THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

POINT OF INTERESTS RECOMMENDATION IN LOCATION-BASED SOCIAL NETWORKS

Weiqing Wang

Master of Computer Science

A thesis submitted for the degree of Doctor of Philosophy at The University of Queensland in 2017

School of Information Technology & Electrical Engineering

Abstract

With the rapid development of location-based social networks (LBSNs), point of interests (POI) recommendation has become an important means to help people discover attractive and interesting locations from billions of locations globally. However, this recommendation is very challenging compared to the traditional recommender systems. A user may visit only a limited number of POIs, leading to a very sparse user-item matrix. This matrix becomes even sparser when the user travels to a distant place as most of the items visited by a user are usually located within a short distance from the user's home. Moreover, user interests and behavior patterns may vary dramatically across different time and different geographical regions. On the other hand, in reality, human movement exhibits sequential patterns. Thus, how to predict users' next move based on her previous visited locations is important and challenging in LBSNs. Our project focuses on offering a more accurate and efficient recommender system by overcoming the aforementioned challenges, and it contains the following three parts:

In the first part, we design ST-SAGE, a spatial-temporal sparse additive generative model for POI recommendation. ST-SAGE considers both personal interests of the users and the preferences of the crowd in the target region at the given time by exploiting both the co-occurrence patterns of POIs and the content of POIs. To further alleviate the data sparsity issue, ST-SAGE exploits the geographical correlation by smoothing the crowd's preferences over a well-designed spatial index structure called *spatial pyramid*. To speed up the training process of ST-SAGE, we implement a parallel version of the model inference algorithm on the GraphLab framework.

The second part aims to leveraging sequential patterns in POI recommendation. However, this is very challenging, considering 1) users' check-in data in LBSNs has a low sampling rate in both space and time, which renders existing location prediction techniques on GPS trajectories ineffective; 2) the prediction space is extremely large, with millions of distinct locations as the next prediction target, which impedes the application of classical Markov chain models; and 3) there is no existing framework that unifies users' personal interests and the sequential influence of recently visited locations in a principled manner. In light of the above challenges, we propose a sequential personalized POI recommendation framework (*SPORE*) which introduces a novel latent variable *topic-region* to model and fuse sequential influence with personal interests in the latent and exponential space. The advantages of modeling the sequential effect at the topic-region level include a significantly reduced prediction space, an effective alleviation of data sparsity and a direct expression of the semantic meaning of users' spatial activities.

In the third part, we focus on speeding up the online top-k recommendation process. In this part, we design two methods. The first method is able to extract the exact top-k results. It is a scalable query processing technique for top-k recommendation developed by extending the threshold algorithm (TA), which accesses items from the K sorted lists at location l on time t and computes the top-k results by scanning the minimum number of POIs. The second method, which is called as asymmetric Locality Sensitive Hashing (ALSH) technique, approximates the top-k results with less time. It speeds up the online top-k recommendation process by extending the traditional LSH to Maximum Inner-Product Search (MIPS).

Declaration by Author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

- **Weiqing Wang**, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, Xiaofang Zhou, "ST-SAGE: A Spatial-Temporal Sparse Additive Generative Model for Spatial Item Recommendation", ACM Transactions on Intelligent System and Technology, 2016 (**TIST**)
- **Weiqing Wang**, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, Xiaofang Zhou, "SPORE: A Sequential Personalized Spatial Item Recommender System", The 32nd IEEE International Conference on Data Engineering, 2016 (**ICDE**)
- **Weiqing Wang**, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, Xiaofang Zhou, "Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation", Proc. of 2015 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2015 (**KDD**)
- **Weiqing Wang**, Shazia Sadiq, Xiaofang Zhou, "EISA: An Efficient Information Theoretical Approach to Value Segmentation in Large Databases", The 16th Asia Pacific Web Conference, 2014 (**APWeb**)
- Hongzhi Yin, Liang Chen, **Weiqing Wang**, Xingzhong Du, Quoc Viet Hung Nguyen, Xiaofang Zhou, "Mobi-SAGE: A Sparse Additive Generative Model for Mobile App Recommendation", The 33nd IEEE International Conference on Data Engineering, 2017 (**ICDE**)
- Hongzhi Yin, Bin Cui, Xiaofang Zhou, **Weiqing Wang**, Zi Huang, Shazia Sadiq, "Joint Modeling of User Check-in Behaviors for Real-time Point-of-Interest Recommendation", ACM Transaction on Information Systems, 2016 (**TOIS**)
- Hongzhi Yin, Bin Cui, Xiaofang Zhou, **Weiqing Wang**, Zi Huang, Shazia Sadiq, "Joint Modeling of Users' Interests and Mobility Patterns for Point-of-Interest Recommendation", The ACM Multimedia Conference, 2015 (**ACM-MM**)
- Tieke He, Zhenyu Chen, Jia Liu, Xiaofang Zhou, Xingzhong Du, **Weiqing Wang**, "An Empirical Study on User-Topic Rating Based Collaborative Filtering Methods", World Wide Web Journal, 2016 (**WWWJ**)

Publications included in this thesis

Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, Xiaofang Zhou, "Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation", Proc. of 2015 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (**KDD'15**) - incorporated in Chapter 3.

Contributor	Statement of contribution
Weiqing Wang	Experiment design and conduction (80%) Paper writing (70%)
Hongzhi Yin	Experiment design and conduction (20%) Paper writing (15%)
Ling Chen	Paper writing (5%)
Yizhou Sun	Proof reading (40%)
Shazia Sadiq	Paper writing (10%)
Xiaofang Zhou	Proof reading (60%)

Weiqing Wang, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, Xiaofang Zhou, "SPORE: A Sequential Personalized Spatial Item Recommender System", The 32nd IEEE International Conference on Data Engineering (**ICDE'16**) - the model is incorporated in Chapter 4 and the online efficiency part is incorporated in Chapter 5..

Contributor	Statement of contribution
Weiqing Wang	Experiment design and conduction (90%) Paper writing (80%)
Hongzhi Yin	Paper writing (10%)
Shazia Sadiq	Paper writing (5%)
Ling Chen	Paper writing (5%)
Min Xie	Experiment design and conduction (10%)
Xiaofang Zhou	Proof reading (100%)

Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, Xiaofang Zhou, "ST-SAGE: A Spatial-Temporal Sparse Additive Generative Model for Spatial Item Recommendation", ACM Transactions on Intelligent System and Technology (**TIST'16**) - the model is incorporated in Chapter 3 and the online efficiency part is incorporated in Chapter 5.

Contributor	Statement of contribution
Weiqing Wang	Experiment design and conduction (90%) Paper writing (80%)
Hongzhi Yin	Experiment design and conduction (10%) Paper writing (10%)
Ling Chen	Paper writing (10%)
Yizhou Sun	Proof reading (20%)
Shazia Sadiq	Proof reading (40%)
Xiaofang Zhou	Proof reading (40%)

Contributions by others to the thesis

For all the published research work included in this thesis, Dr Hongzhi Yin and Prof. Shazia Sadiq, as my principle advisors, both have provided very helpful insight into the overall as well as the technical details and research problems; guidance for problem formulation as well as constructive comments and feedback. They also assisted with both the refinement of the idea and the pre-submission edition.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgments

I would like to express my special thanks to my principal supervisors, Dr. Hongzhi Yin and Prof. Shazia Sadiq, for their generous support, and their valuable and in-depth guidance for my PhD study and research. In the past three years, I learned a lot from them. With their help, I learned how to discover fresh and intriguing research topics, how to peer-review publications, and how to write high-quality papers. The three-year research experience will also be of great benefit in my future career.

I would also like to thank my associative supervisor, Prof. Xiaofang Zhou. I appreciate all his contributions of time and ideas, to make my Ph.D. experience stimulating and rewarding. He has offered a lot of support beyond the research.

In addition, I thank the following people from the DKE group in the School of Information Technology and Electrical Engineering, for their insightful discussions with me: Prof. Xue Li and Dr. Helen Huang. I also want to thank the PhD students I worked with. I really appreciate having had the opportunity to work with them, and their friendships will always be remembered in my life.

Last but not least, I will express my thanks to my husband and son. My husband always gave me the encouragement to help me get through times that were hard, sad and confused. During the last three years, I feel that we both learned a lot about life and strengthened our commitment and determination to each other and to live life to the fullest. My son is the best present that I have ever had. He gave me the courage to get through all the confusions.

Keywords

machine learning, data mining, user modeling, recommender system, social media, spatial data, temporal data, sequential patterns, top-k recommendation

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080109, Pattern Recognition and Data Mining, 40%

ANZSRC code: 080607, Information Engineering and Theory, 30%

ANZSRC code: 080603, Conceptual Modeling, 30%

Fields of Research (FoR) Classification

FoR code: 0806, Information Systems 50%

FoR code: 0801, Artificial Intelligence and Image Processing, 40%

FoR code: 0803, Computer Software, 10%

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we give a brief introduction of the research in this thesis, including the background, problem characterization, contributions, and the organization of the thesis.

1.1 Background

The rapid development of Web 2.0, location acquisition and wireless communication technologies have fostered a number of *location-based social networks (LBSNs)*, such as Foursquare, Gowalla, Facebook Places and Loopt, where users can check-in at venues and share life experiences in the physical world via mobile devices [3]. According to the survey[1], about 52% of Americans access social networks on mobile devices by January, 2016. On the other hand, "Pew Research Center" also shows that, by January of 2016, nine in ten smart-phone owners use location-based services in US.

In [4], LBSNs are defined as social networks in which GPS features of mobile devices are used to locate users and they let users share their location and other information from their mobile devices. Gao et al. state in [47] that LBSNs consist of "4W" (i.e., who, when, where and what) information [79], as shown in Figure 1.1. The content information (*what*) refers to tags, images, videos, comments, and posts generated or uploaded by users on social media, which represents the semantic information of users' activities. *Who* refers to the extensive knowledge about an individual, such as her demographics and social structure. The location information (*where*) bridges the gap between online social networks and the physical world. The temporal information (*when*) also provides rich contextual information of users' activities. These information are also called as "LBSN context".

A point-of-interest (POI) is a specific location (e.g., hotel, restaurant, museum, store) that a user

[1]https://www.statista.com/topics/2478/mobile-social-networks/

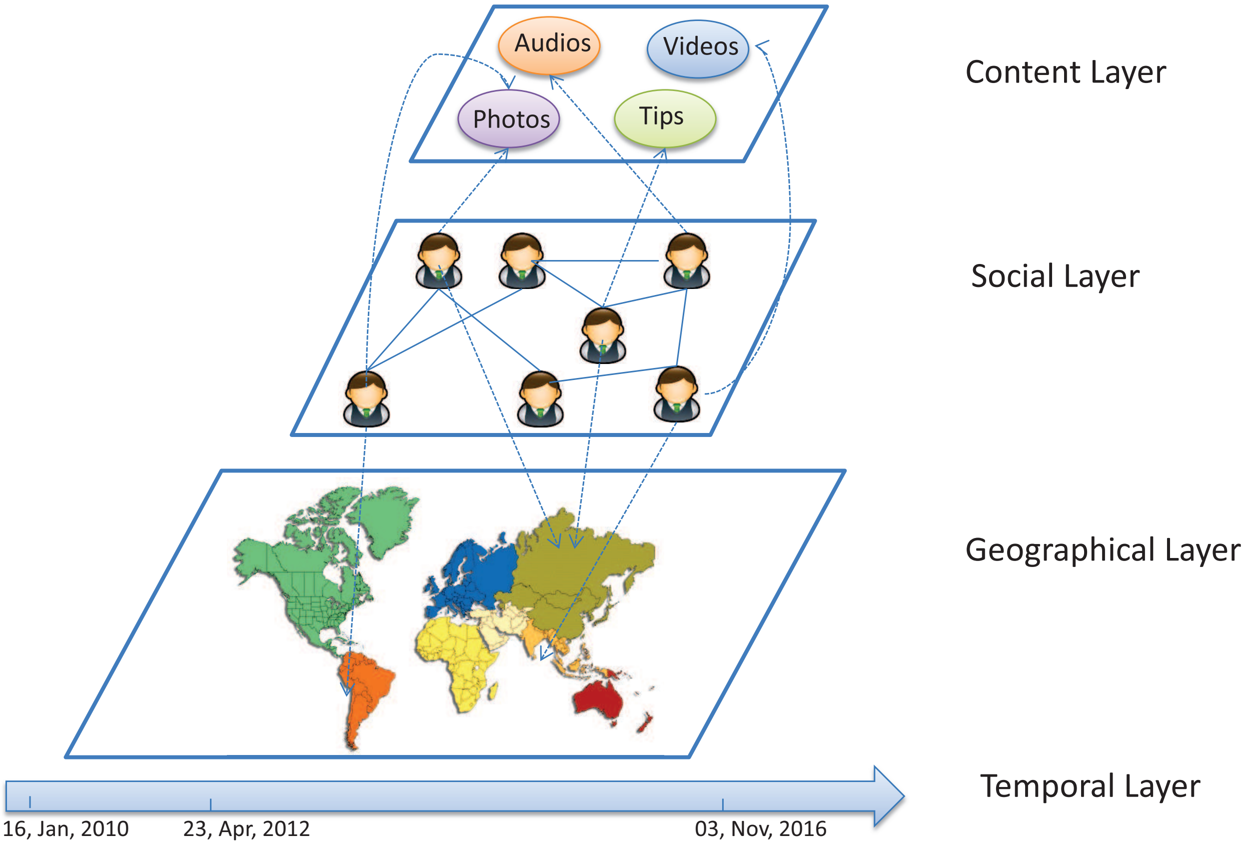


FIGURE 1.1: The Information Layout of LBSNs

may find useful or interesting. The rapid growth of cities has developed an increasing number of POIs, e.g., restaurants, theaters, stores, hotels, to enrich peoples life and entertainment, providing us with more choices of life experience than before. Making a satisfying decision efficiently among the large number of POI choices becomes a tough problem for a user. To facilitate a user's exploration and decision making, POI recommendation has been developed as an essential function in LBSNs [18, 4, 42]. Given a user and a set of POIs she has checked-in, POI recommendation aims at recommending her some POIs for her future visits based on the LBSN context related to this user [47].

1.2 Problem formulations

In this paper, we focus on the problem of POI recommendation by mining their historical behavior data in LBSNs. For ease of presentation, we define the key data structures and notations used in this paper, which are summarized in Table 1.1.

Definition 1 ***(POI)*** *A POI is a geographical point with specific functions (e.g., hotel, restaurant, museum, store) that user may find useful or interesting.*

Variable	Interpretation
U, R, V	the set of users, locations and POIs
W	the vocabulary set
D_u	the profile of user u
$v_{u,i}$	the POI of i^{th} record in D_u
$l_{u,i}$	the location of POI $v_{u,i}$
l_u	the home location of the user u
$W_{u,i}$	the set of words describing POI $v_{u,i}$
$w_{u,i,n}$	the n^{th} content word describing POI $v_{u,i}$
$t_{u,i}$	the time of i^{th} check-in record in D_u

TABLE 1.1: Notations of The Input Data

In our model, a POI has three attributes: identifier, location and content. We use v to represent a POI identifier, l_v to denote its corresponding location identifier and W_v to represent the set of words describing the POI (e.g., tags and categories). The location information available for each POI v in the collected raw datasets is in the form of the (latitude, longitude) pair. Then, index structures are applied to partition and index the entire geographic area. The choice of index structures depends on the nature of the applications. The details of the index structures are given in the chapters describing the proposed models.

Definition 2 ***(User Activity)*** *A user activity is made of a five tuple* $(u,\ v,\ l_v,\ W_v,\ t)$ *which indicates that the user* u *visits the POI* v*, located at* l_v *and described as* W_v *at the time* t*.*

Definition 3 ***(User Profile)*** *For each user* u*, we create a user profile* D_u*, which is a set of user activities associated with* u*. The dataset* D *used in our model consists of user profiles, that is,* $D = \{D_u : u \in U\}$.

Then, given a dataset D as the union of a collection of user profiles, we aim to provide POI recommendation for users. We formulate our problem as follows.

Problem 1 ***(POI Recommendation)*** *Given a user activity dataset* D*, a target user* u *with his/her current location* l *and current time* t *(that is, the query is* $q = (u,\ l,\ t)$*), our goal is to recommend a list of POIs that* u *would be interested in.*

In the last decade, recommender systems have been widely studied for various applications, including music recommendation, book recommendation, paper recommendation, etc. Various techniques have been developed for effective recommendation, e.g., collaborative filtering, matrix factorization, LDA, graph-based methods, etc. However, due to the specificity of human mobility on LBSNs, it is not sufficient to apply the classic methods in recommending POIs. We will present the specific challenges of POI recommendation in LBSNs in the following section.

1.3 Challenges

Data sparsity has been a severe challenge in many recommender systems. This is also one of the most important problems in POI recommendation. There are millions of POIs in LBSNs, but a user can only visit a limited number of them. For POI recommendation, this problem gets more severe in two important scenarios: out-of-town recommendation and sequential recommendation. Existing research on personalized POI recommendation mainly explores the geographic influence to improve the recommendation accuracy, based on the observation that the geographic proximity between spatial items affect users check-in locations [42, 43]. Recently, there are works that further integrate social influence in LBSNs to recommend items as common interests shared by social friends [29]. In terms of the temporal effect of user check-in activities in LBSNs, most existing work only investigate the temporal cyclic patterns of checkins [22]. To our knowledge, there are few works focusing on making recommendation in out-of-town scenario or sequential recommender systems in LBSNs.

1.3.1 Out-of-Town Recommendation

POI recommendation becomes more important and useful when a user travels to an unfamiliar area, where he/she has little knowledge about the neighborhood. In this scenario, the recommender system is proposed as *recommendation for out-of-town users in LBSNs* in [18]. POI recommendation in this scenario is a highly challenging problem because of the following three main reasons.

1. *Travel Locality*. While LBSNs expand rapidly, the number of POIs visited by an individual user is rather small compared to the total number of POIs in a LBSN, which results in a very sparse user-item matrix. Moreover, the observation of travel locality exacerbates this problem. The observation of travel locality [39] shows that most of users check-ins are left in their living regions (e.g., home cities). An investigation shows that the check-in records generated by users in their non-home cities only take up 0:47% of the ones generated in their home cities [39]. This observation aggravates the data sparsity problem with POI recommendation for out-of-town users (e.g., if we want to recommend POIs located at Los Angeles to people from New York City).

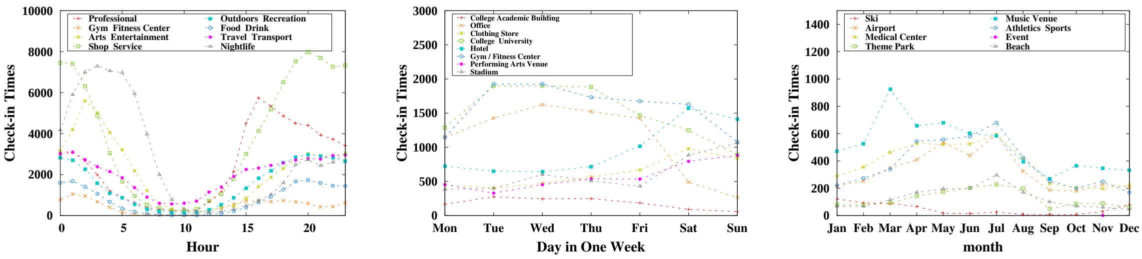
Illustration of Spatial Dynamics of User Interests

City	Top Item Types	Percentage of Check-ins(%)
Gold Coast (AU)	Beach	71.36%
	Surf Spot	14.82%
	Theme Park	9.60%
Las Vegas (US)	Casino	80.32%
	Nightlife	10.61%
	Outlet	5.82%
Boston (US)	College	78.32%
	Museum	9.45%
	Park	7.65%

2. *Spatial Dynamics of User Behavior.* When a user travels to a different region, his/her behavior patterns may change. Through Foursquare API[2], we extract the top three categories of check-in POIs in three different cities including Boston, Las Vegas and Gold Coast for a group of users. The group size is 3000 and each user in this group has check-in records in all three cities. The percentage of check-ins of each POI category is shown in Table 1.3.1. We observe that when users are in Las Vegas, they are more interested in visiting Casino (80.32%), Nightlife (10.61%) and Outlet (5.82%), while the same users prefer Beach (71.36%), Surf Spot(14.82%) and Theme Park (9.60%) when in Gold Coast.

