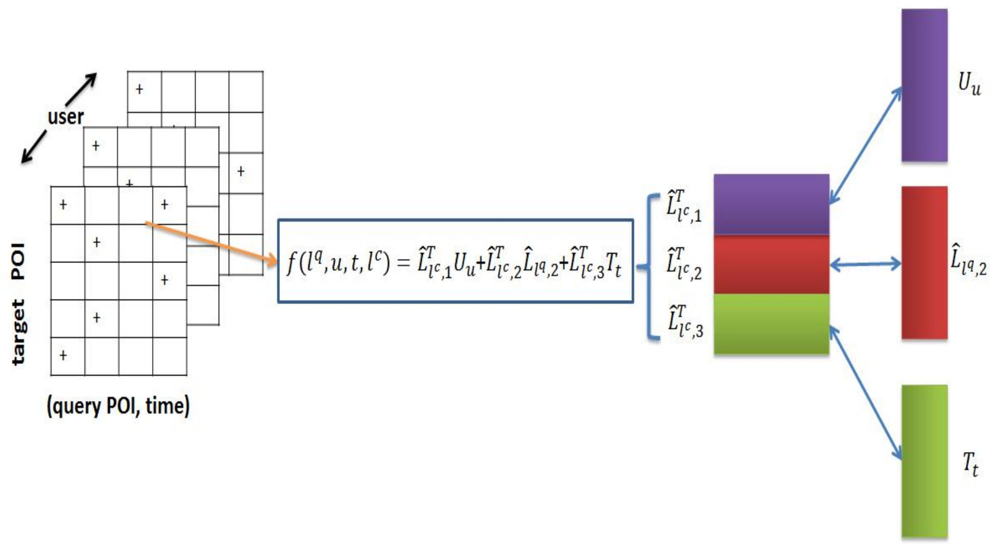


Figure 6.3: STELLAR model formulation demonstration

represents the "successive check-in possibility" of a user to a candidate POI at the timestamp given the query POI.

We establish a latent ranking framework to learn the score function, which employs pairwise tensor interactions to represent the following three key factors affecting users' check-in behavior: (1) the preference of a user u to a candidate POI l^c, (2) the temporal effect of time t on a candidate POI l^c, and (3) correlation of the last checked-in POI l^q and a candidate POI l^c. Correspondingly, the score value of $f(u, l^q, t, l^c)$ is determined by user-POI interaction, time-POI interaction, and POI-POI interaction together. In this case, a single vector representation for each POI is not semantically enough to capture the three different kinds of interactions. Therefore, we define a $3 \times d$ matrix to represent POI latent feature, where for each POI, there are three latent vectors used to describe the POI-user interaction, POI-time interaction and POI-POI interaction, respectively. As shown in Figure 6.3, we formulate the function $f(u, l^q, t, l^c)$ as

$$f(u, l^q, t, l^c) = \hat{L}^T_{l^c,1} U_u + \hat{L}^T_{l^c,2} \hat{L}_{l^q,2} + \hat{L}^T_{l^c,3} T_t, \tag{6.2}$$

where $U_u, T_t \in R^d$ are latent vectors of user u and time t, $\hat{L}_{l^c,1}, \hat{L}_{l^c,2}, \hat{L}_{l^c,3} \in R^d$ are candidate POI l^c's three d-dimension vectors which correspondingly interact with users, other POIs and time labels, and $\hat{L}_{l^q,2}$ is query POI l^q's latent vector interacting to the candidate POI. To ensure the interactions are positive, all latent vectors are non-negative. Further we denote $U \in R^{d\times|\mathcal{U}}|$ as the user latent matrix and $T \in R^{d\times|\mathcal{T}|}$ as the time latent matrix. In addition, we use $\hat{L}$, a $3 \times d \times |\mathcal{L}|$ tensor, to denote the POI latent factor.

From the observations in Figure 6.1, we find that successive POIs in a shorter interval contain a stronger correlation. To depict this observation, we introduce a weight utility function to differentiate the strong and weak correlations. The weight score value is in the range of [0,1], and the function is non-increasing with the duration of two successive check-ins. When two successive check-ins happen within a threshold interval, we assume they are highly correlated. Otherwise the correlation decreases with the increase of the time interval. In formal, we define the weight utility function as follows:

$$w = \begin{cases} 0.5 + \frac{2}{\Delta T} & \Delta T \geq s \\ 1 & otherwise \end{cases}, \tag{6.3}$$

where ΔT is the interval of successive check-ins, in unit of hour; and s is the threshold of differentiating the correlations. In our experiments, s is set as 4 to get best performance. The check-in time of query POI l^q and current time t determine the interval ΔT. So we are able to refine the score function as

$$f(u, l^q, t, l^c, w) = \hat{L}^T_{l^c,1} U_u + w \cdot \hat{L}^T_{l^c,2} \hat{L}_{l^q,2} + \hat{L}^T_{l^c,3} T_t, \tag{6.4}$$

where w is the weight value to measure the POI-POI interaction. The STELLAR model is proposed to handle the following two challenging issues: (1) **Disastrous sparsity**. Prior methods learn a model from a tensor with only three tuples. In our formulation, we focus on tuples of four elements, (user, POI, time, POI), which increase the sparsity of the tensor significantly. (2) **Coupled interaction**. The tuples in previously proposed tensor related methods are independent. For example, in [72, 75], the tuple includes user, item, and tag, which are independent. This is easier for updating the models. However, in our constructed tensor, the tuple includes two POIs coupled in the updating. This makes the previous tensor decomposition methods[91, 6, 75] unsatisfactory. Our method represents the POI feature via a matrix and then models the three kinds of interactions separately. Further, we simplify the tensor completion problem as a combination of three low rank matrix factorization problems, which mitigates the sparsity trouble.

6.4.3 Model Inference and Learning

We make the model inference via learning the ranking order of successive check-in possibilities. Because we care more about the ranking order of the candidate POIs rather than the real values of check-in possibilities when recommending successive POIs for users. We follow the optimization criteria used in [73] and propose a pairwise ranking based objective function for the proposed STELLAR model.

We demonstrate the inference procedure following [73]. First, we suppose that the scores of $f(u, l^q, t, l^c)$ at checked-in POIs are higher than the unchecked-in counterparts. Then we define the order $l^c_p >_{u,l^q,t} l^c_n$, which means at time t, given query POI

l^q, user u visits POI l_p^c but not l_n^c. Further we suffice to extract the set of all pairwise preference constraints

$$D_S := \{(u, l^q, t, l_p^c, l_n^c) | l_p^c >_{u,l^q,t} l_n^c\}. \tag{6.5}$$

Suppose the tuples in D_S are independent of each other, then to learn the parameters in the score function is to minimize the negative log likelihood of all the pair orders. Further, we add a Frobenius norm term to regularize the parameters to avoid the risk of overfitting. Then the objective function is

$$\begin{aligned} \mathcal{O} := \arg\min_{\Theta} \sum_{(u,l^q,t,l_p^c,l_n^c)\in D_S} & - ln(\sigma(f(u, l^q, t, l_p^c) - \\ & f(u, l^q, t, l_n^c))) + \lambda ||\Theta||_F^2, \end{aligned} \tag{6.6}$$

where σ is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$, λ is the regularization parameter, and Θ denotes the parameter set, including U, T and $\hat{L}$.

We leverage the stochastic gradient decent (SGD) algorithm to learn the objective function for efficacy. Denote $\delta = 1 - \sigma(y_{u,l^q,t,l_p^c,l_n^c})$, then we get the derivative of each parameter $\theta \in \Theta$ for a tuple $(u, l^q, t, l_p^c, l_n^c)$ as

$$\frac{\partial \mathcal{O}}{\partial \theta} = \begin{cases} -\delta \cdot (\hat{L}_{l_p^c,1} - \hat{L}_{l_n^c,1}) + \lambda \cdot U_u & \theta = U_u \\ -\delta \cdot (\hat{L}_{l_p^c,3} - \hat{L}_{l_n^c,3}) + \lambda \cdot T_t & \theta = T_t \\ -\delta \cdot w \cdot (\hat{L}_{l_p^c,2} - \hat{L}_{l_n^c,2}) + \lambda \cdot \hat{L}_{l^q,2} & \theta = \hat{L}_{l^q,2} \\ -\delta \cdot U_u + \lambda \cdot \hat{L}_{l_p^c,1} & \theta = \hat{L}_{l_p^c,1} \\ -\delta \cdot w \cdot \hat{L}_{l^q,2} + \lambda \cdot \hat{L}_{l_p^c,2} & \theta = \hat{L}_{l_p^c,2} \\ -\delta \cdot T_t + \lambda \cdot \hat{L}_{l_p^c,3} & \theta = \hat{L}_{l_p^c,3} \\ \delta \cdot U_u + \lambda \cdot \hat{L}_{l_n^c,1} & \theta = \hat{L}_{l_n^c,1} \\ \delta \cdot w \cdot \hat{L}_{l^q,2} + \lambda \cdot \hat{L}_{l_n^c,2} & \theta = \hat{L}_{l_n^c,2} \\ \delta \cdot T_t + \lambda \cdot \hat{L}_{l_n^c,3} & \theta = \hat{L}_{l_n^c,3}. \end{cases} \tag{6.7}$$

To ensure the non-negativity, we project the learned parameter to non-negative value. We define the projected operator $P(\cdot) : R^d \to R^d$ as $P[x_i] = \max(0, x_i), i = 1, \ldots, d$. For each sampled tuple $(u, l^q, t, l^c_p, l^c_n) \in D_S$, we update each parameter $\theta \in \Theta$ through the derivative,

$$\theta \leftarrow P(\theta - \gamma \frac{\partial \mathcal{O}}{\partial \theta}); \tag{6.8}$$

where γ is the learning rate. To train the model, we draw the tuple from D_S via the bootstrap sampling rule, following [73]. Algorithm 5 gives the detailed procedure to learn the STELLAR model. The convergent condition is satisfied when the negative log likelihood value for a fixed sampled tuples does not decrease.