3. *Temporal Dynamics of User Behavior.* Temporal influence plays an important role in analyzing users' daily activities in LBSNs [84, 56, 13, 37]. For example, a user is more likely to go to a restaurant rather than a bar at noon. Therefore, the recommendation should be time-aware. To illustrate the temporal dynamics of user behavior in LBSNs, we analyze the check-in distributions in time slices divided by different granularities on Foursquare. Following [22], we analyze people's check-in behavior in three granularities: daily (hour in one day), weekly (day of week) and yearly (month in one year). There are 36 categories on Foursquare and we show the check-in distributions of 8 selected categories for each granularity. From Figure 1.2(a), we can see that most categories exhibit a similar daily pattern - the check-ins start to decrease since midnight and reach the minimum at 10 am. The check-in activities start to rise at 10 am and peak in the afternoon or evening. But the check-in times of some categories, i.e., nightlife spot, peak in the midnight. Figure 1.2(b) tells that people tend to visit some places (such as gym, college, office and so on) in the weekdays while prefer

[2] `https://developer.foursquare.com/`



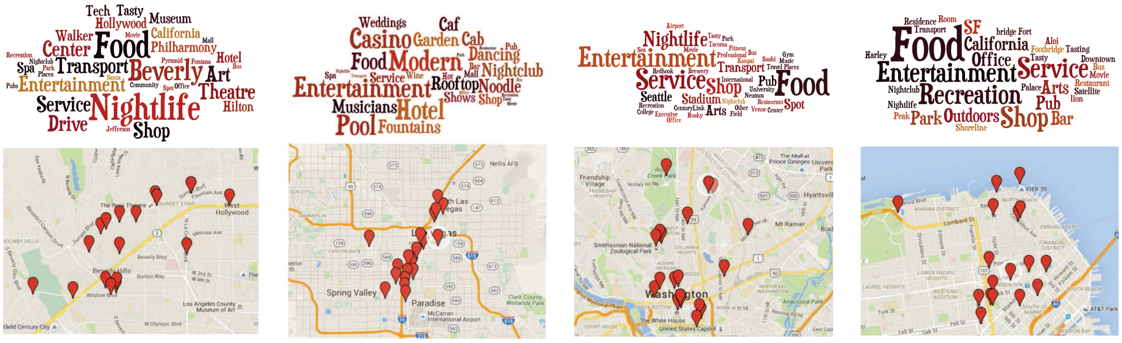
(a) Behavior Drift in One Day (b) Behavior Drift in One Week (c) Behavior Drift in One Year

FIGURE 1.2: Illustration of Temporal Dynamics of User Behavior on Foursquare

other places (i.e., clothing stores and performing arts venues) in the weekends. Monthly pattern refers to that people prefer to visit different locations in different months. For example, people prefer to go skiing in the winner months while go to beach in the summer months in Figure 1.2(c). To model the temporal influence, an intuitive solution is splitting time into time slices at the predefined granularity (e.g., hourly or seasonal) and then modeling the temporal preference to POIs of a user in a time slice by the items visited by the user in the time slice. However, splitting time into slices will make the data sparser.

The traditional collaborative filtering-based (CF-based) recommendation methods which make recommendations to a user based on similar users preferences perform poorly in the problem of out-of-town recommendation. For example, there would be very few activity records in Los Angeles for a user living in Pittsburgh but traveling in Los Angeles. Similar users would be those who exhibit similar location check-in behaviors, most of whom also live in Pittsburgh. Thus, most of the spatial items recommended would be in Pittsburgh instead of Los Angeles. Latent factor models also fail to solve these problems. Suppose we use existing topic models (e.g., LDA) to analyze user activity history data on Foursquare. A user is viewed as a document, and the items visited by her are viewed as words in the document. Each document (user) is a mixture of a small number of topics.We study both the semantic and spatial property for each topic. For each topic, we choose the top-20 spatial items with the highest generation probabilities. For the semantic analysis, we employ wordle[3] to create word clouds for each discovered topic based on the tags of the top-20 spatial items. For the spatial analysis, we present the locations of the top-20 spatial items on Google Maps for each topic. We present four example topics in Figure 1.3. From the results, we observe that the spatial items in one topic share similar geographical information but diverse semantic information. Two spatial items are grouped into the same topic because they are geographically close to each other so that they are frequently

[3]http://www.wordle.net/



(a) Topic 1 (b) Topic 2 (c) Topic 3 (d) Topic 4

FIGURE 1.3: Topics Discovered by LDA on Foursquare

visited together, rather than being represented as the same theme. For a user living in Beijing but traveling in Shanghai, therefore, the favorite topics discovered by LDA still capture the spatial items located in Beijing; that is, there is a failure in recommending spatial items for out-of-town users.

Recently, several methods [3, 18, 81] have been developed to make recommendations for both home-town and out-of-town users. These approaches either do not address all three challenges for POI recommendation, or address the challenges with ineffective strategies. For example, Ference et al. proposed a CF-based method that considers both users who have visited many common POIs with the target user and friends mined from the target user's social connections in LBSNs in [18]. Including the visiting records of social friends is supposed to handle the travel locality. For example, when similar users cannot provide effective clues for out-of-town recommendations, the visiting records of social friends will be used. Nonetheless, according to the survey in [13], when the user travels more than 100km, the check-in probability at the same place visited by any of the social ties, is merely 10%. Bao et al. presented a recommender system considering both user personal interests and the opinions of local experts in [3]. They model individual users' interests based on the category information of POIs. The opinions of local experts are used to solve the *travel locality* problem. Unfortunately, since the chosen local experts are users who share same or similar interests with the target user, this approach is unable to address *spatial dynamics of user behavior*.

1.3.2 Leveraging Sequential Information

Human movement exhibits sequential patterns, which serve as the basis for mobility prediction [13, 66]. In particular, an analysis has been conducted on three publicly available real-world datasets, Foursquare, Gowalla and Brightkite in [87], which calculates the probabilities of each of the next POIs immediately visited by a user after visiting a given POI. The results are shown in Figure 1.4.

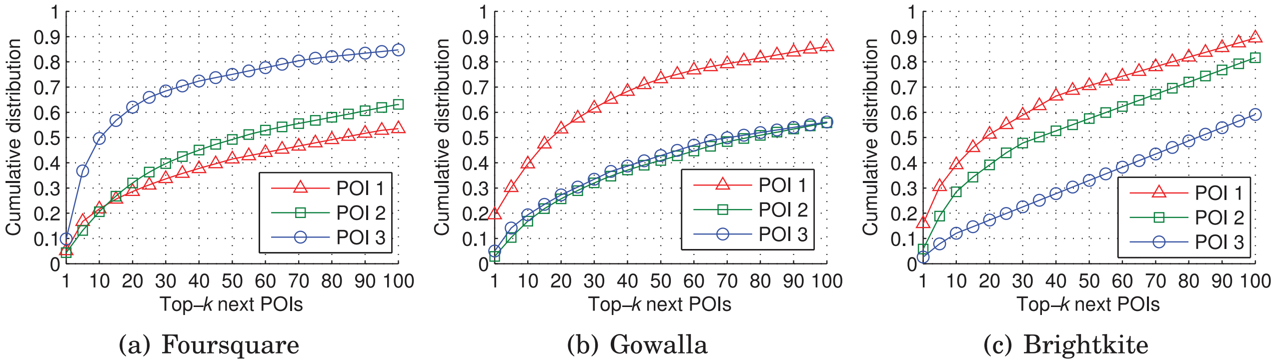


FIGURE 1.4: The Cumulative Distribution of the Directly Next POIs from a POI in the Real-World Datasets

These results show that each selected POI transits to the top hundred items out of several hundred thousand items with a probability greater than 0.5. The nonuniform distribution of transition probabilities between POIs indicates there are underlying sequential patterns between POIs visited by users. These sequential patterns result from different factors, such as time in one day (e.g., people tend to go to restaurants at dinner time and then relax in cinemas or bars at night [87]), geographical proximity (e.g., tourists often sequentially visit London Eye, Big Ben and Downing Street [83]) and the coherence between human preference and the type of places (e.g., people usually check in at a stadium before a restaurant instead of the reverse way because it is not healthy to exercise right after a meal [29]).

Nevertheless, leveraging sequential information for POI recommendation is highly challenging, mainly due to the following problems:

Low-sampling rate. There are a number of studies that predicate locations on GPS trajectories [97, 67]. At first glance, these approaches can be directly applied to LBSN data, since both the GPS and LBSN data contain location and time information. However, analysis [11] of the check-in records collected from Gowalla, a popular LBSN, shows that LBSN data has a low sampling rate in both space and time, compared to GPS trajectories. According to the analysis, only 10% of users have more than 58 check-in records over a 12-month period, representing a low check-in frequency over time. In addition, 40% of all consecutive check-ins have a spatial distance larger than 1 kilometer, much longer than the gap in GPS trajectories which is typically 5-10 meters [97]. Thus, it is difficult to model the dependency between two check-in locations in LBSNs using the location prediction techniques on GPS trajectories.

Huge prediction space. Sequential recommendation methods have been proposed in the literature [10, 88, 98], most of which are based on Markov chains. Suppose there are a collection of V POIs and the next item depends on the previous n items. The sequential recommendation methods then need to estimate $|V|^{n+1}$ free parameters in the nth order Markov chain model, which is extremely computational-expensive. To reduce the size of the prediction space, most related studies [10, 98] exploit sequential influence using a first-order Markov chain, which considers only the last one in a sequence of locations visited by a user to recommend a new location for her. Although the parameter space can be decreased to $|V|^2$, it may still be huge considering that V is usually a large number in LBSNs. Hence, we aim to develop a new method to incorporate the influence from all recently visited locations, rather than just the last one, to make location recommendations within a small predication space.

Unifying personalization and sequential effect. On one hand, most of existing POI recommendation methods focusing on personalization [3, 18, 13] make recommendations according to users' personal interests, but neglect the sequential orders between POIs. On the other hand, existing sequential recommendation methods, such as Markov chain based approaches, capture sequential patterns by assuming equivalent transition probabilities between items for all users, and ignore personalization. A recommendation system that only focuses on one of the two aspects may not produce ideal results. Therefore, we aim to develop a recommendation method which combines both personalization and sequential influence between items, in a unified and principled manner.

1.4 Contributions

1.4.1 ST-SAGE

To deal with the challenges in out-of-town recommendation, we propose ST-SAGE, a spatial-Temporal Sparse Additive Generative Model for Spatial Item Recommendation. ST-SAGE takes into account both user interests and spatial-temporal dynamics of user behavior. ST-SAGE learns user interests as vectors over a set of latent topics, by mining the co-occurrence patterns of spatial items and their content information (e.g., tags and categories). Exploiting the content information of spatial items addresses the travel locality for out-of-town recommendation. The content of spatial items serves as the medium for transferring user interests learned from the home town to unfamiliar regions.

To adapt to *spatial-temporal dynamics of user behavior*, ST-SAGE recognizes two roles of an individual user in a spatial region at a specific time: local or tourist. Given a location and a specific time, visiting records from local users are mined to learn *temporal native preference* as a vector over latent topics. Similarly, visiting records from tourists are used to learn *temporal tourist preference*. Users with the same role at a location are more likely to have similar preferences and behavior patterns at a specific time. Thus, to recommend spatial items to a target user u at location l and time t, we consider not only u's personal interests, but also the temporal preference of the crowd who have the same role as u. In this way, we leverage the crowd's temporal preferences to overcome the sparsity of individuals activity data generated out of town at a specific time. Specifically, given a user, a specific time and her current location, we first find the group of users sharing the same role with u when they visited l, and then produce the time-aware recommendation based on their temporal preferences and the user's personal interests.

By using the visiting history to model native preference and tourist preference, however, we face the data sparsity issue, especially when the target region is small. When there are insufficient visiting records in the target region, the two variables cannot be inferred accurately. To overcome the data sparsity, ST-SAGE integrates the spatial index - *spatial pyramid* which is a tree structure proposed in [39]. It is first constructed by partitioning locations of spatial items into spatial grids of varying sizes at different hierarchies. Then, ST-SAGE applies the *additive* framework [2, 33] to learn the native preference and tourist preference of each region. Briefly, when learning the native preference and tourist preference in a region, the two variables learned for all of the region's ancestor grids in the spatial pyramid will be added. If there are few or no activities in a region, we can still infer native preference and tourist preference guided by its ancestor grids. Through integrating with the spatial pyramid, ST-SAGE gains another advantage by allowing users to switch between different scales of geo-regions (e.g., zoom in/out on a Google Map) without re-learning the parameters. In this way, this model can be seamlessly connected with Google Map.

1.4.2 SPORE

To incorporate the sequential influence in POI recommendation, we propose SPORE, a sequential personalized spatial item recommender system. SPORE seamlessly fuses the sequential influence of visited spatial items and the personal interests of individual users in a principled way. Technically, SPORE is a latent class probabilistic generative model designed to mimic users' decision-making process for choosing spatial items. We model personal interests and sequential influence based on the latent variable *topic-region* in SPORE. A topic-region z corresponds to a semantic topic (i.e., a soft cluster of words describing spatial items) and a geographical region (i.e., a soft cluster of locations of spatial items) at the same time. The generative process of users' check-in behaviors in SPORE is briefly illustrated as follows. Given a target user u at time t, SPORE first chooses a topic-region z for u based on her personal interests and her visited items before t. The selected topic-region z in turn generates a spatial item v following z's semantic and geographical distributions.

By introducing the latent factor topic-region, SPORE effectively overcomes the challenge posed by *low-sampling rates*. Specifically, SPORE addresses the sparsity of LBSN data by considering the hidden variable topic-region, which groups spatial items with similar semantic meanings and geographic locations, rather than focusing on the fine granularity of data such as consecutive points in GPS trajectories.

Our proposed SPORE is able to reduce *the prediction space* effectively. In particular, for each spatial item v, we learn a distribution θ_v^{seq} over a set of topic-regions where each component $\theta_{v,z}^{seq}$ represents the probability of visiting the topic-region z after visiting v. An obvious advantage of predicting the topic-region of a user's activity at the next step is a significantly reduced prediction and model parameter space, because the number of topic-regions is much smaller than the number of spatial items. Additionally, to capture the influence from high order items, SPORE adds the influence of the previously visited items in the exponential space to avoid the inference of mixture weights for each visited item, inspired by the the Sparse Additive Generative model (SAGE) [16]. In this way, SPORE accurately captures the influence from more items previously visited by the target user, and at the same time reduces the exponential complexity $|V|^{n+1}$ of the classic nth order Markov Chain into linear complexity $|V| \times K$ (K is the number of topic-regions).

1.4.3 Efficient On-line Recommendation

For both models, in the online recommendation phase, given a query user u and her current location l on time t, the system computes a ranking score for each POI v at location l and then returns the top-k POIs with highest scores. To speed up the process of online recommendation, a scalable query processing technique for top-k recommendation is developed by extending the threshold algorithm (TA), which accesses items from the K sorted lists at location l on time t and computes the exact top-k results by scanning the minimum number of spatial items. The TA algorithm returns the exact top-k spatial items with the highest scores. However, in reality, the users may be more interested in a quicker response with an approximate top-k result. On the other hand, the TA algorithm needs to maintain and access K sorted lists of items and frequently update the threshold, which makes it slow when K is large.

Locality Sensitive Hashing (LSH) [15] based techniques are common and successful in industrial practice for solving the KNN problem efficiently. Both the running time and the accuracy guarantee of LSH based KNN are in a way independent of the dimensionality of the data. Furthermore, LSH is massively parallelizable, which makes it ideal for large modern datasets. Although LSH is popular in both Euclidean distance and Cosine similarity, there are few work extending LSH to the Maximum Inner-Product Search (MIPS), which is the case in our model. Thus, we propose an algorithm calles as asymmetric LSH by applying two different hash functions to the spatial items and the queries respectively to transform the MIPS into classic nearest neighbor search.

1.4.4 Summaries

The main contributions of our work are summarized as follows.

- We design a novel spatial-temporal sparse additive generative model (ST-SAGE) for POI recommendation which incorporates and exploits content information and the crowd's temporal preference at a region to address the problems of travel locality and spatial-temporal dynamics of user's behavior.

- We design a novel sequential personalized POI recommendation framework, SPORE, which learns and fuses sequential influences and personal interests in a *latent* and *exponential* space, effectively overcoming the challenges of *low sampling rates*, *huge prediction space*, and *unifying personal interests and sequential effect* in a principled way.

- We develop two efficient top-k query processing techniques to speed up the process of online recommendation. One technique produces the exact top-k result by scanning the minimal number of items by extending the Threshold Algorithm (TA) and another one speeds up the process further but returns the approximate top-k result by extending locality sensitive hashing to the maximum inner-product search.

- We conduct extensive experiments to evaluate both the effectiveness and efficiency of the proposed models on two real datasets and one large-scale synthetic dataset.

1.5 Thesis organization

The rest of this thesis is organized as follows: In Chapter 2, we review the field of POI recommendation and summarize the relevant work. In Chapter 3 we present ST-SAGE which aims at dealing with the challenges in out-of-town recommendation. In Chapter 4, we incorporate the sequential patterns into POI recommendation. In Chapter 5, we introduce two techniques to accelerate the on-line recommendation. Finally the conclusion and the possible future research directions suggested by the thesis are given in Chapter 6.

Chapter 2

Literature Review

Recommender systems have been widely studied among various categories, such as movie recommendation on NetFlix, job recommendation on LinkedIn and item recommendation on Amazon. POIs recommendation belongs to a sub-category of recommender systems. As a result, technologies of general recommender systems are also practically applicable to POIs recommendation although the performance may be limited due to the specific properties of human mobility on LBSNs. In this chapter, we firstly give a literature review on general recommender systems, and after that, we review the existing methods of POIs recommendation.

2.1 General Recommender Systems

Recommender systems aim at helping users find items of interest among a large amount of items by generating personalized recommendation [1]. The existing methods can be generally classified into three categories: collaborative filtering, content-based, and hybrid models. Among them, collaborative filtering (CF) is one of the most successful approaches which has been proven effective in practice [60, 68]. The input of CF is a user-item rating matrix (i.e., user-POI check-in frequency matrix). The key idea of CF is that if two users have similar behavior pattern on the similar items (e.g., visiting similar POIs, buying similar products and preferring similar books), they will most likely behave similarly over the other items in the future. There are two main types of CF approaches: memory-based CF and model-based CF. The memory-based CF approach makes use of the whole user-item rating matrix to produce recommendations. This method has been widely adopted in many commercial systems. Generally, there are two approaches of memory-based CF: user-based [27] and item-based CF [60]. The key difference between these two approaches is whose similarity it relies on to produce the recommendation. User-based CF makes recommendation based on the preferences from K users most similar to him. Similarly, for a user u, item-based CF recommends K items which are most similar to the items that u likes before. These two techniques can be applied on POI recommendation. For example, we can apply user-based CF on POI recommendation as follows. Assume that we want to recommend N POIs to user u. First, select K most similar users to u. Then, infer the preferences of the similar users on the POIs not visited by u. At last, rank u's preferences on those unvisited POIs and select the top N POIs for recommendation.

Memory-based collaborative filtering approaches are efficient and easy to adopt. However, it has two shortfalls when applied to large-scale sparse data. The first one is the sparsity problem. In many real-world applications, the user-item matrix is usually very sparse with a density of 10^{-4} to 10^{-5}. Under sparse data, the inferred similarity measured from the activities history may not be reliable due to the insufficient information observed [57]. In an extreme case of "cold-start" problem, a new user who has no rating/check-in history would have the similarity value of 0 to any other users. The other shortfall is that memory-based CF is of poor scalability on large dataset as it leverages the whole user-item matrix to produce recommendation which requires a large storage space and also is computationally expensive.

In sight of the shortfalls of memory-based collaborative filtering, model based collaborative filtering techniques are proposed. Overall, they make use of data mining and machine learning techniques to learn a model from training data and then apply the model on test data to predict user interests on different items. Various models have explored in this model including latent factor models, classification/regression models and so on. Among all these models, latent factor models such as the matrix factorization model have been widely applied [9].

The basic idea of the matrix factorization method is to assume that there are certain latent factors related to both users' preferences and items' properties. Take the restaurant in POI recommendation as an example, latent factors could be price, flavor, environment and so on. These latent factors may dominate the occurrences of major check-in activities. Each check-in is a result of the combinational effect of user's preferences and location's property on these factors. For instance, a user who likes Japanese food and concerns the price may be interested in a restaurant that serves Japanese food with a discount price.