Complexity. Calculating the preference score of a tuple (u, l^q, t, l^c) costs $O(d)$, where d is the latent vector dimension. The updating procedure for each parameter is also in $O(d)$. Hence training an example (u, l^q, t, l^c) is in $O(k \cdot d)$, where k is the number of sampled unchecked POIs. Therefore, the runtime of training the model is in $O(N \cdot k \cdot d)$, where N is the number of training examples.

ALGORITHM 5: STELLAR model learning algorithm

Input: Training tuples $\{(u_i, l^q_i, t_i, l^c_i)\}_{i=1,\ldots,N}$

Output: $U, T, \hat{L}$

1: Initialize $U, T, \hat{L}$
2: **repeat**
3: Draw (u, l^q, t, l^c_p) uniformly from training tuples
4: For $s = 1, \cdots, k$, where k is #sampled unchecked POIs
5: Draw $(u, l^q, t, l^c_p, l^c_n)$ uniformly
6: Update parameters according to Eq. (6.8)
7: **until** convergence
8: **return** $U, T, \hat{L}$

Table 6.2: Performance comparison

		BPRMF	WRMF	LRT	FPMC−LR	TLAR	SLAR	STELLAR
Gowalla	P@5	0.025	0.031	0.033	0.048	0.053	0.050	**0.059**
	R@5	0.020	0.022	0.030	0.167	0.204	0.197	**0.226**
Foursquare	P@5	0.031	0.033	0.061	0.109	0.119	0.114	**0.129**
	R@5	0.027	0.028	0.053	0.347	0.373	0.368	**0.425**

6.5 Experiment

We conduct experiments to answer the following questions: 1) how our model performs comparing with state-of-the-art models? 2) whether our time indexing scheme works well? 3) how the parameters affect the model performance?

6.5.1 Experimental Setting

We evaluate our model on two datasets with statistics shown in Table 6.1. The system recommends a user a list of POIs, given his/her last checked-in POI and timestamp as the query. It is equivalent to solve the collaborative retrieval task [100], treating (query POI, time id, weight) as the query for each user. Following setting in [100], we extract tuples of (user, query POI, time id, weight, POI) from all successive check-ins. Here we get time id from the check-in timestamp via encoding procedure. And the weight value is calculated according to the interval between two successive check-ins through the utility function in Eq. (6.3). In order to make our model effective for future check-ins, we split the tuples into two parts, 80% and 20% according to time sequential order. So we take the first group of tuples for training and the second group for test. Finally, we measure different models through Precision@5 and Recall@5, which are general metrics for POI recommendation problem used in prior work [9, 15, 107].

6.5.2 Comparison Methods

Our Methods. We propose three methods: **TLR**, **SLR**, and **STELLAR**. TLR and SLR methods are special cases of STELLAR, which correspondingly only ignore the POI-POI interaction and time-POI interaction.

Baselines. We compare our proposed model with state-of-the-art latent ranking models and POI recommendation methods. Prior work [44, 54] indicates that treating the check-ins as implicit feedback is better to recommend POIs. Hence we introduce two comparative latent ranking methods that model the check-ins as implicit feedback: **WRMF** [30, 66] and **BPRMF** [73]. In addition, we introduce two state-of-the-art POI recommendation methods: **LRT** [15] and **FPMC-LR** [9]. LRT is state-of-the-art model that incorporates temporal information in a latent ranking model to improve POI recommendation. FPMC-LR is the state-of-the-art successive POI recommendation model.

6.5.3 Experimental Results

In the following, we demonstrate the performance comparison. We set latent dimension as 40, and train different models to get their best performances at appropriate parameters.

Baselines vs. Our Methods. Table 6.2 shows the experimental results on Foursquare and Gowalla data. We see that: 1) Our proposed model outperforms state-of-the-art latent ranking methods and POI recommendation models. Compared with state-of-the-art successive POI recommendation method, STELLAR model gains about 22.9% and 35.3% improvement for Gowalla, and 18.3% and 22.5% improvement for Foursquare on Precision@5 and Recall@5. We observe that all models perform much better on Foursquare dataset than Gowalla dataset, even though it is sparser. The reason lies in Foursquare data contain much less POIs. 2) Our proposed models and FPMC-LR perform much better than other models, especially at recall measure. The reason lies in that these models leverage more conditions for each query. Our models recommend a user POIs given a user's recent check-in, the specific timestamp, or both; and FPMC-LR recommends POIs given a user's recent checked-in POIs. On the contrary, other three models give general recommendations.

LRT vs. TLAR. The experimental results show that TLAR outperforms LRT model. Our model depicts the temporal effect with a latent feature, which gets rid of sparsity problem suffering in LRT model. Furthermore, since TLAR is a special case of STELLAR, it means that STELLAR model captures the temporal effect well from the timestamps.

FPMC-LR vs. SLAR. The experimental results show that SLAR outperforms FPMC-LR model. It means SLAR model improves the recommendation performance by differentiating the correlations of successive check-ins.

6.5.4 Discussion of Time Indexing Scheme

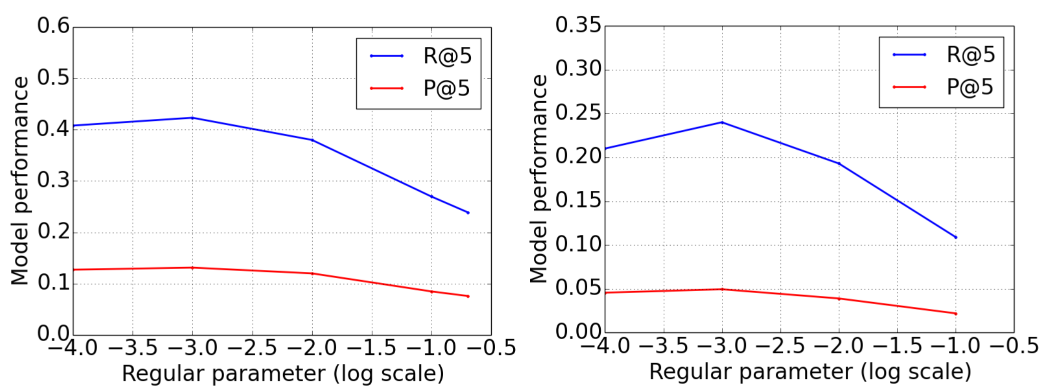
Our three-slice time indexing scheme effectively captures the temporal effect in three scales. In order to demonstrate its efficacy, we ignore one slice to index the time and then compare their results with our model, shown in Table 6.3. 'M', 'W', and 'D' represent month, week, and day slice respectively. Our model demonstrates the best performance.

Table 6.3: Comparison of different time schemes

		M+W	M+D	W+D	M+W+D
Gowalla	P@5	0.051	0.053	0.054	**0.059**
	R@5	0.207	0.208	0.219	**0.226**
Foursquare	P@5	0.118	0.120	0.121	**0.129**
	R@5	0.371	0.389	0.398	**0.425**

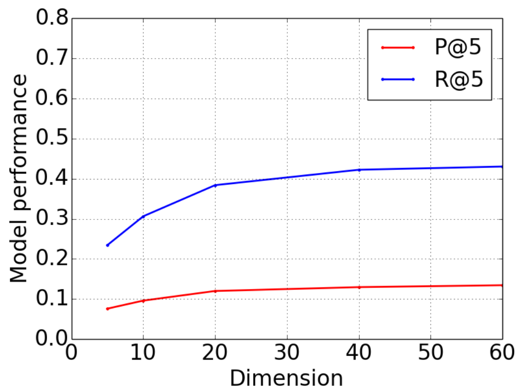
6.5.5 Parameter Effect

The regularization and latent dimension are important parameters to learn a latent ranking model. Figure 6.4 and Figure 6.5 demonstrate the effect of the parameters on model performance. For simplicity, we set the same value for all latent vectors' regularizations in the model. The model has best performance when $\lambda = 0.001$. The performance of Stellar steadily rises with the increase of latent vector dimension. For the trade-off of performance and computation cost, we suggest to set dimension $d = 40$.

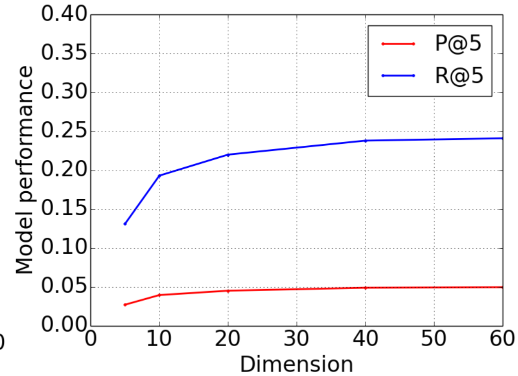


(a) Fousquare (b) Gowalla

Figure 6.4: The effect of regularization



(a) Foursquare



(b) Gowalla

Figure 6.5: The effect of latent dimension

6.6 Conclusion

In this chapter, we study the problem of successive POI recommendation. Compared with previous work, we show that successive POI recommendation is a time-subtle recommendation task. To capture the time impact, we first design a time indexing scheme to smoothly encode timestamps to particular time ids and then incorporate the time ids into our proposed STELLAR model. The STELLAR model is built upon a ranking-based pairwise interaction tensor factorization framework with a fine-grained modeling of the interactions among time, user, and POI. Experimental results on two datasets, Foursquare and Gowalla, show that the STELLAR model outperforms state-of-the-art models.