The recommender systems based on classification/regression extract the features for both users and items to construct the feature space on training data. The observed pair of user and item (i.e., a user has checked-in at that POI) is assigned with a positive label while the unobserved pair is assigned with a negative label. Based on these labels, a classification/regression model is learned based on the training data with certain learning models including decision tree, logistic regression and so on. Then the learned model is applied on the test data to infer the likelihood of a user's preference in an item [55].

Other than the user-item matrix, hybrid CF leverages other information, for example, content information and combines CF with the content-based recommender systems. The results from each recommender system are weighted and combined into the final result for recommendation.

2.2 POI Recommendation

POI recommendation, also called location or place recommendation, has been considered as an essential task in the domain of recommender system.

2.2.1 POI Recommendation with GPS Data

The task of POI recommendation is highly related to human mobility. It was traditionally studied on mobile data, i.e., cellphone-based GPS data. As we know, cellphones have been widely used to facilitate humans' life. They can be considered as mobile sensors of human beings as generally, a user takes his/her cell phone with him/her most of the time. These mobile sensors can provide abundant information about human mobility. Typically, the cellphone-based GPS data contains a set of time-stamped GPS points that a user has been to, along with the mobile activities such as listening to music, watching videos and so on. Thus various work have explored these information to study the human mobility, which promote a set of location-based applications including POI recommendation [5, 28, 58, 65, 69].

The raw GPS data is a sequence of time-stamped latitude/longtitude pairs and there is no mapping information between geographical coordinates and specific real world POIs. Thus a POI is usually determined by the stay points which a user spent sufficient long time extracted from users' GPS trajectory logs [92, 97]. Since content and social information is usually not available on such datasets, spatial and temporal patterns are commonly adopted with collaborative filtering methods to perform POI recommendation. Zheng et al. [96] proposed a recommendation framework for location recommendation taking advantage of sequence property, region popularity and hierarchical property of geographical spaces. Yong et al. recommend locations and travel packages with GPS trajectory data based on collaborative filtering method [26]. Ye et al. [78] studied life patterns for each individual from GPS data and then used these patterns to produce location prediction and recommendation. Zheng et al. [95] consider both the location interests and users' travel experience with a tree-based hierarchical graph to build a HITS-based inference model. Leung et al. [38] studied different user classes and temporal preferences for collaborative location recommendation with a dynamic clustering algorithm.

The content information could be available in certain types of GPS data. Zheng et al. [92, 93, 94] proposed a user-centric collaborative location and activity filtering approach to find like-minded users and similar activities at different locations with tensor decomposition. The dataset is collected through voluntary users while user comments regarding to geographical visits are used. In tour recommendation [25, 48] and tourist POI recommendation [34], content related to travel package or tourist POIs, such as package description or POI attributes, are used to analyze users interested topics for POI recommendation.

The data availability of GPS data is limited due to the user privacy problem as the GPS data is obtained from users' cell phones through telecommunication services. Most users do not feel comfortable to share their mobile data even for research purposes. Thus, GPS data usually contains a limited number of users over long period. Gao et al. [21] summarized the characteristics and limitations of leveraging GPS data for POI recommendation, as three aspects. First, the available mobility data is small-scale due to the user privacy problem. The inference on such limited data may be biased due to various factors such as demography, gender and age. In the big data era, more data are helpful in analyzing human mobility to get a more accurate statistically significant conclusions. Second, GPS data has no semantic indications as it only includes the geographical coordinates. Thus, it is not easy to associate such coordinates with real-world points of interests, such as casinos, restaurants, shopping centers and so on. Although one can use third-party library to map coordinates into POIs, it does not work well on places with dense POIs, as it is difficult to distinguish POIs close to each other based on geographical coordinates. Third, social connections are not easily obtained from GPS data too. Even though social connections can be inferred through the history of one's phone calls, messages, or bluetooth connections. It is difficult to collect this kind of data due to the privacy concerns. There are some efforts collecting social information on GPS data through communication network or Bluetooth network with a number of participated users who grant permissions [40, 71]. However, social information obtained in this way may be in low quality. For example, Bluetooth may not be commonly used thus connections inferred through this way may be biased; users who have phone communications do not necessarily indicate their friendships, not to mention that they share common interests of locations.

2.2.2 POI Recommendation with LBSN Data

Recently, with the easy access of large-scale user activity records in LBSNs, many recent studies investigate the POI recommendation with LBSN data. Most existing work has tried to improve POI recommendation in LBSNs by exploiting the geographical influence, social influence, temporal effect and content information of POIs. Ye et al. [76] introduced POI recommendation into LBSNs. Most current work on POI recommendation on LBSNs focuses on taking advantage of both the geographical and social properties to improve the recommendation effectiveness for the strong correlations between geographical distance and social connections discovered in previous work [12, 13, 24, 61, 85]. These studies mainly exploit geographical influence and social connections separately, and then combine their output together with a fused model.

2.2.2.1 Geographical Influence

These works try to study how check-in probability is influenced by geo-distance by modeling check-in distribution over geographical distance. There are three main types of mobility patterns modeled in this series of work, including Levy Flight (Power-Law distribution of geographical distance), Multi-Center Gaussian Distribution and Kernel Density Estimation.

Levy Flight. This kind of mobility pattern is also called as Power-Law distribution of check-ins. This mobility pattern is based on several key observations. One observation is that people tend to move to nearby places and occasionally to distant places and 20% of consecutive check-ins happened within 1km while 60% between 1 and 10km [9, 13]. Another important observation is that a few places have many check-ins while most of places have few check-ins which is similar to that a user goes to a few places many times while to many places a few times [24]. Ye et al. [77] proposed to capture the geographical influence by investigating the geographical clustering phenomenon of user check-in activities in LBSNs by employing a power-law probabilistic model to capture the geographical influence among POIs, and realize collaborative POI recommendations based on geographical influence via naive Bayesian method. By modeling the check-in probability with a power-law distribution, they recommend the location with the highest probability to be co-visited with locations from the user's check-in history.

Multi-Center Gaussian Distribution. This distribution has different interpretations from different perspectives. From the geographical perspective, this distribution means that users' check-ins usually center on certain location areas and users rarely check-in at locations which are far away from the centers [13]. From the temporal perspective, this distribution represents that the probability of visiting a location centers on certain time periods and decreases during other time periods [22]. This can also be interpreted as biased probability decreasing speed around a center or various peeks at multiple centers. Cho et al. proposed a model which is an effective two-state mixture of Gaussians with a time-dependent state prior. This means that this model classifies each of the users check-ins as either being generated by the "home" or "work" state. The temporal part of the model governs the transition between home/work states and then depending on the state, geographic location of the check-in is generated the time varying mixture of two time-invariant 2-dimensional Gaussian distributions.

Kernel Density Estimation. Zhang et al. [86] proposed a Kernel density estimation method to model the geographical influence without knowing a specific type of distribution. There are three steps in this work. First, it computes distance between every pair of locations that have been checked-in by the user, denoted as D. Second, it defines the distance distribution with a kernel function. At last, it recommends locations with high probability according to the distance distribution.

2.2.2.2 Social Influence

There are two general types of POI recommendation exploiting social influence. The first one is the friend-based collaborative filtering proposed by Ye et al. [76]. This work assumes that more closed friends have more similar trajectory. Thus it infers the similarity between friends based on the similarity between their trajectories. It also assumes a power-law relation between trajectory similarity and geographical distance. The other type is to add social regularization into matrix factorization [51, 8]. This work fuses the similarity between two users with matrix factorization. Based on many existing works, we can observe that social information can consistently improve the recommendation performance, however, the improvement is very limited [24, 77].

2.2.2.3 Geo-Social Influence

Geo-Social influence indicates that people tend to explore nearby POIs of a POI that they or their friends have visited before [77]. Many recent studies [42, 9, 77, 13] showed that there is a strong correlation between user check-in activities and geographical distance as well as social connections, so most of current POI recommendation studies mainly focus on leveraging the geographical and social influences to improve recommendation accuracy. For example, Ye et al. [77] delved into POI recommendation by investigating the geographical influences among locations and proposed a system that combines user preferences, social influence and geographical influence. Cheng et al. [9] investigated the geographical influence through combining a multi-center Gaussian model, matrix factorization and social influence together for location recommendation. Lian et al. [42] incorporated spatial clustering phenomenon resulted by geographical influence into a weighted matrix factorization framework to deal with the matrix sparsity. Ference et al. [18] designed a collaborative recommendation framework which considers the activity records generated by both friends and similar users in a mixture way.

2.2.2.4 Temporal Effect

The temporal effect of user check-in activities in LBSNs has also attracted much research interest. POI recommendation with temporal effect mainly leverages temporal cyclic patterns and temporal sequential patterns on LBSNs.

Exploring Cyclic Patterns: Gao et al. [22] investigated the temporal cyclic patterns of user check-ins in terms of temporal non-uniformness and temporal consecutiveness. They make use of a hierarchical Pitman-Yor language model to capture the temporal chronological patterns with the consideration of power-law distribution and short-term effect. Ye et al. [75] introduced temporal dimension of daily and weekly check-ins to identify the types of unknown geographic target on LBSNs. Yuan et al. [84] also incorporated the temporal cyclic information into a user-based collaborative filtering framework for time-aware POI recommendation. While it has been shown in multiple studies that human movement in LBSNs clearly demonstrates sequential patterns [87, 83, 29], leveraging sequential patterns for spatial item recommendation has not been well-studied.

Leveraging Sequential Information. Existing works on sequential recommendation mostly utilize the Markov chain property to predict the next check-ins. Cheng et al. [10] exploited sequential influence using the first-order Markov chain that only considers the last location in a user's visiting sequence to recommend a new location for the user. However, in reality, the new location may not only rely on the latest location but also earlier ones visited by the user [88]. Zhang et al. predicted the next location by adding the influence of the earlier visited locations with an additive Markov chain. They manually set a decay rate parameter for previous locations based on the assumption that locations with recent check-in timestamps usually have stronger influence than those with old timestamps.

To adapt to the sparse data in LBSNs, Ye et al. [11] proposed modeling the sequential patterns of spatial items at the category level using a hidden Markov model (HMM). The abstract states in HMM can model sparse LBSN data well as the hidden states capture essential behavioral patterns of LBSN users. The accuracy of this method depends highly on the category information.

2.2.2.5 Exploring Content Information

Most recently, researchers explored the content information of POIs to alleviate the problem of data sparsity. Hu et al. [30] proposed a spatial topic model for POI recommendation considering both spatial aspect and textual aspect of user posts from Twitter. Yin et al. [81] exploited both personal interests and local preferences based on the contents associated with POIs. Liu et al. [44] studied the effect of POI-associated tags for POI recommendation with an aggregated LDA and matrix factorization method. Gao et al. [23] studied both POI-associated contents and user sentiment information (e.g., user comments) into POI recommendation and reported their good performance. Yang et al. [74] introduced sentiment information into POI recommendation and the evaluation results show that it is of better performance over the state-of-the-art approaches. Bo et al. [31] incorporated content information into social correlations and proposed a topic model for POI recommendation. Yuan et al. studied content information with its spatio-temporal patterns [84].

2.2.3 Efficient Online Recommendation

Many approaches are proposed to improve the online POI recommendation by reducing the number of spatial items to scan. These approaches can be divided into two categories: the ones offering exact top-k recommendation and the ones making approximate top-k recommendation.

2.2.3.1 Exact Top-k Recommendation

These work aim at pruning the item search space in finding the k items with highest scores.

Most existing methods in GPS data index the spatial items using a tree structure such as R-Tree and Metric Tree, which is similar to the techniques used in KNN problems [90, 91]. In the classical KNN problems, both the queries and items are represented as vectors with K dimensions. For each query, they aim at recommending those items that have nearest distance to the given query with the indexing structure. The distances between queries and items are measured in Euclidean distance.

Another line of research tries to adapt the techniques in the classical KNN problems to the models where the distances between queries and items are measured in inner-product [59]. This line of research notices that the straightforward application of the techniques in KNN problems is infeasible in the situation where distances are measured in inner-product as inner-products lack the distance coincidence which is a very basic property of coincidence that Euclidean distance usually has. According to the analysis in [59], the maximum inner-product search is equivalent to nearest-neighbor search in Euclidean metric space only if the norms of all the spatial items have the same length, which is, however, impractical in many models including both ST-SAGE and SPORE.

Even if the method in [59] can prune the item search space, its efficiency suffers from the curse of the dimension K which is $O(K^{12})$ [49]. Some researchers proposed the Threshold-based Algorithm (TA) to minimize the number of items to scan in finding the exact top-k results in [17, 81, 80]. This algorithm is instance optimal and there is no other deterministic algorithm that has a lower optimality ratio [17].

2.2.3.2 Approximate Top-k Recommendation

In reality, the users may be more interested in a quicker response with an approximate top-k result. Specifically, for a query, an approximate top-k recommendation returns k spatial items with approximate highest scores [80]. Yin et al. extend TA algorithm to an approximate top-k recommendation with a guarantee of a specific approximation to the accurate result in [80]. Locality Sensitive Hashing (LSH) [15] based techniques are another line of techniques which can be applied to achieve approximate top-k recommendations. LSH based methods are common and successful in industrial practice for solving the KNN problem efficiently. Both the running time and the accuracy guarantee of LSH based KNN are in a way independent of the dimensionality of the data. However, similar with the tree-based methods, it is infeasible in the situation where distances are measured in inner-product as it is designed for Euclidean metric space.

2.3 Summary

As described above, while there are many studies to alleviate the data sparsity by exploiting geographical-social influence, temporal effect and content information, they did not address the challenges arising from either *travel locality* or *interest drift* for the out-of-town recommendation. For example, most of the above work assumed that users are in their home towns, they did not consider user interest drift across regions. Our work in this paper distinguishes from previous work in several points. First, to the best of our knowledge, we are the first to simultaneously address the three challenges arising in out-of-town recommendation in a unified model. Second, although Yin et al. [81] exploited the local crowd's preferences, they ignored the user's role and did not distinguish native preference from tourist preference. Third, we propose a novel effective method to represent and infer both native and tourist preferences based on a well-designed spatial index structure.

In terms of integrating the sequential patterns, our work in this thesis distinguishes itself from previous research in several aspects. First, to the best of our knowledge, this is the first effort that automatically integrates sequential effect and personalization in a unified model. Second, although existing research [88] has also exploited the influence of more than one previously visited locations using an additive Markov chain method, the weights for each visited item in a sequence need to be set manually. In contrast, our proposed SPORE model adopts the sparse additive technique to add the influence of all visited items in the exponential space, which avoids the inference of item weights. Third, we introduce a novel latent variable *topic-region* to model both personal interests and temporal influence in the exponential space. The discovered topic-regions cluster both content-similar and geographically close items together. By introducing the latent factor topic-region, SPORE effectively overcomes the challenge brought by low-sampling rate and reduces the prediction space.

For the efficient online recommendation, we first extend the TA-based algorithm to achieve an exact top-k recommendation. Then we propose an asymmetric LSH (ALSH) for the following two reasons. First, the users may be more interested in a quicker response with an approximate top-k result. Second, the TA algorithm needs to maintain and access K sorted lists of items and frequently update the threshold, which makes it slow when K is large. ALSH apples two different hash functions to the spatial items and the queries respectively. The main idea of ALSH is to transform the approximate top-k recommendation in our problem into classic nearest neighbor search by introducing two hashing functions so that the probability of new collision event satisfies the conditions in the definition of KNN.

Chapter 3

Out-of-Town Recommendation

3.1 Overview

POI recommendation is very challenging compared to traditional recommender systems, especially when users travel to unfamiliar places. A user may visit only a limited number of spatial items, leading to a very sparse user-item matrix. This matrix becomes even sparser when the user travels to a distant place as most of the items visited by a user are usually located within a short distance from her home. Moreover, user interests and behavior patterns may vary dramatically across different time and different geographical regions. In light of this, we propose ST-SAGE, a spatial-temporal sparse additive generative model for spatial item recommendation in this chapter. ST-SAGE considers both personal interests of the users and the preferences of the crowd in the target region at the given time by exploiting both the co-occurrence pattern of spatial items and the content of spatial items. To further alleviate the data sparsity issue, ST-SAGE exploits the geographical correlation by smoothing the crowd's preferences over a well-designed spatial index structure called spatial pyramid. To speed up the training process of ST-SAGE, we implement a parallel version of the model inference algorithm on the GraphLab framework.

3.2 Preliminaries about SAGE

Our model is inspired by the Sparse Additive Generative Model (SAGE) [16], which is an effective generative model without explicit switching variables [30]. The basic idea of the model is that if a variable is affected by several components, it can be generated by the mixture of these components without any explicit indicator variables. The key difference from traditional mixture models is that the mixture occurs in terms of the natural parameters of the exponential family instead of distributions. Such a model is robust given limited training data as it does not have to infer a complex indicator variable to distinguish the set of causes.

To provide a clearer explanation of SAGE, we use a traditional probabilistic mixture generative model, LCA-LDA proposed in [81, 80], as an illustrative example. LCA-LDA is a location-content-aware model that aims to mimic the process of human decision making on spatial items. The model considers the user's personal interest $\boldsymbol{\theta}_u^{user}$ and the influence of local preference $\boldsymbol{\theta}_l^{crowd}$ [1] in a unified manner, and automatically leverages the effect of the two factors. Specifically, given a querying user u at a target location l, the likelihood that user u will prefer item v is computed by combining these two factors through a linear combination as follows.

$$P(v|\boldsymbol{\theta}_u^{user}, \boldsymbol{\theta}_l^{crowd}) = \lambda_u P(v|\boldsymbol{\theta}_u^{user}) + (1 - \lambda_u) P(v|\boldsymbol{\theta}_l^{crowd}) \tag{3.1}$$

where λ_u is the "switching" variable that needs to be inferred for each user. Other than the latent topic, we need to sample the switching variable for each record. Thus the complexity is doubled. Evidently, it cannot be inferred accurately when the training data for the individual user is sparse. In contrast, SAGE combines the two generative facets through simple addition in log space, as shown in Equation 3.2. Clearly, this avoids the need for latent switching variables.

$$P(v|\boldsymbol{\theta}_u^{user}, \boldsymbol{\theta}_l^{crowd}) = P(v|\boldsymbol{\theta}_u^{user} + \boldsymbol{\theta}_l^{crowd}) = \frac{exp(\boldsymbol{\theta}_{u,v}^{user} + \boldsymbol{\theta}_{l,v}^{crowd})}{\sum_{v'} exp(\boldsymbol{\theta}_{u,v'}^{user} + \boldsymbol{\theta}_{l,v'}^{crowd})} \tag{3.2}$$

3.3 Spatial Temporal SAGE Model

In this section, we first formulate the problem definition in this chapter, and then present our proposed spatial temporal SAGE model (ST-SAGE).

3.3.1 Problem Definitions

The notations in this chapter are the same with those listed in Table 1.1 if they are not defined specially in this section.

As we mentioned, the location information available for each POI v in the collected raw datasets is in the form of the (latitude, longitude) pair. Then a spatial pyramid structure [39], is applied to partition and index the entire geographic area in this chapter. The granularities range from cities to

[1] It does not distinguish between native preference and tourist preference

streets, depending on the nature of the applications. The details of the spatial pyramid are described in subsection 3.3.4.

Definition 4 ***(User Home Location)*** *Following the recent work of [41], given a user u, we define the user's home location as the place where the user lives, denoted as l_u.*

Due to privacy problem, user home locations are not always available. For a user whose home location is not explicitly given, we adopt the method similar to [61] by inferring the user's home location as the cell in the spatial pyramid with the most of his/her check-ins.

Definition 5 ***(Time)*** *In this chapter, time t is an ordinal variable and we use t to index the t^{th} time slice which corresponds to a specific time period.*

The time information available for each record in the collected raw datasets is in the form of timestamps (e.g., "2010-07-24, 13:45:06"), and we divide the timestamps into time slices at the predefined granularity (e.g., hourly or seasonal) when preprocessing the datasets. The model in this chapter focuses on the temporal cyclic effect on the patterns of users' behaviors.

Definition 6 ***(User Activity)*** *A user activity in this chapter is extended to a six tuple $(u,\ v,\ l_v,\ W_v,$ $t,\ s)$*

The user u visits the spatial item v, located at l_v and described as W_v, at time t, in the role of s. If $s = 0$, the user is recognized as a local and the activity occurs in u's home town. If $s = 1$, the user u plays the role of tourist when visiting v.

Given a dataset D as the union of a collection of user profiles, we aim to provide spatial item recommendation for both home-town and out-of-town users. We formulate our problem in this chapter taking into account the two scenarios in a unified fashion as follows.

Problem 2 *Given a user activity dataset D, a target user u with his/her current location l and the querying time t (i.e., the query is $q = (u,\ l,\ t)$), our goal is to recommend a list of spatial items that u would be interested in. Given a distance threshold d, the problem becomes an* ***out-of-town recommendation*** *if the distance between the target user's current location and his/her home location (e.g., $|l - l_u|$) is greater than d. Otherwise, the problem is a* ***home-town recommendation***.

Following related studies [18, 53], we set $d = 100km$ in our work, since a distance around $100km$ is the typical radius of human "reach" $-$ it takes 1 to 2 hours to drive such a distance.

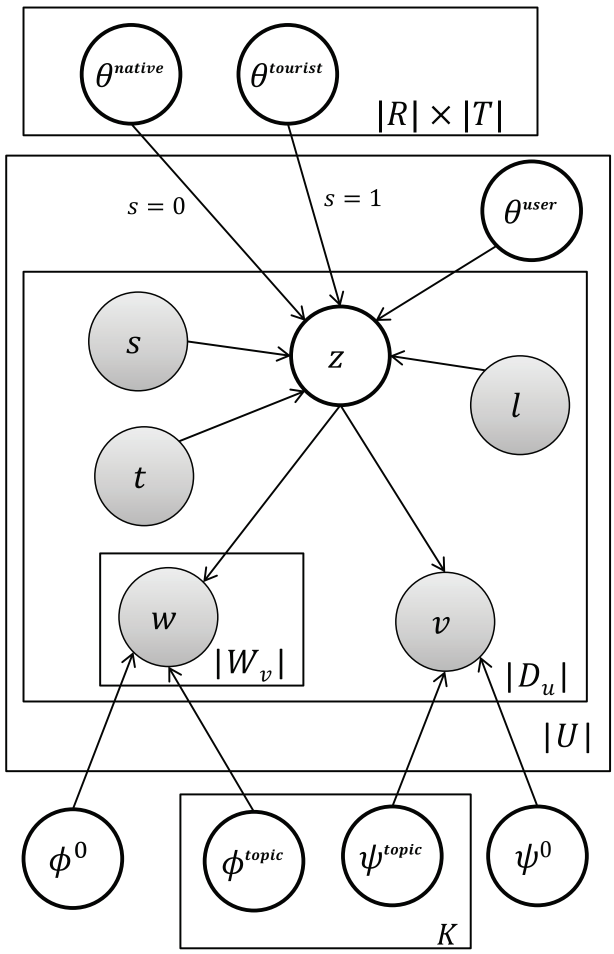


FIGURE 3.1: The Graphical Representation of Our Model

3.3.2 Model Description

To model user activities, we propose the spatial temporal sparse additive generative model (ST-SAGE). Figure 3.1 shows the graphical representation of ST-SAGE. We first introduce the notations of our model, which are listed in Table 3.1. Our input data, which are users' activity profiles, are modeled as observed random variables, shown as shaded circles in Figure 3.1. Similar to existing models [30, 81], the topic index of each user activity is considered as a latent random variable, which is denoted as z.

User Interest Modeling. Intuitively, a user chooses a spatial item at a given location on a specific time by matching his/her personal interests with the content of that item. Inspired by the early work on user interest modeling [30, 44, 81], ST-SAGE also adopts latent topics to characterize users' interests. Specifically, we infer an individual user's interest vector over a set of topics according to his/her visited spatial items and their associated contents, denoted as $\boldsymbol{\theta}_u^{user}$. Thus, our model alleviates the travel locality for out-of-town recommendation, as the content of the spatial items plays the role of medium through which user interests inferred from their home town can be transferred to out-of-town regions. Besides, we also introduce a background vector over topics $\boldsymbol{\theta}^0$ to capture common topics among all users. The purpose of using a background model $\boldsymbol{\theta}^0$ is to make the user interests $\boldsymbol{\theta}_u^{user}$ learned from the dataset more discriminative.

Spatial Dynamics Modeling. To adapt to spatial dynamics of user behavior, we exploit the preference of the crowds who share the same role with the target user u. For example, the preference of the tourists will be leveraged if the target user is currently out-of-town. Technically, we introduce two parameters: *native preference* and *tourist preference*. Given a location l, the native preference represents the preference of people living at location l, denoted as $\boldsymbol{\theta}_l^{native}$. In contrast, the tourist preference represents the preference of tourists travelling in location l, denoted as $\boldsymbol{\theta}_l^{tourist}$. Note that, distinguishing native preference from tourist preference is one of the fundamental differences between our model and the LCA-LDA model [81, 80] which also exploits local activity records at the target location.

Temporal Dynamics Modeling. As users tend to visit different spatial items at different time, our ST-SAGE model is expected to capture this temporal dynamic. To model the temporal influence, an intuitive approach is to split time into time slices at the predefined granularity (e.g., hourly or seasonal) and then model the temporal preferences of a user based on his/her visited spatial items during a time slice. However, splitting a user's activity data into multiple slices will make the data much sparser in a specific time slice, which inevitably makes the inference of personal temporal preferences overfitting. In light of this, we propose to exploit and integrate the collective temporal preferences of the crowd with the same role instead of personal temporal preferences, as shown in Figure 3.1. The visiting time information t shown as a shaded circle. The collective preferences of the public with the same roles $\boldsymbol{\theta}_l^{native}$ and $\boldsymbol{\theta}_l^{tourist}$ are extended to time-aware vectors $\boldsymbol{\theta}_{l,t}^{native}$ and $\boldsymbol{\theta}_{l,t}^{tourist}$.

Topics Modeling. To take full advantage of the strengths of both content-based and collaborative filtering-based recommendation methods, a topic z in our ST-SAGE model is not only associated with a word vector $\boldsymbol{\phi}_z^{topic}$, but also with a vector over spatial items $\boldsymbol{\psi}_z^{topic}$. This design enables $\boldsymbol{\phi}_z^{topic}$ and $\boldsymbol{\psi}_z^{topic}$ to be mutually influenced and enhanced during the topic discovery process by associating them. Thus, the discovered topic z, on one hand, can cluster the content-similar items together. On the other hand, it can also capture the item co-occurrence patterns to link relevant items together, similar to item-based collaborative filtering methods. We also introduce two background models for words and items, respectively: $\boldsymbol{\phi}^0$ and $\boldsymbol{\psi}^0$. The purpose of using background models is to make the topics learned from the dataset more discriminative, since $\boldsymbol{\phi}^0$ and $\boldsymbol{\psi}^0$ assign high probabilities to

TABLE 3.1: Notations of Model Parameters

Variable	Interpretation
K	the number of topics
$z_{u,i}$	the topic assigned to spatial item $v_{u,i}$
$\boldsymbol{\theta}^0$	the topic vector of the background
$\boldsymbol{\theta}_u^{user}$	the topic vector, representing the intrinsic interest of user u
$\boldsymbol{\theta}_{l,t}^{native}$	the topic vector of l on time t, representing the native preference at l on time t
$\boldsymbol{\theta}_{l,t}^{tourist}$	the topic vector of l on time t, representing the tourist preference at l on time t
$\boldsymbol{\phi}^0$	the word vector of the background
$\boldsymbol{\psi}^0$	the spatial item vector of the background
$\boldsymbol{\phi}_z^{topic}$	content word vector of topic z
$\boldsymbol{\psi}_z^{topic}$	spatial item vector of topic z

non-discriminative and non-informative words and items.

The generative process of the ST-SAGE model for an individual user activity in the user profile D_u is as follows.

- Draw a topic index $z_{u,i}$

  $z_{u,i} \sim P(z_{u,i}|s_{u,i}, l_{u,i}, t_{u,i}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist})$

- For each content word $w_{u,i,n}$ in $W_{u,i}$, draw

  $w_{u,i,n} \sim P(w_{u,i,n}|\boldsymbol{\phi}^0, z_{u,i}, \boldsymbol{\phi}^{topic})$

- Draw a spatial item $v_{u,i}$

  $v_{u,i} \sim P(v_{u,i}|\boldsymbol{\psi}^0, z_{u,i}, \boldsymbol{\psi}^{topic})$

For each user activity, ST-SAGE first chooses the topic this activity is about. To generate the topic index z, we utilize a multinomial model as expressed in Equation 3.3.

$$\begin{aligned} & P(z_{u,i}|s_{u,i}, l_{u,i}, t_{u,i}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist}) \\ =& P(z_{u,i}|\boldsymbol{\theta}_u^{user} + (1 - s_{u,i}) \times \boldsymbol{\theta}_{l_{u,i},t_{u,i}}^{native} + s_{u,i} \times \boldsymbol{\theta}_{l_{u,i},t_{u,i}}^{tourist}) \end{aligned} \tag{3.3}$$

where $P(z|\boldsymbol{\theta}_u^{user} + (1-s) \times \boldsymbol{\theta}_{l,t}^{native} + s \times \boldsymbol{\theta}_{l,t}^{tourist})$, denoted as $\alpha_{u,s,l,t,z}$, is computed as in Equation 3.6. Once the topic z is generated, the spatial item v and the associated content words are generated as expressed in Equations 3.4 and 3.5, respectively.

$$P(v_{u,i}|\boldsymbol{\psi}^0, z_{u,i}, \boldsymbol{\psi}^{topic}) = P(v_{u,i}|\boldsymbol{\psi}^0 + \boldsymbol{\psi}_{z_{u,i}}^{topic}) \tag{3.4}$$

$$P(w_{u,i,n}|\boldsymbol{\phi}^0, z_{u,i}, \boldsymbol{\phi}^{topic}) = P(w_{u,i,n}|\boldsymbol{\phi}^0 + \boldsymbol{\phi}_{z_{u,i}}^{topic}) \tag{3.5}$$

where $P(v|\boldsymbol{\psi}^0 + \boldsymbol{\psi}_z^{topic}) = \gamma_{z,v}$ and $P(w|\boldsymbol{\phi}^0 + \boldsymbol{\phi}_z^{topic}) = \beta_{z,w}$ are computed as in Equation 3.6. Note that in order to model topics based on the background word/item vectors, for each topic, ST-SAGE models the difference from the background word/item vector in log-frequencies, instead of the frequencies themselves.

3.3.3 Model Inference

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables **w**, **v** and **s**, and the marginalization is performed with respect to the latent random variable **z**. However, it is difficult to be maximized directly. Therefore, we apply a mixture of EM and a Monte Carlo sampler, called the Gibbs EM algorithm [70], to maximize the complete data likelihood in Equation 3.7, where $\circleddash$ is the set of all the parameters. In the E-step, we sample latent topic assignments by fixing all other parameters using Gibbs sampling. In the M-step, we optimize model parameters $\circleddash$ by fixing all topic assignments. The two steps are iterated until convergence.

$$\begin{aligned}
\alpha_{u,s,l,t,z} &= \frac{exp(\boldsymbol{\theta}_z^0 + \boldsymbol{\theta}_{u,z}^{user} + (1-s) \times \boldsymbol{\theta}_{l,t,z}^{native} + s \times \boldsymbol{\theta}_{l,t,z}^{tourist})}{\sum_{zz} exp(\boldsymbol{\theta}_{zz}^0 + \boldsymbol{\theta}_{u,zz}^{user} + (1-s) \times \boldsymbol{\theta}_{l,t,zz}^{native} + s \times \boldsymbol{\theta}_{l,t,zz}^{tourist})}, \\
\beta_{z,w} &= \frac{exp(\boldsymbol{\phi}_w^0 + \boldsymbol{\phi}_{z,w}^{topic})}{\sum_{ww} exp(\boldsymbol{\phi}_{ww}^0 + \boldsymbol{\phi}_{z,ww}^{topic})}, \\
\gamma_{z,v} &= \frac{exp(\boldsymbol{\psi}_z^0 + \boldsymbol{\psi}_{z,v}^{topic})}{\sum_{vv} exp(\boldsymbol{\psi}_z^0 + \boldsymbol{\psi}_{z,vv}^{topic})}
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
P(\mathbf{z},\ \mathbf{w},\ \mathbf{v}|\circleddash,\ \mathbf{s},\ \mathbf{u},\ \mathbf{l},\ \mathbf{t}) &= P(\mathbf{z}|\mathbf{s},\ \mathbf{u},\ \mathbf{l},\ \mathbf{t},\ \boldsymbol{\theta}^0,\ \boldsymbol{\theta}^{user},\ \boldsymbol{\theta}^{native},\ \boldsymbol{\theta}^{tourist})P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic})P(\mathbf{v}|\mathbf{z}, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{topic}) \\
&= \prod_{u=1}^{|U|}\prod_{i=1}^{|D_u|} \alpha_{u,s_{u,i},l_{u,i},t_{u,i},z_{u,i}} \prod_{u=1}^{|U|}\prod_{i=1}^{|D_u|}\prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{u,i,n}} \prod_{u=1}^{|U|}\prod_{i=1}^{|D_u|} \gamma_{z_{u,i},v_{u,i}}
\end{aligned} \tag{3.7}$$

More specifically, we iteratively draw latent topic **z** for all activities in the E-step. According to the Gibbs Sampling, when sampling $z_{u,i}$ as expressed in Equation 3.8, we assume all other variables are fixed. $z_{\neg u,i}$ represents the topic assignments for all user activities except the i'th activity for user u.

$$P(z_{u,i}|z_{\neg u,i},\ \mathbf{w},\ \mathbf{v},\ \mathbf{s},\ \mathbf{u},\ \mathbf{l},\ \mathbf{t},\ \ominus) \propto \alpha_{u,s_{u,i},l_{u,i},t_{u,i},z_{u,i}} \times \prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{u,i,n}} \times \gamma_{z_{u,i},v_{u,i}} \tag{3.8}$$

In the M-step, we optimize the parameters $\ominus$ to maximize the log likelihood of the objective function with all topic assignments fixed. To update the parameters, we use the gradient descent learning algorithm PSSG (Projected Scaled Sub-Gradient) [62], which is designed to solve optimization problems with L1 regularization on parameters. More importantly, PSSG is scalable because it uses the quasi-Newton strategy with line search that is robust to common functions. Let L be the log-likelihood of the model. According to the limited-memory BFGS (Broyden Fletcher Goldfarb Shanno) [46] updates for the quasi-Newton method, the gradients of model parameters $\boldsymbol{\theta}^0$, $\boldsymbol{\theta}^{user}$, $\boldsymbol{\theta}^{native}$ and $\boldsymbol{\theta}^{tourist}$ are provided as follows.

$$\frac{\partial L}{\partial \boldsymbol{\theta}^0_z} = d(z) - \sum_{u=1}^{|U|}\sum_{i=1}^{|D_u|} \alpha_{u,s_{u,i},l_{u,i},t_{u,i},z} \tag{3.9}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}^{user}_{u,z}} = d(u,z) - \sum_{i=1}^{|D_u|} \alpha_{u,s_{u,i},l_{u,i},t_{u,i},z} \tag{3.10}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}^{native}_{l,t,z}} = (1-s) \times (d(l,t,z) - \sum_{j=1}^{|D(l,t)|} \alpha_{u_j,s_j,l,t,z}) \tag{3.11}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}^{tourist}_{l,t,z}} = s \times (d(l,t,z) - \sum_{j=1}^{|D(l,t)|} \alpha_{u_j,s_j,l,t,z}) \tag{3.12}$$

where $d(z)$ is the number of activities assigned to topic z, $d(u,z)$ represents how many activities are assigned to topic z in D_u, $d(l,t,z)$ denotes the number of activities assigned to topic z at location l on time t, $D(l,t)$ is the set of activities occurring at location l on time t and u_j denotes the user who generates the j-th activity record.

Similarly, the gradients of model parameters $\boldsymbol{\phi}^0$, $\boldsymbol{\phi}^{topic}$, $\boldsymbol{\psi}^0$, and $\boldsymbol{\psi}^{topic}$ are computed as follows:

$$\frac{\partial L}{\partial \boldsymbol{\phi}^0_w} = d(w) - \sum_{z=1}^{K} d(z) \times \beta_{z,w} \tag{3.13}$$

$$\frac{\partial L}{\partial \boldsymbol{\phi}^{topic}_{z,w}} = d(z,w) - d(z) \times \beta_{z,w} \tag{3.14}$$

$$\frac{\partial L}{\partial \boldsymbol{\psi}^0_v} = d(v) - \sum_{z=1}^{K} d(z) \times \gamma_{z,v} \tag{3.15}$$

ALGORITHM 1: The Algorithm of Model Inference

Input: A user profile dataset D;

Output: estimated parameters $\boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist}, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{topic}$;

1 Initialize all the parameters and the latent topics according to the "Initialization Strategy";

2 **while** *the model is not convergent* **do**

3   **for** *each* D_u *in* D **do**

4   **for** *each record* (u, v, l_v, W_v, s) **do**

5   Draw a latent topic z with Gibbs Sampling according to Equation 3.8;

6   **end**

7   **end**

8   Update $\boldsymbol{\theta}^0$, $\boldsymbol{\theta}^{user}$, $\boldsymbol{\theta}^{native}$, $\boldsymbol{\theta}^{tourist}$, $\boldsymbol{\phi}^0$, $\boldsymbol{\phi}^{topic}$, $\boldsymbol{\psi}^0$ and $\boldsymbol{\psi}^{topic}$ with BFGS method as expressed in Equation 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16 respectively;

9 **end**

10 Return $\boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist}, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{topic}$;

$$\frac{\partial L}{\partial \boldsymbol{\psi}_{z,v}^{topic}} = d(z, v) - d(z) \times \gamma_{z,v} \tag{3.16}$$

where $d(w)$ is the number of activities in which the word w appears, and $d(z, w)$ is the number of activities where the word w is assigned to the topic z. $d(v)$ is the number of activities associated with item v, and $d(z, v)$ represents the number of activities in which topic z is assigned to item v.

Initialization Strategy: The last but not least important technique is how to initialize the model. Different initialization values of parameters can lead to significantly different results. Here, we use the following initialization steps. Taking the initialization of $\boldsymbol{\phi}^0$ and $\boldsymbol{\phi}^{topic}$ as an example, we initialize $\boldsymbol{\phi}^0$ as log frequencies of words in the whole corpus and $\boldsymbol{\phi}^{topic}$ as log frequencies of words assigned to topic z minus the same word in $\boldsymbol{\phi}^0$. Similar strategy can be applied to $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ values. For latent topics, we initialize them randomly.

Algorithm 1 describes the process of the inference of ST-SAGE. First, we initialize all the parameters according to the aforementioned "Initialization Strategy"(Line 1). Then, before the objective function in Equation 3.7 becomes convergent, we first sample a latent topic for each activity record (Lines 3-7) and then update all the parameters according to the BFGS method (Lines 8). When the objective function is convergent, the learned parameters are returned as our expected outputs (Line 10).

Time Complexity. There are two steps in the Gibbs EM algorithm: Gibbs sampling and gradient descent learning. We assume that the algorithm needs I iterations to reach convergence. For each iteration, its time complexity is analyzed as follows. In the E-step, it needs to go through all user check-in records, and for each check-in it requires $O(K)$ operations to compute the posterior probability distribution for sampling topic z. Thus, the time complexity in this step is $O(K \times D)$ where D=$\sum_u |D_u|$ is the total number of check-ins in the dataset. In the M-step, we use the gradient descent learning algorithm to update the model parameters $\ominus = \{\boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist}, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}, \boldsymbol{\varphi}^0, \boldsymbol{\varphi}^{topic}\}$ based on the topic assignments sampled in the E-step. We assume that the gradient descent algorithm needs J iterations to converge, and in each iteration the time complexity to compute the gradients for the model parameters $\ominus$ is $O(K \times D)$. Thus, the time complexity in the M-step is $O(J \times K \times D)$ where J is generally less than 50. The total time complexity for our model inference is $O(I \times (J+1) \times K \times D)$=$O(I \times J \times K \times D)$, which indicates that the computational time is linear with respect to the number of check-ins in our dataset.