□ **End of chapter.**

Chapter 7

Conclusion and Future Work

In this chapter, we summarize the main contributions of this thesis and provide several interesting future directions.

7.1 Conclusion

POI recommendation is an important application in LBSNs. Due to its special geographical and temporal characteristics, POI recommendation is more challenging than traditional recommendation tasks. In order to understand the user check-in activity in LBSNs, we analyze the user mobility from geographical and temporal perspective respectively and show how to improve the POI recommendation through the geographical influence and temporal influence. Moreover, we propose two POI recommendation systems: Geo-Teaser and STELLAR.

In particular, in chapter 3, we understand the human mobility in LBSNs from the geographical perspective and attempt to model the geographical influence for POI recommendation. In particular, we propose two models— GMM and GA-GMM to capture geographical influence. More specifically, we exploit GMM to automatically learn users' activity centers; further, we utilize GA-GMM to improve GMM by eliminating outliers. Experimental results on a real-world LBSN dataset show that GMM beats several popular geographical capturing models regarding POI recommendation, while GA-GMM excludes the effect of outliers and enhances GMM.

In chapter 4, we study the human mobility in LBSNs from the temporal perspective. We summarize the temporal characteristics of user mobility in LBSNs in three aspects: periodicity, consecutiveness, and non-uniformness. Moreover, we observe that the temporal characteristics exist at different time scales, which cannot be modeled in prior work. To this end, we propose the ATTF model for POI recommendation to capture the three temporal features together, as well as at different time scales. Experiments on two real-world datasets show that the ATTF model achieves better performance than the state-of-the-art temporal models for POI recommendation.

In chapter 5, we propose the Geo-Teaser system for POI recommendation. In particular, inspired by the success of the word2vec framework to model the sequential contexts, we first propose a *temporal POI embedding* model to learn POI representations under some particular temporal state. The temporal POI embedding model captures the contextual check-in information in sequences and the various temporal characteristics on different days as well. Furthermore, we propose a new way to incorporate the geographical influence into the pairwise preference ranking method through discriminating the unvisited POIs according to geographical information. Then we develop a geographically hierarchical pairwise preference ranking model. Finally, we propose a unified framework to recommend POIs combining these two models. To verify the effectiveness of our proposed method, we conduct experiments on two real-life datasets. Experimental results show that the Geo-Teaser model outperforms state-of-the-art models.

In chapter 6, we propose the STELLAR model for time-aware successive POI recommendation. In particular, the proposed STELLAR model is built upon a ranking-based pairwise tensor factorization framework with a fine-grained modeling of user-POI, POI-time, and POI-POI interactions for successive POI recommendation. In addition, we design a novel three-slice indexing scheme to represent the timestamps, which captures the user check-ins' specific characteristics: preference variance and periodicity. Moreover, we propose a new interval-aware weight utility function to differentiate successive check-ins' correlations, which breaks the time interval constraint in prior work. Evaluations on two real-world datasets demonstrate that the STELLAR model outperforms state-of-the-art successive POI recommendation model about 20% in Precision@5 and Recall@5.

7.2 Future Work

A bunch of studies has been proposed for POI recommendation. Summarizing the existing work, we point out the trends and new directions in three possible aspects: ranking-based model, online recommendation, and deep learning based recommendation.

7.2.1 Ranking-based Model

Several ranking-based models [14, 41, 128] have been proposed for POI recommendation recently. Most of the previous methods attempt to estimate the user check-in probability over POIs [7, 15, 16]. However, for the POI recommendation task, we do not care about the predicted check-in possibility value but the preference order. Some work has proved that it is better for the recommendation task to learn the order rather than the real value [73, 38, 92, 99, 101]. BPR loss [73] and WARP loss [92, 99] are two popular pairwise loss criteria to learn the ranking order. Researchers in [8, 14, 128] leverage the BPR loss to learn a POI recommendation model, and Li et al. [41] use the WARP loss. Also, He et al. [26] propose a list-wise ranking model for POI recommendations. The existing work using ranking-based model has shown its advantage in model performance. Then, learning to rank, as an important technique for information retrieval [5, 50], may be used more for POI recommendation to improve performance in the future.

7.2.2 Online Recommendation

The online POI recommendation model has advantages over offline models in two aspects: cold-start problem and adaptability to the user behavior variance. Most of the previous work recommends POIs via the offline model. Hence, the previous work is apt to suffer the two problems: (1) cold-start problem, the proposed model performs not satisfying for new users or users who have only a few check-ins; (2) user behavior variance, the proposed model, may perform awfully if a user's behavior changes since it learns user behavior according to historical records. Researchers in [1, 111] utilize offline model and online recommendation to improve the recommendation results. However, there is no work using online model for POI recommendation. In fact, online recommendation models based on multi-armed bandits [4] have been proposed for movie recommendation and advertisement recommendation [70, 129]. In the future, online recommendation methods will be a new direction for POI recommendation.