3.3.4 Spatial Smoothing

To combat data sparsity when modeling the *temporal native preference* and *temporal tourist preference*, we adopt a quad tree structure which is proposed in [19], called spatial pyramid, to partition and index the entire geographic area. The spatial pyramid is proposed in [39, 3], which is constructed by partitioning item locations into spatial regions of varying sizes at different hierarchies. More specifically, the spatial pyramid decomposes the space into H levels. Level 0 has only one grid cell. For a given level h, the space is partitioned into 4^h grid cells of equal area. Thus, the space can be divided recursively into numerous cells at different levels with different granularity. The graphical representation of the spatial pyramid is shown in Figure 3.2.

One of the fundamental assumptions in spatial data mining is that everything is related to everything else but nearby things are more related than distant things, which is proposed in [20] as the first law of geography [63]. This law is known as the "spatial autocorrelation". The spatial pyramid structure can encode this law in an effective manner. That is, for each location l, it can be represented by a path from the root node to its corresponding leaf node. We use a vector to describe the path, (l_1, l_2,..., l_h, ..., l_H), where l_h is a grid at level h that contains the location l. Based on the vector representation of a location, we can easily compute the proximity between two locations. For example, if two locations in the spatial pyramid share more ancestors, then these two locations are more proximate.

When user activity data at a location l and time t is very sparse, both the temporal native preference

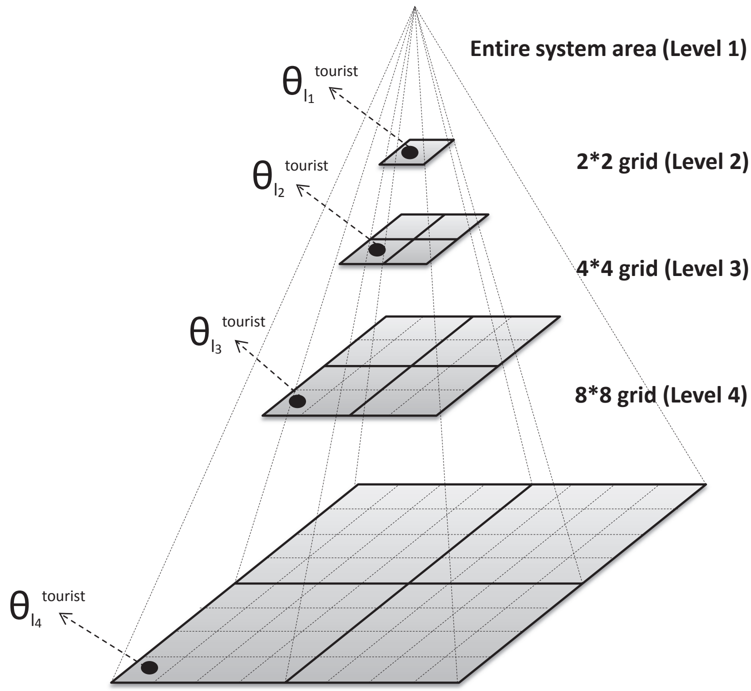


FIGURE 3.2: The Spatial Pyramid

$\boldsymbol{\theta}_{l,t}^{native}$ and the temporal tourist preference $\boldsymbol{\theta}_{l,t}^{tourist}$ may not be estimated accurately. To address this issue, we exploit geographical correlation to enhance the prior knowledge about the model parameters $\boldsymbol{\theta}_{l,t}^{native}$ and $\boldsymbol{\theta}_{l,t}^{tourist}$. Intuitively, if two locations l and l' are proximate in the geographical space, their temporal local preferences $\boldsymbol{\theta}_{l,t}^{native}$ and $\boldsymbol{\theta}_{l',t}^{native}$ should be similar to each other. The intuition can be applied to the temporal tourist preferences $\boldsymbol{\theta}_{l,t}^{tourist}$ and $\boldsymbol{\theta}_{l',t}^{tourist}$ similarly. To integrate the information of geographical correlation into our ST-SAGE model, we apply the *additive* framework [2, 33] to compute temporal native preference $\boldsymbol{\theta}_{l,t}^{native}$ and tourist preference $\boldsymbol{\theta}_{l,t}^{tourist}$ at location l and time t based on the path vector representation. Specifically, given a location l, its temporal native preference and temporal tourist preference at time t are represented as follows.

$$\boldsymbol{\theta}_{l,t}^{native} = \sum_{h=1}^{H} \boldsymbol{\theta}_{l_h,t}^{native}, \quad \boldsymbol{\theta}_{l,t}^{tourist} = \sum_{h=1}^{H} \boldsymbol{\theta}_{l_h,t}^{tourist} \tag{3.17}$$

According to the above equations, both the temporal native preference and the temporal tourist preference of a location depend on all of its ancestors up to the root. This representation method enables neighboring locations to share similar preferences as desired, i.e., the preferences are smoothed over the spatial pyramid. Meanwhile, if there are few or no activities at a location, we can still infer its preference guided by its ancestors. Besides, once *temporal native preference* and *temporal tourist preference* for each level are learned, this modeling makes the switch between various granularity fast and convenient by changing the lowest level in the model without re-training the parameters.

3.3.5 Application Scenario

Figure 3.3 demonstrates an application scenario of ST-SAGE where the input consists of four elements: the current location, the current time, the number of the recommendations, and the size of the target region (e.g., zoom in/out on Google Map [3]). The location and time are captured automatically by the functions of the mobile phones.

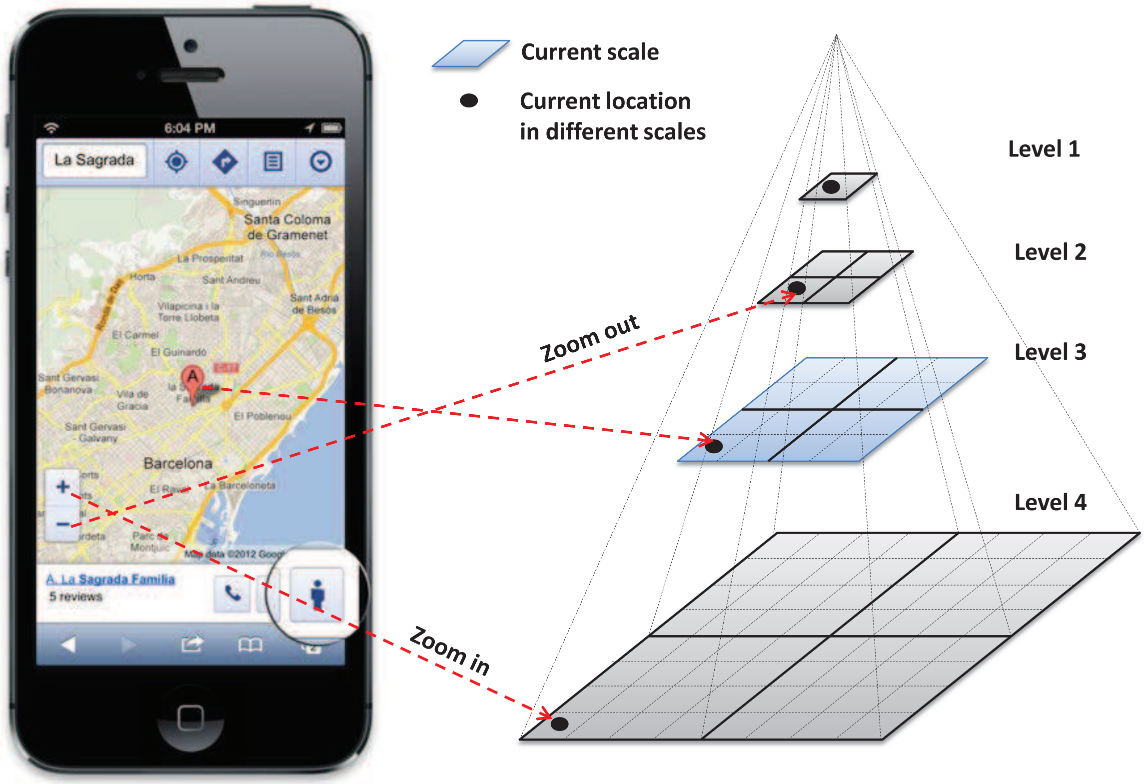


FIGURE 3.3: The Application Scenario

By indexing the geographical space using a spatial tree structure, our model supports users to change the scale of the target region conveniently by switching between different levels in the tree structure. Although the previous work of [80] also supports the change to granularity , it has to re-train all parameters for the new granularity while our model does not. Take the application in Figure 3.3 as an example. The current scale corresponds to "Level 3" of the spatial pyramid. If the querying user zooms out on the map, the map becomes smaller and the granularity of the regions becomes coarser. ST-SAGE will automatically change the scale from "Level 3" to "Level 2". As the model parameters for all levels have been learned, it is not necessary to re-train the parameters for this "zooming out". Similarly, if the user zooms in on the map, ST-SAGE simply switches the scale from the upper level into the lower level without re-training.

3.3.6 Parallel Implementation

We implement a parallel ST-SAGE inference algorithm on the distributed GraphLab framework [50] and parallel gradient descent learning framework PSSG (Projected Scaled Sub-Gradien) [62]. PSSG is scalable because it not only uses the quasi-Newton strategy with line search that is robust to common functions, but also adopts the multi-core parallel processing strategy. There are two steps in the Gibbs EM algorithm: Gibbs sampling and gradient descent learning. We decompose the inference procedure of ST-SAGE into a two-step parallel processing. In the E step, we implement the Gibbs sampling algorithm in the GraphLab framework [50]. In the M step, PSSG [62] is adopted as the parallel gradient descent learning framework in ST-SAGE.

GraphLab framework is proposed to support asynchronous, dynamic, graph-parallel computation while ensuring data consistency, which is required by Gibbs Sampling to ensure statistical correctness, and achieving a high degree of parallel performance in the shared-memory setting. GraphLab framework has demonstrated superior performance over popular parallel systems, e.g., MapReduce and Spark, for many machine learning algorithms [32].

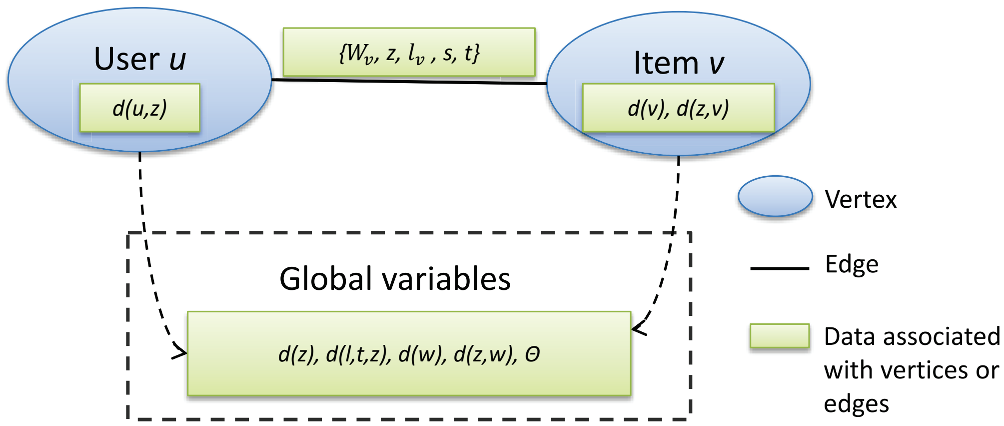


FIGURE 3.4: Data Graph

This framework implements the $gather - apply - scatter(GAS)$ model which abstracts the program into three phases. In the $gather$ phase, each vertex aggregates data from the $scope$ of the vertex. The $scope$ of a vertex ve includes the data stored in ve as well as the data stored in all adjacent vertices and adjacent edges. The gather result is used to update the data stored in this vertex in the $apply$ phase. Lastly, each vertex triggers its neighboring vertices or modifies adjacent edge data.

GraphLab framework stores the algorithm state as an undirected graph called the **data graph**. In this graph, one can associate arbitrary data with each vertex and edge. In ST-SAGE, we store the state as a graph in Figure 3.4. Specifically, we construct an undirected graph that connects each user with each spatial item. An edge between a user u and a spatial item v contains the location l, the role of u at l (a tourist or a local) denoted as s, the time t, content words of item v and the topic indicator z. For each user u, the number of her check-ins assigned to each topic z, denoted as $d(u,z)$, is stored in her associated vertex. Similarly, for each spatial item v, the number of its associated check-ins (i.e., $d(v)$) and the number of its associated check-ins assigned to each topic z (i.e., $d(z,v)$) are stored in its corresponding vertex.

We accelerate the Gibbs EM algorithm by simultaneously sampling new topics according to Equation 3.8. The variables in Equation 3.8 are either maintained globally or in vertices. To improve efficiency further, the global variables are periodically aggregated from the vertices, while the variables in vertices are updated during the $gather$ and $apply$ phases. New topics are sampled in the $scatter$ phase. As the variables in the Gibbs EM algorithm are updated with the BFGS algorithm, we run the BFGS algorithm during the $apply$ phase where $gather\_result$ is applied to update the related variables. Algorithm 2 shows the GAS procedures of the ST-SAGE Gibbs Sampler.

ALGORITHM 2: GAS program of ST-SAGE Gibbs Sampler

Gather (v, e);

if $v.type = user$ **then**

  return α related parameters and $d(z), d(u,z), d(l,t,z)$;

end

if $v.type = item$ **then**

  return β, γ related parameters and $d(w), d(z,w), d(v), d(z,v), d(z)$;

end

Apply $(v, gather_result)$;

if $v.type = user$ **then**

  call BFGS to update α related parameters with Equation 3.9, 3.10, 3.11 and 3.12 according to $gather\_result$;

end

if $v.type = location$ **then**

  call BFGS to update β, γ related parameters with Equation 3.13, 3.14, 3.15 and 3.16 according to $gather\_result$;

end

Scatter (v, e);

sample z according to Equation 3.8;

With this two-step parallel processing, our inference implementation leads to satisfying efficiency and scalability on large data, as shown in the empirical study.

3.4 Spatial Item Recommendation

The ranking scores of spatial items are computed according to $P(v, W_v|u, l, s, t, \ominus)$, as shown in Equation 3.18.

$$\begin{aligned} P(v, W_v|u, l, s, t, \ominus) = & \sum_{z=1}^{K} P(v, W_v, z|u, l, s, t, \ominus) \\ = & \sum_{z=1}^{K} P(z|u, s, t, l, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist}) \times P(W_v|z, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}) \times P(v|z, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{topic}) \\ = & \sum_{z=1}^{K} \alpha_{u,s,t,l,z} \times \left(\prod_{n=1}^{|W_v|} \beta_{z,w_{v,n}} \right)^{\frac{1}{|W_v|}} \times \gamma_{z,v} \end{aligned} \tag{3.18}$$

where W_v denotes the content words describing item v. We adopt the geometric mean for the probability of topic z generating the word set W_v, considering that the number of words associated with different spatial items may be different.

To accelerate the online recommendation process, we propose a ranking framework in Equation 3.19 which separates the offline computation from the online computation to the maximum extent.

$$\begin{gathered} S(q, v) = \sum_{z=1}^{K} F(z, v) W(q, z) \\ F(z, v) = (\prod_{n=1}^{|W_v|} \beta_{z,w_{v,n}})^{\frac{1}{|W_v|}} \times \gamma_{z,v}, \quad W(q, z) = \alpha_{u,l,s,t,z} \end{gathered} \tag{3.19}$$

In Equation 3.19, $F(z, v)$ represents the offline score which denotes the score of spatial items v with respect to topic z. This part is computed offline since it is independent from the query $q = (u, l, t)$. On the other hand, $W(q, z)$ is inferred online, denoting the preference of query q on topic z. Note that the principal time-consuming components of $W(q, z)$ are also computed offline (e.g., $\boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}$ and $\boldsymbol{\theta}^{tourist}$). This design enables the maximum separation of the online computation from the offline computation and in turn reduces the query time.

When a query $q = (u, l, t)$ arrives, we first compute the query preference weight on each topic (i.e., $W(q, z)$), and then aggregate $F(z, v)$ over each topic with the weight $W(q, z)$ for each spatial item. At last, k spatial items with the highest scores are returned as the query result. ST-SAGE is trained offline, while the recommendation performed online is a process of combining the various factors. This scheme guarantees quick and dynamic computational responses in terms of computing spatial item rankings in real time.

3.5 Experiments

In this section, we first describe the settings of experiments including the datasets, comparative approaches and the evaluation method. We then demonstrate the experimental results in terms of both the recommendation effectiveness and efficiency of ST-SAGE which include the efficiency of training.

3.5.1 Experimental Settings

3.5.1.1 Datasets

We perform experiments on two real large-scale LBSN datasets (i.e. Foursquare and Twitter) and one large scale synthetic dataset with 10 million check-ins. The detailed information of the two real-life datasets are as follows.

Foursquare. This dataset contains the check-in history on 111,813 spatial items of 4,163 users who live in California, USA from 12/2009 to 07/2013. It contains the social networks, home location, check-in venue identifiers, location of each venue in terms of latitude and longitude, and the content of each check-in venue for each user. The total number of check-ins in this dataset is 483,813.

Twitter. This dataset is based on the publicly available Twitter dataset [12]. Twitter supports third-party location sharing services like Foursquare and Gowalla, where users of these services opt-in to share their check-ins on Twitter. But the original dataset does not contain the category or tag information about venues. So, we crawled the category and tag information associated with each venue from Foursquare with the help of its publicly available API [2]. The enhanced dataset contains 114,058 users, 62,547 spatial items and 1434,668 check-in activities from 09/2010 to 01/2011. Each check-in record has the same format as the above Foursquare dataset. However, this dataset does not contain user social network information. As the home locations of users in this dataset are not explicitly given, we adopt the method in [61] by inferring the user's home location as the cell in the spatial pyramid with the most check-ins.

To make the experiments repeatable, we make the datasets and our code publicly available [3].

[2]https://developer.foursquare.com/

[3]http://net.pku.edu.cn/daim/yinhongzhi/index.html

3.5.1.2 Comparative Approaches

We compare our ST-SAGE model with the following five methods representing state-of-the-art spatial item recommendation techniques.

JIM. JIM [82] is a joint probabilistic generative model which integrates the temporal effect, geographical-social influence, content effect and word-of-mouth effect. The word-of-mouth effect refers to that the probability of a user visiting a spatial item is largely affected by the popularity of this spatial item at the target region.

UTE+SE. UTE+SE, proposed in [84], is a collaborative recommendation model which incorporates both the temporal and spatial information. Given a user and specific time, it first finds the users sharing similar temporal preference with her and then produce the time-specific recommendations based on their historical check-ins made around the time.

LCA-LDA. LCA-LDA is a location-content-aware recommender model which is developed to support spatial item recommendation for users traveling in new cities [81, 80]. This model takes into account both personal interests and local preferences of each city by exploiting both item co-visiting patterns and content of spatial items. Compared with ST-SAGE, LCA-LDA is a traditional mixture model which introduces “switching” variables to consider multiple factors. Besides, LCA-LDA ignores the roles of users and does not distinguish between *tourist preference* and *native preference*.

UPS-CF. UPS-CF, proposed in [18], is a collaborative recommendation framework which is especially designed for out-of-town users. This framework integrates user-based collaborative filtering and social-based collaborative filtering. That is, it recommends spatial items to a target user according to the activity records of both his/her friends and similar users.

CKNN. CKNN [3] projects a user’s activity history into the category space and models user preference using a weighted category hierarchy. When receiving a query, CKNN retrieves all users and items located in the querying location, formulates a user-item matrix online, and then applies a user-based CF method to predict the rating of a querying user on an unvisited item. Note that the similarity between two users in CKNN is computed according to their weights in the category hierarchy, making CKNN a hybrid recommendation method.

To further validate the benefits brought by exploiting temporal information and spatial smoothing based on the spatial pyramid, we compare our model with three other simplified version.

ST-SAGE-S1 is the simplified version of the ST-SAGE model which does not consider the temporal influence. This means that $\boldsymbol{\theta}^{native}$ and $\boldsymbol{\theta}^{tourist}$ are $|R|$ instead of $|R| \times |T|$ vectors. For each user activity, we sample the topic index z according to the following Equation instead of Equation

3.3. This simplified version is equal to the Geo-SAGE model proposed in [72].

$$\begin{aligned}&P(z_{u,i}|s_{u,i}, l_{u,i}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist})\\ =&P(z_{u,i}|\boldsymbol{\theta}_u^{user} + (1 - s_{u,i}) \times \boldsymbol{\theta}_{l_{u,i}}^{native} + s_{u,i} \times \boldsymbol{\theta}_{l_{u,i}}^{tourist})\end{aligned}$$

ST-SAGE-S2 is the simplified version which models the temporal information individually without smoothing. Specifically, we sample the topic index z according to the following Equation:

$$\begin{aligned}&P(z_{u,i}|s_{u,i}, l_{u,i}, t_{u,i}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{native}, \boldsymbol{\theta}^{tourist})\\ =&P(z_{u,i}|\boldsymbol{\theta}_{u,t_{u,i}}^{user} + (1 - s_{u,i}) \times \boldsymbol{\theta}_{l_{u,i}}^{native} + s_{u,i} \times \boldsymbol{\theta}_{l_{u,i}}^{tourist})\end{aligned}$$

ST-SAGE-S3 is the simplified version of ST-SAGE which does not exploit the geographical correlation in the spatial pyramid. Thus, the inferred temporal native preference and temporal tourist preference for location l at time t are not reliable when there are few or even no user activity records.

3.5.1.3 Evaluation Methods

We evaluate both the effectiveness and efficiency of ST-SAGE. For the efficiency part, we evaluate both the training efficiency and the online recommendation efficiency. At last, we also demonstrate the trade-off between the effectiveness and efficiency.

Recommendation Effectiveness. Since our ST-SAGE model is designed for both home-town recommendation and out-of-town recommendation, we evaluate the recommendation effectiveness of our model under each of the scenarios. Given a user profile in terms of a collection of user activities, we divide the user's activities into a training set and a test set. For the home-town recommendation scenario, we randomly select 30% of the activity records occurring at the user's home town as the test set, and use the remaining activity records as the training set. Similarly, for the scenario of out-of-town recommendation, we randomly select 30% of the activity records generated by the user when he/she travels out of town as the test set, and use the remaining activity records as the training set. To decide whether an activity record occurs when the user is in his/her home town or elsewhere, we measure the location distance between the user's home town and the spatial item (e.g., $|l_u - l_v|$). If the distance is greater than $100km$, we assume the activity occurs when the user is out-of-town. The threshold $d = 100km$ is selected because a distance around $100km$ is the typical human radius of "reach", which takes 1 to 2 hours to drive.

According to the above dividing strategies, we split the user activity dataset D into the training set D_{train} and the test set D_{test}. To evaluate the recommendation methods, we adopt the evaluation methodology and measurement Accuracy@k, which is applied in [30, 7, 14, 35, 81]. Specifically, for each user activity record $(u,\ v,\ l_v,\ W_v, s, t)$ in D_{test}: 1) we compute the ranking score for spatial item v and all other spatial items which are within the circle of radius d centered at l_v and unvisited by u previously; 2) we form a ranked list by ordering all of these spatial items according to their ranking scores. Let r denote the position of the spatial item v within this list. The best result corresponds to the case where v precedes all the unvisited spatial items (that is, $r = 1$); and 3) we form a top-k recommendation list by picking the k top ranked spatial items from the list. If $r \leq k$, we have a hit (i.e., the ground truth item v is recommended to the user). Otherwise, we have a miss.

The computation of Accuracy@k proceeds as follows. We define hit@k for a single test case as either the value 1, if the test item v appears in the top-k results, or the value 0, if otherwise. The overall Accuracy@k is defined by averaging over all test cases:

$$Accuracy@k = \frac{\#hit@k}{|D_{test}|}$$

where $\#hit@k$ denotes the number of hits in the test set, and $|D_{test}|$ is the number of all test cases.

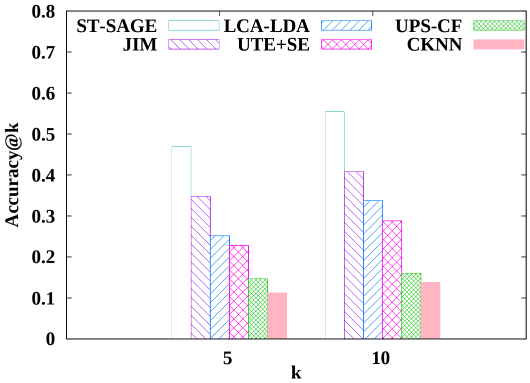
Training Efficiency. We evaluate the efficiency of the model training. The efficiency of the training of ST-SAGE is mainly affected by the number of activity records in the dataset and the number of nodes in the GraphLab framework. Therefore, we evaluate the model training efficiency of ST-SAGE over the dataset with various numbers of nodes.

3.5.2 Recommendation Effectiveness

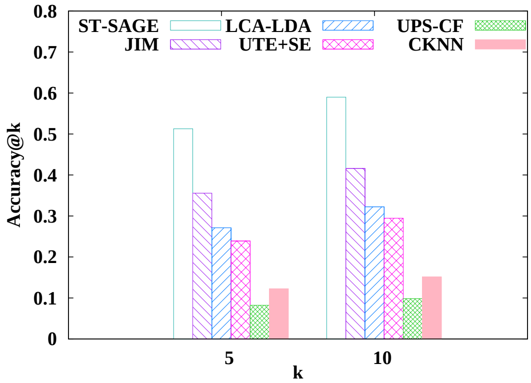
In this part, we first present the experimental results of the comparison between the other state-of-art recommendation methods on two real-life datasets for both out-of-town recommendation and hometown recommendation. Second, we validate the benefits brought by different factors such as native preference or the tourist preference, distinguishing between tourist preference and native preference, and spatial smoothing based on the spatial pyramid.

3.5.2.1 Results and Analysis

First, we present the experimental results of the comparison between recommendation methods with well-tuned parameters. Figures 3.5 and 3.6 report the effectiveness of recommendation on the Foursquare and Twitter datasets, respectively. From the figures, we observe that the accuracy values gradually rise with respect to the increase of k. This is because, by returning more spatial items, it is more likely that items that users would like to visit will be discovered. We show the performance when k is set to 5 and 10.



(a) Out-of-Town



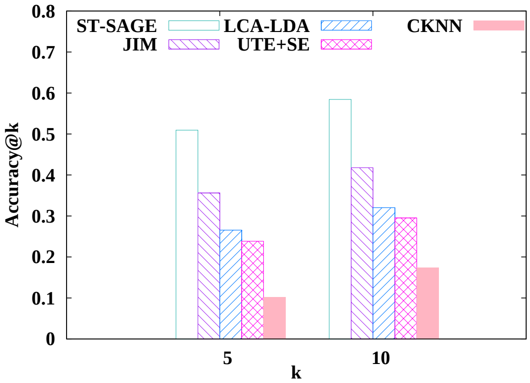
(b) Home-Town

FIGURE 3.5: Performance on Foursquare Dataset

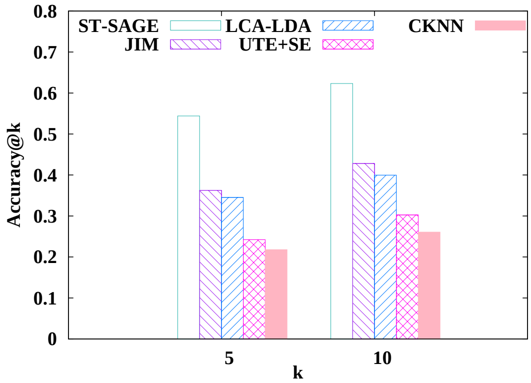
It is apparent that the recommendation methods have significant performance disparity in terms of the top-k accuracy. Figure 3.5(a) presents the recommendation accuracy in the scenario of out-of-town recommendation where the accuracy of ST-SAGE is about 0.457 when $k = 5$, and 0.535 when $k = 10$. This means that there is a probability of 45.7% that ST-SAGE will place an appealing point of interest in the top-5 recommendations and a 53.5% probability that it will be placed in the top-10 recommendations. Clearly, our proposed ST-SAGE model outperforms other competitor models significantly, demonstrating the advantages of ST-SAGE over other competitor methods. Several observations are made from the results: 1) ST-SAGE outperforms two other models (UTE+SE [84] and JIM [82]), demonstrating the advantages of ST-SAGE over other competitor methods which also integrate the temporal information. 2) CKNN performs worst as it does not explore the spatial-temporal dynamics since the local experts discovered by CKNN have the same or similar interests as target users. 3) UPS-CF falls behind ST-SAGE and LCA-LDA, showing the advantages of using the latent topic models to capture users' interests by exploiting the content of their visited spatial items. Through the medium of content, ST-SAGE and LCA-LDA transfer the users' interests inferred in the home town to out-of-town regions. In contrast, UPS-CF is a mixture of collaborative filtering and social filtering, which ignores the effect of content. Besides, according to the recent survey in [13], for movement farther than 100km from home location, the probability of visiting the exact same location as a friend has visited in the past is low.

Figure 3.6 shows the recommendation effectiveness on the Twitter dataset. As the social relationship is unavailable on this dataset, UPS-CF has not been evaluated on this dataset. The trend of the comparison result is similar to that presented in Figure 3.5.

There are two parameters in ST-SAGE, namely, the height of the spatial pyramid (H) and the



(a) Out-of-Town



(b) Home-Town

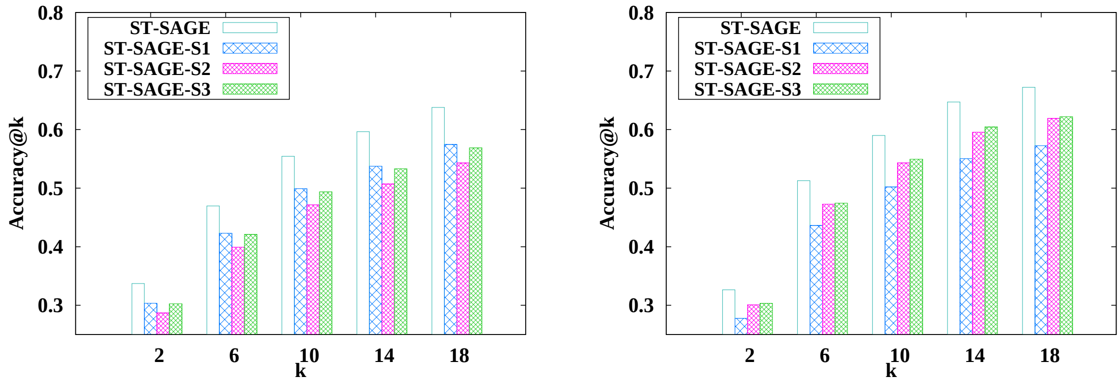
FIGURE 3.6: Performance on Twitter Dataset

number of topics (K). The experimental results presented above are obtained with the optimal parameter settings: (1) the optimal height of the spatial pyramid is 5 for both Foursquare and Twitter datasets; (2) the optimal values of K are 60 for the Foursquare dataset, and 100 for the Twitter dataset.

3.5.2.2 Impact of Different Factors

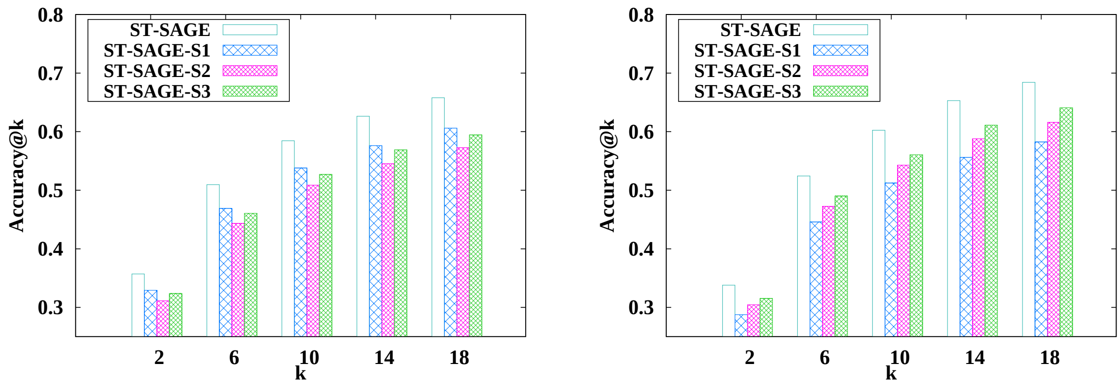
To validate the respective benefits acquired by exploiting the temporal information and the spatial smoothing based on the spatial pyramid, we compare ST-SAGE with the three variant versions: ST-SAGE-S1, ST-SAGE-S2 and ST-SAGE-S3. The results of this comparison are shown in Figures 3.7 and 3.8. From the results, we observe that ST-SAGE consistently outperforms the three variant versions for both out-of-town recommendation and home-town recommendation, which evaluates that the benefit of exploiting the temporal information and the spatial smoothing. We also observe that ST-SAGE-S1 performs better in out-of-town recommendation while ST-SAGE-S2 performs better in home-town recommendation on two datasets. This can be explained by that the personal activity is sparse in out-of-town recommendation and this sparsity leads to inaccuracy in inferring the personal temporal preference in ST-SAGE-S2.

To study the impact of two parameters in ST-SAGE, i.e. H and K, we tried different setups for these two parameters. Due to space constraints, we only show the Accuracy@10 for the out-of-town recommendation on the Foursquare dataset. We tested the performance of the ST-SAGE model by varying the height of spatial pyramid H from 2 to 7 and the number of topics K from 30 to 80. The results are presented in Table 3.2. From the results, we observe that as H increases, the $Accuracy$ values of ST-SAGE first increase, and then decrease. One possible reason for early increasing of $Accuracy$ values is that increasing the height increases the exploitation of spatial effect and makes



(a) Out-of-Town (b) Home-Town

FIGURE 3.7: Impact of Different Factors on Foursquare Dataset



(a) Out-of-Town (b) Home-Town

FIGURE 3.8: Impact of Different Factors on Twitter Dataset

the inference of native and tourist preferences more precise. Later, $Accuracy$ decreases as H gets larger, because increasing the height makes users' activity data in a region cell sparser. ST-SAGE achieves its best performance when the height of spatial pyramid is set to 5, which could be a trade-off of aforementioned two factors. On the other hand, we also observed that the performance first improves with the increase of the number of topics K and then the increment becomes small. The reason is that K represents the model complexity. Thus, when K is too small, the model has limited ability to describe the data. However, when K exceeds a threshold ($K = 60$), the model is complex enough to handle the data. At this point, it is less helpful to improve the model performance by increasing K. Thus, we choose $H = 5, K = 60$ as the best trade-off between the accuracy and efficiency on the Foursquare dataset.

TABLE 3.2: Impact of Parameters

Height	Number of Topics					
	30	40	50	**60**	70	80
2	0.46	0.498	0.518	0.53	0.534	0.535
3	0.464	0.502	0.525	0.539	0.54	0.543
4	0.471	0.502	0.529	0.546	0.551	0.554
5	0.478	0.513	0.534	**0.554**	0.556	0.555
6	0.456	0.492	0.513	0.522	0.522	0.527
7	0.421	0.451	0.476	0.485	0.489	0.492

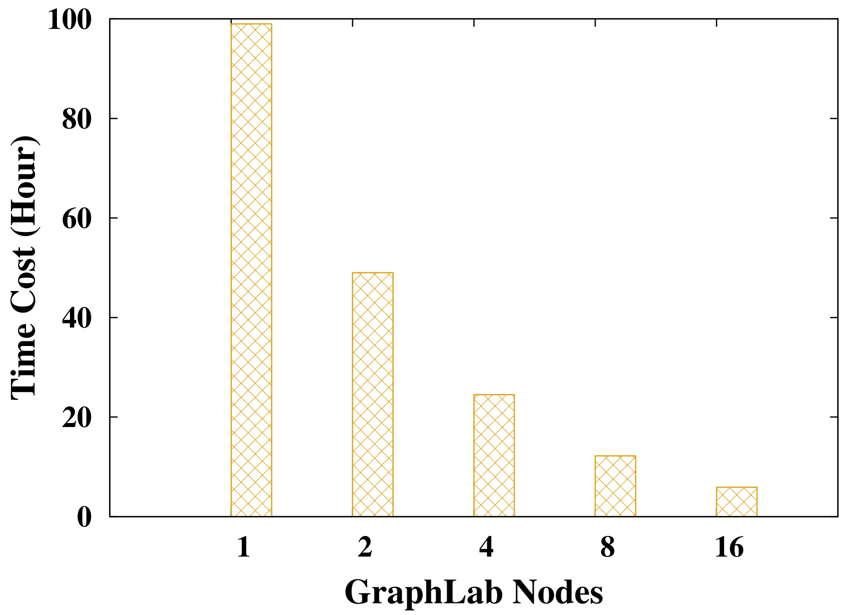


FIGURE 3.9: Training Time of ST-SAGE on GraphLab

3.5.3 Model Training Efficiency

We extend the model inference in Algorithm 2 to the distributed GraphLab system to improve the training efficiency. The training time of ST-SAGE on the Foursquare dataset is reported under different machine settings in Figure 3.9.

The results in Figure 3.9 demonstrate that the training time of ST-SAGE decreases significantly with growing size of distributed GraphLab nodes. We reduce the training time for ST-SAGE from approximately one hundred hours to to less than 10 hours. Though the basic implementation of ST-SAGE is costly, the parallel implementation guarantees the efficiency very well. This clearly shows the advantage of ST-SAGE's design, which has a model structure that is sufficiently loosely coupled to guarantee the parallel processing. This model is feasible in practical deployment.

3.6 Conclusion

In this chapter, we propose a spatial-temporal sparse additive generative model, ST-SAGE, for spatial item recommendation, which effectively overcomes the challenges arising from travel locality and spatial-temporal dynamics of user behavior. Specifically, to combat travel locality, ST-SAGE exploits both the co-occurrence pattern of spatial items and their content to infer and transfer user interests. To address spatial dynamics of user behavior, ST-SAGE incorporates the native or tourist preference at the target location. To combat the data sparsity in modeling the temporal dynamics of user behavior, we enhance the model with smoothing by taking advantage of the temporal preferences of the other similar users. To alleviate the data sparsity confronted by the inference of native preference and tourist preference for each region, ST-SAGE employs an additive framework to smooth the preferences over a well-designed spatial pyramid. Besides, we develop a scalable and parallel learning algorithm for ST-SAGE based on the GraphLab to improve the efficiency of the model training, and efficient top-k query processing techniques are deployed to speed up the process of online recommendation. We conduct extensive experiments to evaluate the performance of our ST-SAGE model on two real datasets and one large-scale synthetic dataset. The experimental results reveal the advantages of ST-SAGE over other spatial item recommendation methods, for both out-of-town and home-town recommendations in terms of both recommendation effectiveness and efficiency, which demonstrate the effectiveness of ST-SAGE in facilitating travel for users in their home towns as well as in regions they are not familiar with.

Chapter 4

Leveraging Sequential Information

4.1 Overview

Although human movement exhibits sequential patterns in LBSNs, most current studies on POI recommendations do not consider the sequential influence of locations. Leveraging sequential patterns in POI recommendation is, however, very challenging, considering $1)$ users' check-in data in LBSNs has a low sampling rate in both space and time, which renders existing location prediction techniques on GPS trajectories ineffective; $2)$ the prediction space is extremely large, with millions of distinct locations as the next prediction target, which impedes the application of classical Markov chain models; and $3)$ there is no existing framework that unifies users' personal interests and the sequential influence of recently visited locations in a principled manner.

In light of the above challenges, we propose a sequential personalized POI recommendation framework (*SPORE*) which introduces a novel latent variable *topic-region* to model and fuse sequential influence with personal interests in the latent and exponential space. The advantages of modeling the sequential effect at the topic-region level include a significantly reduced prediction space, an effective alleviation of data sparsity and a direct expression of the semantic meaning of users' spatial activities. We evaluate the performance of SPORE on two real datasets. The results demonstrate a significant improvement in SPORE's ability to recommend POIs compared with the state-of-the-art methods.

4.2 The SPORE Model

In this section, we first present the formal definition of the sequential recommendation, and then describe our sequential personalized recommendation model based on it.

4.2.1 Problem Formulation

Note that, the notations in this chapter are the same with those listed in Table 1.1 if they are not defined specially in this section.