7.2.3 Deep Learning Based Recommendation

Inspired by the success of deep learning, the neural network method has been used to model the check-in sequences. Liu et al. [49] employ RNN to find the sequential correlations. In addition, several studies [13, 52, 102, 127] leverage the embedding learning for POI recommendations. Liu et al. [52] model the check-in sequences through the word2vec framework [61] to capture the sequential contexts. Xie et al. [102] propose a graph-based framework for POI recommendations to systematically model the POI, user, and time relations in an embedding space and learn the representations through the word2vec framework. Moreover, we [127] propose a temporal POI embedding based on Skip-Gram model [61] and combine it with a geographically pairwise user preference ranking model to recommend POIs. In the future, more advanced techniques, such as LSTM [27], can be used for POI recommendation.

□ **End of chapter.**

Appendix A

Publications during Ph.D. Study

Journal and Book Chapter

1. **Shenglin Zhao**, Tong Zhao, Irwin King, and Michael R. Lyu. "C^2SEER: **C**ontext and **C**ontent Aware **Se**quential **E**mbedding **R**ank Model for Point-of-interest Recommendation". ACM TIST (Manuscript)

2. **Shenglin Zhao**, Irwin King, and Michael R. Lyu. "A Survey on Point-of-interest Recommendation in Location-based Social Networks". ACM TWeb (Under Review).

3. **Shenglin Zhao**, Michael R. Lyu, and Irwin King. "Aggregated Temporal Tensor Factorization Model for Point-of-interest Recommendation". Neural Processing Letters.

4. **Shenglin Zhao**. "Location-based Social Network Analysis". Encyclopedia of Social Network Analysis and Mining (Under Review)

Conference

5. **Shenglin Zhao**, Irwin King, and Michael R. Lyu. “Geo-Pairwise Ranking Matrix Factorization Model for Point-of-interest Recommendation”. ICONIP 2017 (Best paper nominee).

6. Sheng Zhang, **Shenglin Zhao**, Mingxuan Yuan, Jia Zeng, Jianguo Yao, Irwin King, and Michael R. Lyu. “Traffic Prediction Based Power Saving in Cellular Networks: A Machine Learning Method”. SIGSPATIAL 2017.

7. Jiajun Cheng, **Shenglin Zhao**, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. “Aspect-level Sentiment Classification with HEAT (Hierarchical Attention) Network”. CIKM 2017.

8. **Shenglin Zhao**, Michael R. Lyu, Irwin King, Jia Zeng, and Mingxuan Yuan. “Mining Business Opportunities from Location-based Social Networks”. SIGIR 2017.

9. **Shenglin Zhao**, Tong Zhao, Irwin King, and Michael R. Lyu. “Geo-Teaser: Geo-Temporal Sequential Embedding Rank for Point-of-interest Recommendation”. WWW 2017 (Cognitive Computing Track).

10. **Shenglin Zhao**, Michael R. Lyu, and Irwin King. “Aggregated Temporal Tensor Factorization Model for Point-of-interest Recommendation”. ICONIP 2016.

11. Qi Xie, **Shenglin Zhao**, Zibin Zheng, Jieming Zhu, and Michael R. Lyu. “Asymmetric Correlation Regularized Matrix Factorization for Web Service Recommendation”. ICWS 2016.

12. **Shenglin Zhao**, Tong Zhao, Haiqin Yang, Michael R. Lyu, and Irwin King. "STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation". AAAI 2016.

13. **Shenglin Zhao**, Haiqin Yang. "Scalable Point-of-interest Recommendation via Geo-embedding Pairwise Matrix Factorization". WSDM 2015 workshop on Scalable Data Analytics.

14. **Shenglin Zhao**, Irwin King, and Michael R. Lyu. "Capturing Geographical Influence in POI Recommendations". ICONIP 2013.

Note: The papers [1, 2, 3, 4, 8, 9, 11, 12, 13] are partially involved in this thesis.

□ **End of chapter.**

Bibliography

[1] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 199–208, 2012.

[2] P. Bhargava, T. Phan, J. Zhou, and J. Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 130–140. ACM, 2015.

[3] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[4] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[6] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, 1970.

[7] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 17–23. AAAI Press, 2012.

[8] C. Cheng, H. Yang, I. King, and M. R. Lyu. A unified point-of-interest recommendation framework in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):10, 2016.

[9] 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.

[10] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *Fifth International AAAI Conference on Weblogs and Social Media*, pages 81–88. AAAI, 2011.

[11] E. Cho, S. A. 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.

[12] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[13] S. Feng, G. Cong, B. An, and Y. M. Chee. Poi2vec: Geographical latent representation for predicting future visitors. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 102–108, 2017.

[14] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan. Personalized ranking metric embedding for next new poi recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2069–2075. AAAI Press, 2015.

[15] 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.

[16] H. Gao, J. Tang, X. Hu, and H. Liu. Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1721–1727. AAAI Press, 2015.

[17] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *Sixth International AAAI Conference on Weblogs and Social Media.* AAAI, 2012.

[18] H. Gao, J. Tang, and H. Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1582–1586. ACM, 2012.

[19] H. Gao, J. Tang, and H. Liu. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining & Knowledge Discovery*, 29(2):299–323, 2015.

[20] Y. Ge, H. Li, and H. Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. 2016.

[21] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[23] C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Advances in information retrieval*, pages 345–359. Springer, 2005.

[24] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

[25] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung. Inferring a personalized next point-of-interest recommendation

model with latent behavior patterns. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[26] L. L. He Jing, Li Xin. Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking. In *International joint conference on Artificial Intelligence (IJCAI)*. AAAI Press, 2017.

[27] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] T. Horozov, N. Narasimhan, and V. Vasudevan. Using location for personalized poi recommendations in mobile environments. In *International Symposium on Applications and the Internet (SAINT'06)*, pages 6–pp. IEEE, 2006.

[29] B. Hu and M. Ester. Social topic modeling for point-of-interest recommendation in location-based social networks. In *2014 IEEE International Conference on Data Mining*, pages 845–850. IEEE, 2014.

[30] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.

[31] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[32] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social net-

works. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[33] 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.

[34] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[35] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 375–384. ACM, 2013.

[36] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, volume 14, pages 1188–1196, 2014.

[37] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[38] J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer. Local collaborative ranking. In *Proceedings of the 23rd international conference on World wide web*, pages 85–96. ACM, 2014.

[39] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[40] H. Li, R. Hong, S. Zhu, and Y. Ge. Point-of-interest recommender systems: A separate-space perspective. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 231–240. IEEE, 2015.

[41] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 433–442. ACM, 2015.

[42] Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Proceedings of the 25th international joint conference on Artificial Intelligence*, pages 3650–3656, 2015.

[43] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui. Content-aware collaborative filtering for location recommendation based on human mobility data. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 261–270. IEEE, 2015.

[44] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 831–840. ACM, 2014.

[45] 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.

[46] B. Liu and H. Xiong. Point-of-Interest recommendation in location based social networks with topic and location awareness. In *Siam International Conference on Data Mining*, pages 396–404. SIAM, 2013.

[47] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1167–1179, 2015.

[48] P. Liu, X. Qiu, and X. Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 25th international joint conference on Artificial Intelligence*, 2015.

[49] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[50] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[51] X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM*

international conference on Information & Knowledge Management, pages 733–738. ACM, 2013.

[52] X. Liu, Y. Liu, and X. Li. Exploring the context of locations for personalized location recommendations. In *Proceedings of the 25th International joint Conference on Artificial Intelligence*, 2016.

[53] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun. Topical word embeddings. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[54] Y. Liu, W. Wei, A. Sun, and C. Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *ACM International Conference on Conference on Information and Knowledge Management*, pages 739–748, 2014.

[55] H. Ma, C. Liu, I. King, and M. R. Lyu. Probabilistic factor models for web site recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 265–274. ACM, 2011.

[56] 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.

[57] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[58] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.

[59] M. Melanie. An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, Fifth printing*, 3, 1999.

[60] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[61] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[62] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[63] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[64] N. Neykov, P. Filzmoser, R. Dimova, and P. Neytchev. Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics & Data Analysis*, 52(1):299–308, 2007.

[65] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *2012 IEEE 12th International Conference on Data Mining*, pages 1038–1043. IEEE, 2012.

[66] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 502–511. IEEE Computer Society, 2008.

[67] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[68] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[69] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[70] L. Qin, S. Chen, and X. Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 461–469. SIAM, 2014.

[71] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

[72] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on*

Knowledge discovery and data mining, pages 727–736. ACM, 2009.

[73] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[74] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.

[75] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

[76] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong. On the levy-walk nature of human mobility. *IEEE/ACM transactions on networking (TON)*, 19(3):630–643, 2011.

[77] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[78] F. Ricci and B. Shapira. *Recommender systems handbook*. Springer, 2011.

[79] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning representations by back-propagating errors.* MIT Press, 1986.

[80] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems 20*, 20:1257–1264, 2008.

[81] J. Sang, T. Mei, and C. Xu. Activity sensor: Check-in usage mining for local recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):41, 2015.

[82] M. Sattari, M. Manguoglu, I. H. Toroslu, P. Symeonidis, P. Senkul, and Y. Manolopoulos. Geo-activity recommendations by using improved feature combination. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 996–1003. ACM, 2012.

[83] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: geo-social metrics for online social networks. In *Wonference on Online Social Networks*, pages 8–8, 2010.