Definition 7 ***(Sequence)*** *A sequence of user u, denoted by $S_u = \{(v_1, t_1), (v_2, t_2), \ldots, (v_n, t_n)\}$, consists of an ordered list of elements, where each element (v_i, t_i) indicates that user u visited spatial item v_i at time t_i ($1 \leqslant i \leqslant n$ and $t_1 \leqslant t_2 \leqslant \ldots \leqslant t_n$).*

Definition 8 ***(Predecessor, Successor)*** *Given a sequence $S_u = \{(v_1, t_1), (v_2, t_2), \ldots, (v_n, t_n)\}$ and a time period threshold $\triangle T$, if v_i and v_j are two items in this sequence and $0 < t_i - t_j \leqslant \triangle T$, we say v_j is a **predecessor** of v_i. Conversely, v_i is a **successor** of v_j.*

Definition 9 ***(Predecessor Set)*** *Given a target user u and time t, the predecessor set, denoted as $P_{u,t} = \{v_i | v_i \in S_u, 0 < t - t_i \leqslant \triangle T\}$, is a set of spatial items visited before t in the given time threshold $\triangle T$.*

Following the previous works [88, 87], we assume that only the spatial items in the predecessor set have sequential influence on a user's decision-making. That is, if the temporal interval between two spatial items is greater than the specified threshold $\triangle T$, it is assumed that there is no influence between the two items. We will study the impact of $\triangle T$ on the quality of spatial item recommendation in Section 4.4.3.

Definition 10 ***(User Activity)*** *A user activity is defined as a five tuple $(u,\ v,\ W_v,\ t,\ P)$ in this chapter, which indicates that the user u visits the spatial item v described as W_v at time t. P is the predecessor set of spatial items that user u has visited before v.*

Then, given a dataset D as the union of a collection of user profiles, we aim to provide spatial item recommendations for users, stated as follows.

Problem 3 *Given a user activity dataset D and a querying user u at time t (i.e., the query is $q = (u,\ t)$), our goal is to recommend a list of spatial items that u would be interested in.*

4.2.2 Model Structure

Overview of SPORE. SPORE is a probabilistic generative model that aims to mimic the process of human decision making when visiting spatial items. It assumes that a user u's decision-making at time

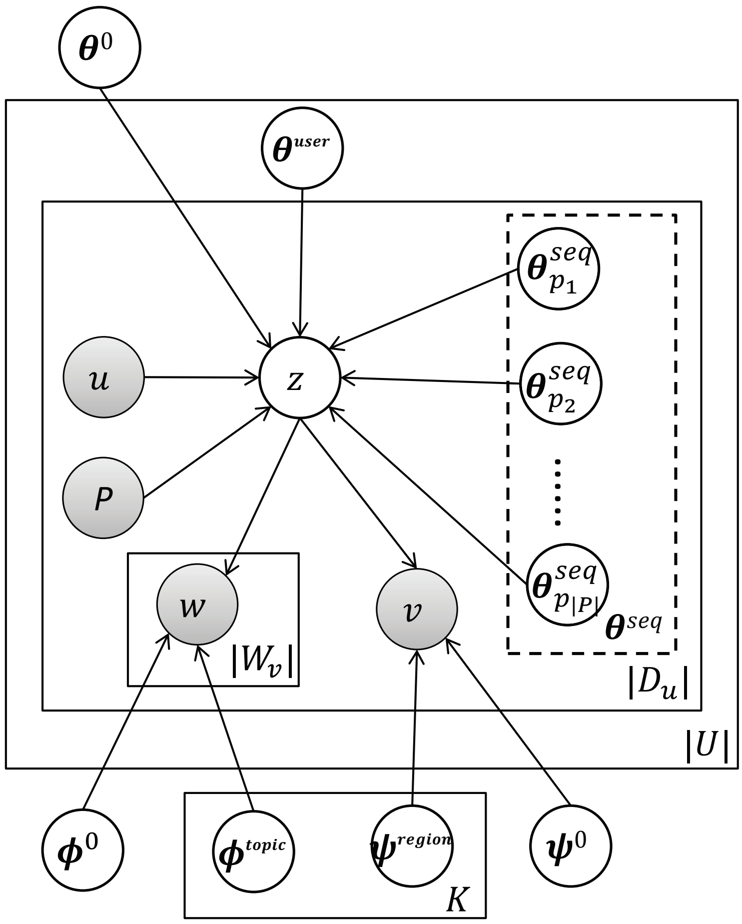


FIGURE 4.1: The Graphical Representation of SPORE

t is influenced by three factors: 1) her personal interests $\boldsymbol{\theta}^{user}_u$; 2) the influence of the items visited before t, $\boldsymbol{\theta}^{seq}_P = \{\boldsymbol{\theta}^{seq}_{p_1}, \boldsymbol{\theta}^{seq}_{p_2}, \ldots, \boldsymbol{\theta}^{seq}_{p_{|P|}}\}$; 3) the general public's preferences $\boldsymbol{\theta}^0$. Figure 4.1 shows the graphical representation of SPORE, and Table 4.1 introduces the notations of model parameters. Our input data (i.e., users' activity profiles) is modeled as observed random variables in our model, shown as shaded circles in Figure 4.1. Because a spatial item has both semantic and geographical attributes, we introduce a joint latent variable *topic-region* which corresponds to both a semantic topic (i.e., a soft cluster of words) and a geographical region (i.e., a soft cluster of locations). All three components (i.e., personal interests, sequential effect and the public's preferences) are modeled as a distribution over a set of topic-regions and influence u's decision-making by generating a topic-region z.

The set of topic-regions is obtained by simultaneously mining both the co-occurrence patterns of spatial items and their content information (e.g., tags and categories). By exploring the co-occurrence of spatial items, we capture the latent region information because frequently co-visited spatial items tend to be geographically close. Many recent studies have shown that people tend to explore items near the ones they have visited before [42]. The introduction of the latent variable *topic-region* is very helpful in alleviating the issues caused by data sparsity and low-sampling rates. It also contributes to reducing the prediction and parameter space, as the number of topic-regions is much smaller than the number of spatial items.

Personalization Component. Inspired by early work on user interest modeling [30], SPORE adopts latent topic-regions to characterize users' interests in terms of both semantic and geographical aspects. Specifically, we infer an individual user's interest distribution over a set of topic-regions according to her visited spatial items and their associated contents, denoted as $\boldsymbol{\theta}_u^{user}$. To capture both the semantic and spatial information, a topic-region z in SPORE is associated with a word distribution $\boldsymbol{\phi}_z^{topic}$ and a distribution over spatial items $\boldsymbol{\psi}_z^{region}$ simultaneously. In this way, SPORE enables these two variables to be mutually influenced and enhanced during the topic-region discovery process. The discovered topic-regions cluster the content-similar items and also groups together items with similar spatial information. Note that in the traditional topic models such as LDA, a document contains a mixture of topics, and each word has a hidden topic label. This is reasonable for long documents. However, the "document" W_v associated with a check-in activity is usually short, and is likely to only contain a single topic [89]. Thus in SPORE, all the words in W_v are assigned with a single topic z, and they are generated from the same word distribution $\boldsymbol{\phi}_z^{topic}$.

Sequential Component. Given the i^{th} check-in activity of user u, let its timestamp be $t_{u,i}$, we model the influence of each spatial item in the predecessor set $P_{u,t_{u,i}}$ denoted as $\boldsymbol{\theta}_{p1}^{seq}$, $\boldsymbol{\theta}_{p2}^{seq}$, ..., $\boldsymbol{\theta}_{p|P_{u,t_{u,i}}|}^{seq}$. To reduce the exponential complexity of classical nth-order Markov Chain to the linear complexity, an intuitive solution is to add the influence of each predecessor, inspired by the idea of nth additive Markov Chain in [88, 87]. However, using the nth additive Markov Chain has the following limitations: 1) it is an ad-hoc method that requires manual setting of weighting schemes with a decay rate parameter; 2) it ignores the difference between various users by using one common decay rate parameter for all users. In reality, different users tend to have different check-in frequencies and time intervals. Thus, the influence of nth items in the predecessor sets of each user are different. That is, the weights of predecessors should be personalized. One natural solution to overcome the two limitations is to learn personalized weights for each user using a traditional mixture model as in Equation 4.1. However, training data is often sparse in LBSNs, especially for each user. It is hard to infer the weighting variables accurately with limited training data. To handle this sparsity problem, inspired by SAGE, SPORE transforms the traditional mixture model into a mixture occurring in terms of natural parameters of the exponential family instead of distributions to avoid computing a weighting scheme for each user, as in Equation 4.2. In this way, SPORE also reduces the exponential complexity $|V|^{n+1}$ of classical nth order Markov Chain into $|V| \times K$ (n corresponds to $|P_{u,t}|$ in SPORE).

$$\begin{aligned} P(v|\boldsymbol{\theta}^{seq}_{P_{u,t_{u,i}}}) &= \sum_{j=1}^{|P_{u,t_{u,i}}|} \lambda_{u,j} P(v|\boldsymbol{\theta}^{seq}_{p_j}) \\ \sum_{j=1}^{|P_{u,t_{u,i}}|} \lambda_{u,j} &= 1 \ (\lambda_{u,j} \geq 0) \end{aligned} \tag{4.1}$$

$$P(v|\boldsymbol{\theta}^{seq}_{P_{u,t_{u,i}}}) = P(v|\boldsymbol{\theta}^{seq}_{p_1} + \boldsymbol{\theta}^{seq}_{p_2} + \ldots + \boldsymbol{\theta}^{seq}_{p_{|P_{u,t_{u,i}}|}}) = \frac{exp(\sum_{j=1}^{|P_{u,t_{u,i}}|} \theta^{seq}_{p_j,v})}{\sum_{v'} exp(\sum_{j=1}^{|P_{u,t_{u,i}}|} \theta^{seq}_{p_j,v'})} \tag{4.2}$$

Public Preference Component. We incorporate the public's preference $\boldsymbol{\theta}^0$ to further alleviate the issue of data sparsity. When users have few check-in activities in their profiles, the public's common preference provides important references for effective recommendations. Additionally, $\boldsymbol{\theta}^0$ plays the role of a background model to make the learned users' interests more discriminative and personalized, since $\boldsymbol{\theta}^0$ captures the common topics among users. Similarly, we introduce background models for words and spatial items as $\boldsymbol{\phi}^0$ and $\boldsymbol{\psi}^0$, respectively. The background models make the relevant parameters learned for topic-regions more discriminative and meaningful, since $\boldsymbol{\phi}^0$ and $\boldsymbol{\psi}^0$ assign high probabilities to non-discriminative and non-informative words and items. We expect such words or spatial items to be accounted for by the background models.

4.2.3 Generative Process of SPORE

The generative process of the SPORE model for a user check-in activity is as follows.

- Draw a topic-region index $z_{u,i}$

  $z_{u,i} \sim P(z_{u,i}|P_{u,t_{u,i}}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{seq})$

- For each content word $w_{u,i,n}$ in $W_{u,i}$, draw

  $w_{u,i,n} \sim P(w_{u,i,n}|\boldsymbol{\phi}^0, z_{u,i}, \boldsymbol{\phi}^{topic})$

- Draw a spatial item $v_{u,i}$

  $v_{u,i} \sim P(v_{u,i}|\boldsymbol{\psi}^0, z_{u,i}, \boldsymbol{\psi}^{region})$

For each user activity, SPORE first chooses the topic-region this activity is about. To generate the topic-region index z, we utilize a multinomial model as expressed in Equation 4.3.

$$P(z_{u,i}|P_{u,t_{u,i}}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{seq}) = P(z_{u,i}|\boldsymbol{\theta}^0 + \boldsymbol{\theta}^{user}_u + \boldsymbol{\theta}^{seq}_{P_{u,t_{u,i}}}) \tag{4.3}$$

Variable	Interpretation
K	the number of topic-regions
$z_{u,i}$	the topic-region assigned to spatial item $v_{u,i}$
$\boldsymbol{\theta}^0$	the topic-region distribution of the background
$\boldsymbol{\theta}_u^{user}$	the topic-region distribution, representing the intrinsic interest of user u
$\boldsymbol{\theta}_{p_{j,v}}^{seq}$	the topic-region distribution of v^{th} spatial item in p_j, representing the influence of the v^{th} item
$\boldsymbol{\phi}_z^{topic}$	content word distribution of topic-region z
$\boldsymbol{\psi}_z^{region}$	spatial item distribution of topic-region z

TABLE 4.1: Notations of Model Parameters

where $\boldsymbol{\theta}^{seq}_{P_{u,t_{u,i}}}$ is the sum of the influences of all the visited spatial items in $P_{u,t_{u,i}}$. It can be expanded as $\sum_{n=1}^{n=|P_{u,t_{u,i}}|} \boldsymbol{\theta}^{seq}_{p_{u,t_{u,i},n}}$. Once the topic-region z is generated, the spatial item v and the associated content words are generated as expressed in Equations 4.4 and 4.5, respectively.

$$P(v_{u,i}|\boldsymbol{\psi}^0, z_{u,i}, \boldsymbol{\psi}^{region}) = P(v_{u,i}|\boldsymbol{\psi}^0 + \boldsymbol{\psi}^{region}_{z_{u,i}}) \tag{4.4}$$

$$P(w_{u,i,n}|\boldsymbol{\phi}^0, z_{u,i}, \boldsymbol{\phi}^{topic}) = P(w_{u,i,n}|\boldsymbol{\phi}^0 + \boldsymbol{\phi}^{topic}_{z_{u,i}}) \tag{4.5}$$

4.2.4 Model Inference

Our goal with model inference is to learn the parameters that maximize the marginal log-likelihood of the observed random variables **w** and **v**. Marginalization is performed with respect to the latent random variable **z**. However, it is difficult to achieve maximization directly. Therefore, we apply a mixture between EM and a Monte Carlo sampler, called the Gibbs EM algorithm [70], to maximize the complete data likelihood in Equation 4.7, where Θ is the set of all the parameters. In the E-step, we sample latent topic-region assignments by fixing all of the other parameters using Gibbs sampling. In the M-step, we optimize the model parameters Θ by fixing all topic-region assignments. The two steps are iterated until convergence.

More specifically, we iteratively draw the latent topic-region **z** for all check-in activities in the E-step. According to the Gibbs Sampling, when sampling $z_{u,i}$ as expressed in Equation 4.10, we assume all other variables are fixed. $\boldsymbol{z}_{\neg u,i}$ represents the topic-region assignments for all user activities except

$$\alpha_{u,P,z} = \frac{exp(\boldsymbol{\theta}^0_z + \boldsymbol{\theta}^{user}_{u,z} + \sum_{i=1}^{|P|} \boldsymbol{\theta}^{seq}_{P_i,z})}{\sum_{zz} exp(\boldsymbol{\theta}^0_{zz} + \boldsymbol{\theta}^{user}_{u,zz} + \sum_{i=1}^{|P|} \boldsymbol{\theta}^{seq}_{P_i,zz})}, \beta_{z,w} = \frac{exp(\boldsymbol{\phi}^0_w + \boldsymbol{\phi}^{topic}_{z,w})}{\sum_{ww} exp(\boldsymbol{\phi}^0_{ww} + \boldsymbol{\phi}^{topic}_{z,ww})}, \gamma_{z,v} = \frac{exp(\boldsymbol{\psi}^0_v + \boldsymbol{\psi}^{region}_{z,v})}{\sum_{vv} exp(\boldsymbol{\psi}^0_{vv} + \boldsymbol{\psi}^{region}_{z,vv})} \tag{4.6}$$

$$\begin{aligned} P(\mathbf{z},\ \mathbf{w},\ \mathbf{v}|\circleddash,\ \mathbf{u},\ \mathbf{P}) &= P(\mathbf{z}|\mathbf{u},\ \mathbf{P}, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{seq}) P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}) P(\mathbf{v}|\mathbf{z}, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{region}) \\ &= \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \alpha_{u,P_{u,t_{u,i}},z_{u,i}} \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{u,i,n}} \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \gamma_{z_{u,i},v_{u,i}} \end{aligned} \tag{4.7}$$

the i^{th} activity for user u.

$$P(z_{u,i}|\boldsymbol{z}_{\neg u,i},\ \mathbf{w},\ \mathbf{v},\ \mathbf{u},\ \mathbf{P}, \circleddash) \propto \quad \alpha_{u,P_{u,t_{u,i}},z_{u,i}} \times \prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{u,i,n}} \times \gamma_{z_{u,i},v_{u,i}} \tag{4.10}$$

In the M-step, we optimize the parameters $\circleddash$ to maximize the log likelihood of the objective function with all topic-region assignments fixed. To update the parameters, we use the gradient descent learning algorithm PSSG (Projected Scaled Sub-Gradient) [46], which is designed to solve optimization problems with L1 regularization on parameters. More importantly, PSSG is scalable because it uses the quasi-Newton strategy with a line search that is robust for common functions. Let L be the log-likelihood of the model. According to the limited-memory BFGS [46] updates for the quasi-Newton method, the gradients of model parameters $\boldsymbol{\theta}^0$, $\boldsymbol{\theta}^{user}$ and $\boldsymbol{\theta}^{seq}$ are provided as follows.

$$\frac{\partial L}{\partial \boldsymbol{\theta}^0_z} = d(z) - \sum_{u=1}^{|U|} \sum_{i=1}^{|D_u|} \alpha_{u,P_{u,t_{u,i}},z} \tag{4.11}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}^{user}_{u,z}} = d(u,z) - \sum_{i=1}^{|D_u|} \alpha_{u,P_{u,t_{u,i}},z} \tag{4.12}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}^{seq}_{v,z}} = d(v,z) - \sum_{j=1}^{|D_v|} \alpha_{u_j,P_j,z} \tag{4.13}$$

where $d(z)$ is the number of activities assigned to topic-region z; $d(u,z)$ represents how many activities are assigned to topic-region z in D_u; $d(v,z)$ denotes the number of activities whose predecessor set contains the spatial item v assigned to topic-region z; D_v is the set of activities whose predecessor set contains the spatial item v; u_j denotes the user who generates the j^{th} activity record in D_v and P_j is the predecessor set of the j^{th} activity in D_v.

Similarly, the gradients of model parameters $\boldsymbol{\phi}^0$, $\boldsymbol{\phi}^{topic}$, $\boldsymbol{\psi}^0$, and $\boldsymbol{\psi}^{region}$ are computed as follows:

$$\frac{\partial L}{\partial \boldsymbol{\phi}^0_w} = d(w) - \sum_{z=1}^{K} d(z) \times \beta_{z,w} \tag{4.14}$$

$$\frac{\partial L}{\partial \boldsymbol{\phi}^{topic}_{z,w}} = d(z,w) - d(z) \times \beta_{z,w} \tag{4.15}$$

$$\frac{\partial L}{\partial \boldsymbol{\psi}_{v}^{0}} = d(v) - \sum_{z=1}^{K} d(z) \times \gamma_{z,v} \tag{4.16}$$

$$\frac{\partial L}{\partial \boldsymbol{\psi}_{z,v}^{region}} = d(z,v) - d(z) \times \gamma_{z,v} \tag{4.17}$$

where $d(w)$ is the number of activities where the word w appears, and $d(z,w)$ is the number of activities where the word w is assigned to the topic-region z. $d(v)$ is the number of activities associated with item v, and $d(z,v)$ represents the number of activities in which topic-region z is assigned to v.

It is worth mentioning that the Gibbs EM algorithm can be easily expressed and implemented in the GraphLab framework, and the inference procedure can be naturally decomposed for parallel processing.

4.3 Spatial Item Recommendation using SPORE

Once we have estimated the model parameter set $\ominus$, given a querying user u_q at time t_q, we first retrieve the spatial item predecessor set P_q for u_q. Then, we compute the probability of user u_q choosing each unvisited spatial item v with description W_v, as in Equation 4.18. Then, we return the top-k spatial items with the highest probabilities as recommendations.

$$\begin{aligned}
&P(\, v,\ W_v|\ominus,\ u_q,\ P_q) = \sum_{z=1}^{K} P(\, v,\ W_v,\ z|\ominus,\ u_q,\ P_q) \\
&= \sum_{z=1}^{K} P(z|u_q, P_q, \boldsymbol{\theta}^0, \boldsymbol{\theta}^{user}, \boldsymbol{\theta}^{seq}) P(W_v|z, \boldsymbol{\phi}^0, \boldsymbol{\phi}^{topic}) P(v|z, \boldsymbol{\psi}^0, \boldsymbol{\psi}^{region}) \\
&= \sum_{z=1}^{K} \alpha_{u_q,P_q,z} (\prod_{n=1}^{|W_v|} \beta_{z,w_{v,n}})^{\frac{1}{|W_v|}} \gamma_{z,v}
\end{aligned} \tag{4.18}$$

In Equation 4.18, W_v denotes the content words describing item v and $w_{v,n}$ is the n^{th} word in W_v. We adopt the geometric mean for the probability of topic z generating the word set W_v, considering that the number of content words is different for different spatial items.