[84] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, Usa, August*, pages 1046–1054, 2011.

[85] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[86] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[87] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, 2015.

[88] D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In *IJCAI*, pages 1340–1346, 2015.

[89] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.

[90] N. D. Thang, C. Lihui, and C. C. Keong. An outlier-aware data clustering algorithm in mixture models. In *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, pages 1–5. IEEE, 2009.

[91] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[92] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th annual international conference on machine learning*, pages 1057–1064. ACM, 2009.

[93] B. Wang, C. M. Wong, F. Wan, P. U. Mak, P. I. Mak, and M. I. Vai. Gaussian mixture model based on genetic algorithm for brain-computer interface. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 9, pages 4079–4083. IEEE, 2010.

[94] 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 374–383. ACM, 2013.

[95] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 391–400. International World Wide Web Conferences Steering Committee, 2017.

[96] X. Wang, Y.-L. Zhao, L. Nie, Y. Gao, W. Nie, Z.-J. Zha, and T.-S. Chua. Semantic-based location recommendation with multimodal venue semantics. *IEEE Transactions on Multimedia*, 17(3):409–419, 2015.

[97] Y. Wang, N. J. Yuan, D. Lian, L. Xu, X. Xie, E. Chen, and Y. Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1275–1284, 2015.

[98] Z.-S. Wang, J.-F. Juang, and W.-G. Teng. Predicting poi visits with a heterogeneous information network. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 388–395. IEEE, 2015.

[99] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, 2010.

[100] J. Weston, C. Wang, R. Weiss, and A. Berenzweig. Latent collaborative retrieval. *ICML*, 2012.

[101] J. Weston, C. Wang, R. Weiss, and A. Berenzweig. Latent collaborative retrieval. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 9–16, 2012.

[102] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 15–24. ACM, 2016.

[103] D. Yang, D. Zhang, Z. Yu, and Z. Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.

[104] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015.

[105] J. Ye, Z. Zhu, and H. Cheng. What's your next move: User activity prediction in location-based social networks. In *SDM*, 2013.

[106] M. Ye, P. Yin, and W.-C. 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.

[107] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334. ACM, 2011.

[108] H. Yin, B. Cui, L. Chen, Z. Hu, and X. Zhou. Dynamic user modeling in social media systems. *ACM Transactions on Information Systems (TOIS)*, 33(3):10, 2015.

[109] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen. Lcars: A spatial item recommender system. *ACM Transactions on Information Systems (TOIS)*, 32(3):11, 2014.

[110] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, Q. V. H. Nguyen, and S. Sadiq. Discovering interpretable geo-social communities for user behavior prediction. In *The 32nd IEEE International Conference on Data Engineering*, 2016.

[111] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 221–229, 2013.

[112] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194. ACM, 2012.

[113] 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.

[114] Q. Yuan, G. Cong, and A. Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 659–668. ACM, 2014.

[115] J. Zhang, X. Kong, and P. S. Yu. Transferring heterogeneous links across location-based social networks. In *ACM International Conference on Web Search and Data Mining*, pages 303–312, 2014.

[116] 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 334–343. ACM, 2013.

[117] J.-D. Zhang and C.-Y. Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452. ACM, 2015.

[118] J. D. Zhang and C. Y. Chow. Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. *IEEE Transactions on Services Computing*, (1):1–1, 2015.

[119] J.-D. Zhang, C.-Y. Chow, and Y. Li. Lore: exploiting sequential influence for location recommendations. In

Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 103–112. ACM, 2014.

[120] J.-D. Zhang, C.-Y. Chow, and Y. Zheng. Orec: An opinion-based point-of-interest recommendation framework. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1641–1650. ACM, 2015.

[121] W. Zhang and J. Wang. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1221–1230. ACM, 2015.

[122] S. Zhao, I. King, and M. R. Lyu. Capturing geographical influence in poi recommendations. In *International Conference on Neural Information Processing*, pages 530–537. Springer, 2013.

[123] S. Zhao, I. King, and M. R. Lyu. A survey of point-of-interest recommendation in location-based social networks. *arXiv preprint arXiv:1607.00647*, 2016.

[124] S. Zhao, I. King, and M. R. Lyu. Aggregated temporal tensor factorization model for point-of-interest recommendation. *Neural Processing Letters*, 2017.

[125] S. Zhao, I. King, and M. R. Lyu. Geo-pairwise ranking matrix factorization model for point-of-interest recommendation. In *International Conference on Neural Information Processing*. Springer, 2017.

[126] S. Zhao, M. R. Lyu, and I. King. Aggregated temporal tensor factorization for point-of-interest recommendation. In *International Conference on Neural Information Processing*. Springer, 2016.

[127] S. Zhao, T. Zhao, I. King, and M. R. Lyu. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 153–162. International World Wide Web Conferences Steering Committee, 2017.

[128] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King. Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[129] T. Zhao and I. King. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1643–1652. ACM, 2016.