To accelerate the online recommendation process, we propose a ranking framework in Equation 4.19 which separates the offline computation from the online calculation to the maximum extent.

$$\begin{aligned}
&S(q,v) = \sum_{z=1}^{K} F(z,v) W(q,z) \\
&F(z,v) = (\prod_{n=1}^{|W_v|} \beta_{z,w_{v,n}})^{\frac{1}{|W_v|}} \times \gamma_{z,v} \\
&W(q,z) = \alpha_{u_q,P_q,z}
\end{aligned} \tag{4.19}$$

	Foursquare	Twitter
# of users	4,163	114,508
# of items	121,142	62,462
# of check-ins	483,813	1,434,668
time span	12/2009–07/2013	09/2010–01/2011

TABLE 4.2: Statistics of The Two Datasets

where $F(z, v)$ represents the offline scoring part which denotes the score of spatial item v with respect to dimension z. This part is computed offline since it is independent from the query $q = (u_q, t_q)$. On the other hand, $W(q, z)$ is inferred in the online part, denoting the preference of query q on dimension z. Note that the main time-consuming components of $W(q, z)$ are also computed offline (e.g., θ^0, θ^{user} and θ^{seq}). This design enables separation the online and offline computations to significantly reduce query time.

When a query $q = (u_q, t_q)$ arrives, we first compute the query preference weight on each dimension (i.e., $W(q, z)$), and then aggregate $F(z, v)$ over each dimension with the weight $W(q, z)$ for each spatial item. At last, k spatial items with the highest scores are returned as the query results. SPORE is trained offline, while recommendation performed online is a combination process of the various factors. This scheme guarantees a quick response.

4.4 Experiment

In this section, we first present the experiment settings and then demonstrate the experimental results which include the recommendation effectiveness, impact of factors and recommendation efficiency.

4.4.1 Experimental Settings

4.4.1.1 Datasets

We conducted our experiments on Foursquare and Twitter which are the same as the sets in Section 3.5.1.1. The basic statistics of them are shown in Table 4.2.

4.4.1.2 Comparative Approaches

We aim to evaluate both the recommendation effectiveness and the efficiency of generating online recommendations.

To evaluate the effectiveness, we compared SPORE with the following four methods which are the state-of-the-art spatial item recommendation techniques. The first three methods exploit the sequential influence, while the fourth one does not consider the sequential effect.

First-order Markov Chain (FMC): Existing Markov Chain (MC) based methods can be divided into two main categories: classical MC [10, 36] and category based MC [11]. As the complexity of classical nth order MC increases exponentially with n, we only explored the first order of classical MC (FMC). The existing FMC considers sequential influence by deriving the sequential probability that a user u will visit her next location v_{n+1} based on only the last visited location v_n in the sequence S_u.

Category based hidden Markov model (HMM): HMM [11] uses a mixed hidden Markov model to predict the category of the user's activity at the next step and then predicts the most likely location by incorporating the estimated category distribution and spatial-temporal influence through estimated parameters for each factor. By modeling the category level, HMM significantly reduces the modeling and prediction space compared with the classic Markov model.

LORE: To overcome the limitation of FMC, LORE [88] first predicts the probability of a user visiting a location by Additive Markov Chain (AMC) which exploits the sequential effect by adding the influence of the user's recently visited locations. LORE then fuses sequential influence with geographical influence and social influence by multiplying them together. Note that, the social influence is not explored on the Twitter dataset as there is no network information available on the databset.

GT-BNMF: GT-BNMF [43, 45] is a Geographical-Topical Bayesian non-negative Matrix Factorization framework, which is designed to model the joint effect of users' interests, geographical influence and region-level POI popularity for POI recommendation. No sequence influence is considered by GT-BNMF.

To further validate the benefits brought by considering users' personal interests, exploiting the sequential influence and considering influence of recently visited items rather than the latest one only, we designed three variant versions of our model. **SPORE-V1** is the first variation of SPORE in which we do not consider users' personal interests; **SPORE-V2** is the second simplified version where the sequential influence is not exploited; and the last variant, **SPORE-V3** only considers the sequential influence of the latest visited items.

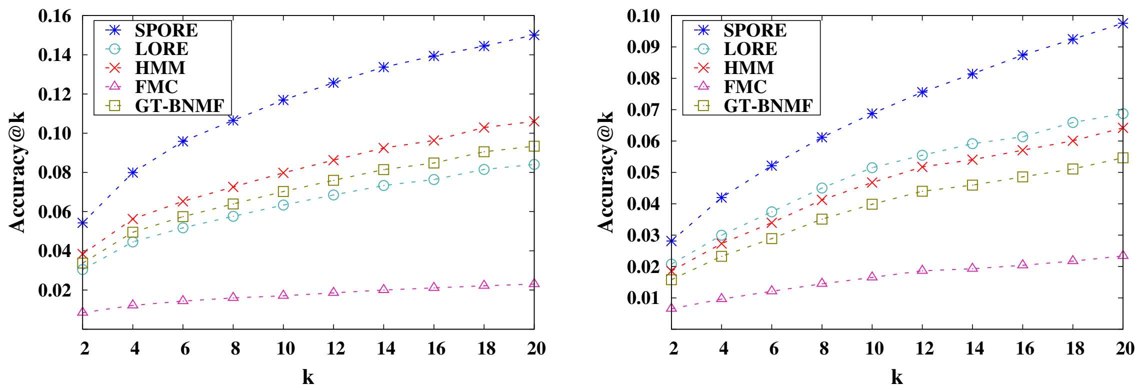
4.4.1.3 Evaluation Methods

Given a user profile, namely a collection of user activities, we first extract the activity sequence of each user. Then, we use the first 80% of activities in the sequence for each user as the training dataset D_{train} and the remaining activities as the test dataset D_{test}. To evaluate the recommendation methods, we adopt the evaluation methodology and measurement $Accuracy@k$ which is the same as the metric in Section 3.5.1.3. For each test case, we use the same method as in Section 3.5.1.3 except that we order all of items instead of randomly selected 1000 items. Thus, the $Accuracy@k$ is much lower in this section. We should note that the low accuracy values are common and reasonable in recommendation domain for the following two main reasons. First, both the Foursquare and Twitter datasets have a low density (i.e.,the densities of user-item matrix are 0.55% and 0.02% for Foursquare and Twitter datasets, respectively), which usually results in a even more sparse test matrix [84]. In addition, the spatial items in the test data of each user may represent only a small portion of POIs that the user is truly interested in. Thus, in this paper, *we focus on the relative improvements we achieved, instead of the absolute values*.

4.4.2 Recommendation Effectiveness

In this section, we present the experimental results of all the recommendation methods with well-tuned parameters. There are two parameters in SPORE, namely, the time period threshold ($\triangle T$) and the number of topic-regions (K). The experimental results presented in this part were obtained with optimal parameter settings: (1) the optimal time period thresholds are 0.5 day for Twitter dataset and 0.2 day for Foursquare dataset; (2) the optimal values of K are 100 for the Twitter dataset and 140 for the Foursquare dataset. Figures 4.2(a) and 4.2(b) report the recommendation effectiveness on the Foursquare and Twitter datasets, respectively. From the results, we observe that the accuracy values gradually increase with the increasing value of k. This is because, by returning more spatial items, it is more likely to discover the ones that users would like to visit. Note that we show only the performance when k is set between 2 and 20. Greater values of k are usually ignored for the top-k recommendation task.

As the check-in data on two datasets are very sparse, the recommendation accuracies for all the comparing methods are low. However, SPORE makes a significant improvement compared with the other competitor methods. On the Twitter dataset, the improvements, in terms of $Accuracy@10$, are 33.57%, 47.03%, 73.2% and 315.99% compared with LORE, HMM, GT-BNMF and FMC, respectively, which clearly demonstrate the advantages of our proposed SPORE model with respect to other



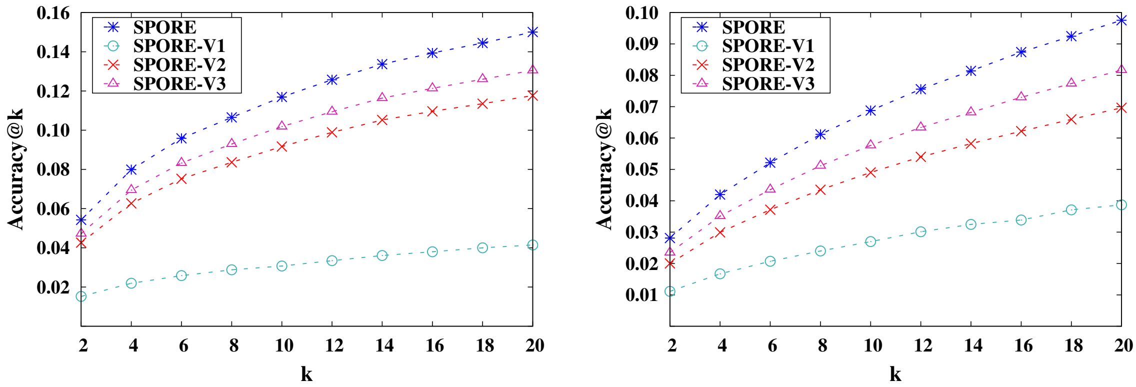
(a) Foursquare (b) Twitter

FIGURE 4.2: Performance on Foursquare and Twitter Datasets

competitor models. Several observations are made from the results: 1) SPORE, GT-BNMF, LORE and HMM outperform FMC significantly showing the advantages of incorporating both the sequential influences of all recently visited items and other factors (such as geographical, social influence). FMC only considers the sequential influence of the last visited item. 2) On the Foursquare dataset, SPORE, GT-BNMF, LORE and HMM all perform much better than on the Twitter dataset due to that the user-item matrix on Twitter dataset is much sparser than the one on the Foursquare dataset. On the other hand, the average time interval between two adjacent check-ins is much larger on the Foursquare dataset than that on the Twitter dataset. Thus, the sequential effect on the Foursquare dataset is not as obvious as on the Twitter dataset. Thus, FMC which only utilizes the sequential information performs worse on the Foursquare dataset. 3) Both LORE and HMM drop behind SPORE, showing the advantage of seamlessly integrating the multiple factors into a unified framework by avoiding the inference of mixture weights for each factor, especially when the activity data of each user is sparse. LORE considers the sequential effect and other factors by simply multiplying them together, which is oversimplified. HMM accomplishes the fusion by inferring a weight for each factor which is inaccurate when the data is sparse. By contrast, our SPORE adds the effects of all the factors in the exponential space to avoid the inference of weight for each factor to gain improved robustness and accuracy. 4) SPORE outperforms GT-BNMF on both datasets, demonstrating the benefits brought by considering sequential influence in personalized spatial item recommendation.

4.4.3 Impact of Different Factors

In this subsection, to validate the benefits brought by exploiting the users' personal interests, leveraging the sequential influences and considering all recently visited items rather than only the last



(a) Foursquare (b) Twitter

FIGURE 4.3: Impact of Different Factors on Both Datasets

one, we compared our SPORE model with its three variant versions: SPORE-V1, SPORE-V2 and SPORE-V3 respectively. We also studied the impact of the number of topic-regions K and the time threshold $\triangle T$.

The results of comparing SPORE with the three variant versions on both datasets are shown in Figure 4.3. The results show that SPORE consistently outperforms the three variant versions on both datasets, indicating the benefits brought by each factor, respectively. For instance, the performance gap between SPORE and SPORE-V2 validates the effectiveness of leveraging the sequential influence of visited items into recommendation. The improvement of SPORE over SPORE-V3 on both datasets shows the advantage of exploiting the influence of all the visited items in the given time threshold rather than only the last visited one. Another observation is that SPORE-V2 and SPORE-V3 outperform SPORE-V1 significantly, showing that the users' personal interests play the most important role in spatial item recommendation. It is also worth noting that the performance gap between SPORE and SPORE-V2 on the Twitter dataset is larger than that on the Foursquare dataset. This can be explained by the fact that the sequential information on the Twitter dataset is much denser than that on the Foursquare dataset.

To study the impact of the two parameters in SPORE, i.e. K and $\triangle T$, we tested different setups for these two parameters. Due to space constraints, we have only shown the experimental results for the top-10 recommendation using the Twitter dataset. We tested the performance of SPORE by varying the number of topic-regions K from 70 to 120 and the time threshold $\triangle T$ from 0.2 days to 0.7 days. The results are presented in Table 4.3. From the results, we observe that the performance first improves quickly with the increase of the number of topic-regions and then the increment becomes small. The number of the topic-regions represents the model complexity. Thus, when K is too

$\triangle T$	K					
	70	80	90	**100**	110	120
0.2	0.0620	0.0630	0.0642	0.0654	0.0656	0.0657
0.3	0.0633	0.0647	0.0660	0.0668	0.0670	0.0670
0.4	0.0644	0.0660	0.0671	0.0680	0.0681	0.0682
0.5	0.0650	0.0667	0.0679	**0.0687**	0.0689	0.0690
0.6	0.0652	0.0670	0.0681	0.0687	0.0690	0.0691
0.7	0.0653	0.0671	0.0681	0.0688	0.0692	0.0692

TABLE 4.3: Impact of Parameters

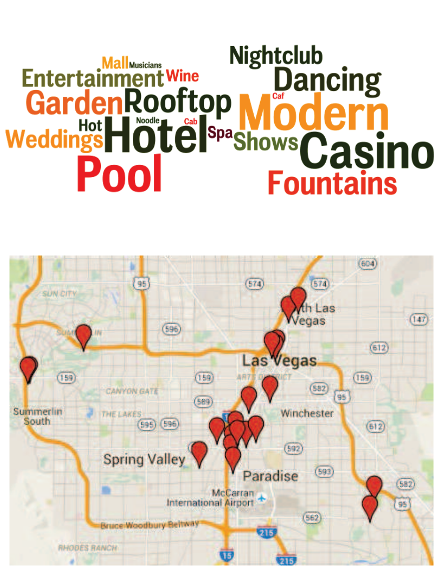
small, the model has limited ability to describe the data. However, when K exceeds a threshold (e.g., $K = 100$), the model is complex enough to handle the data. At this point, it is less helpful to improve the model performance by increasing K. A similar trend is observed for the time period threshold. There are more predecessors when $\triangle T$ becomes larger according to Definition 8. Thus, the performance improves with the increase of $\triangle T$. However, when $\triangle T$ is larger than a threshold (e.g., $\triangle T = 0.5$ day), the added predecessors have little influence on the recommendation performance. As a result, we chose $K = 100, \triangle T = 0.5$ day as the best trade-off between accuracy and efficiency on the Twitter dataset.

4.4.4 Qualitative Analysis of Topic-Regions

In this experiment, we use a case study method to demonstrate the effectiveness of SPORE in detecting topic-regions qualitatively. In this case, we study both the semantic and spatial property for each topic-region on the Twitter dataset. For the semantic analysis of the discovered topic-regions, we choose the top-20 words with the highest generation probabilities $P(w|z, \boldsymbol{\phi}^{topic})$ for each topic-region z. Based on the top words and their corresponding generation probabilities, we employ wordle[1] to create word clouds for each discovered topic-region. For the spatial analysis, we present the locations of the top-20 spatial items with the highest generation probabilities $P(v|z, \boldsymbol{\psi}^{region})$ on Google Maps for each topic-region z. We present four example topic-regions in Figure 4.4.

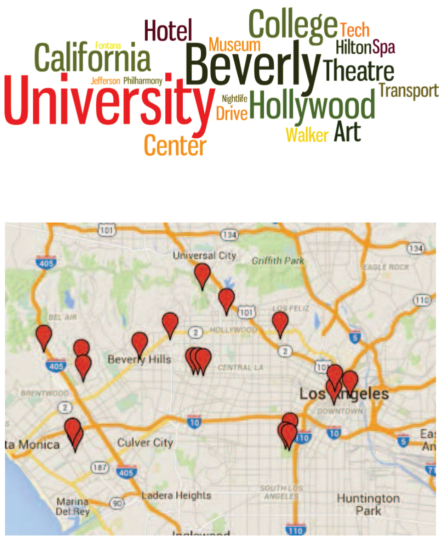
From the results, we observe that the presented four topic-regions are located in three different cities and they focus on different subjects. The topic-regions in Figures 4.4(c) and 4.4(d) are both located around Los Angeles. However, they focus on different subjects. The topic-region in Figure

[1]http://www.wordle.net/

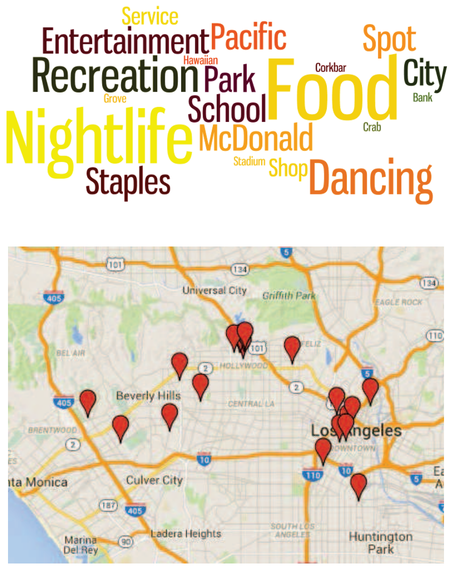


(a) Casinos & Entertainment in LV

(b) Park & Playground in NY



(c) Tourist Hot Spots in LA



(d) Food & Nightlife in LA

FIGURE 4.4: Semantic and Spatial Interpretations of Topic-Regions

4.4(c) mainly focuses on the tourist hot spots in Los Angeles, such as the universities, Beverly Hills and Hollywood while the one in Figure 4.4(d) mainly focuses on the food and nightlife related spots. Thus, the topic-regions discovered by SPORE can be interpreted both semantically and spatially.

4.5 Conclusion

In this chapter, we propose a novel sequential personalized spatial item recommendation framework (*SPORE*). To effectively overcome the challenges arising from *low-sampling rate* and *huge prediction space*, SPORE introduces a novel latent variable *topic-region* to model and fuse the sequential influence and personal interests in the latent space. A topic-region corresponds to both a semantic topic (i.e., a soft cluster of words) and a geographical region (i.e., a soft cluster of locations). The advantages of modeling sequential effects at the topic-region level include a significantly reduced prediction space, an effective alleviation of data sparsity and a direct expression of the semantic meaning of users' spatial activities. To seamlessly fuse sequential effects and personal interests in a unified and principled way, we adopt the sparse additive modeling technique to add them to exponential space thus avoiding the inference of mixture weights for each factor. Extensive experiments are conducted to evaluate the performance of SPORE on two real datasets. The results demonstrate the advantages of SPORE.

Chapter 5

Efficient On-line Recommendation

5.1 Overview

Although the online computation for each query is reduced by separating the offline computation from the online computation in both models, the online computation is still inefficient when there are a large number of POIs because it needs to scan every candidate spatial item to compute a ranking score. In many scenarios, there are a large number of candidate spatial items that may be of interest to users. For example, all the spatial items in Australia should be considered for the query where a user living in USA wants to find interesting places in Australia to make a good travel plan. As ST-SAGE supports the switching of queries between different scales of a geo-region (i.e. from country to street), the number of spatial items in some scenarios could be very large. Thus, efficient top-k recommendation is necessary by reducing the number of spatial items to scan.

5.2 Threshold-Based Query Processing Technique

5.2.1 Motivation

One natural solution for pruning the item search space is to index the spatial items using a tree structure such as R-Tree and Metric Tree, which is similar to the techniques used in KNN problems [90, 91]. Specifically, according to the Equation 3.19, both queries and spatial items are represented by vectors with K dimensions. For each query, ST-SAGE recommends those spatial items that are most related (or have nearest distance in KNN problems) to the given query. The key difference between our problem and classical KNN problems is that the distance is measured in inner-product instead of Euclidean distance. This makes the straightforward application of the techniques in KNN problems infeasible in this situation, as inner-products lack a very basic property of coincidence that generally-used similarity functions usually have. For instance, the Euclidean distance of a point to itself is 0 while the inner-product of a point v to itself is $\| v \|^2$, which may be high or low depending on the length of v. This distance coincidence is the basis of addition assumptions. Without addition assumptions, many properties (triangle inequality and coincidence) in existing efficient KNN methods are no longer valid. According to the analysis in [59], the maximum inner-product search is equivalent to nearest-neighbor search in euclidean metric space only if the norms of all the spatial items have the same length, which is, however, impractical in both ST-SAGE and SPORE.

Ram et al. proposed an adapted technique of maximum inner-product search in [59] based on the metric tree. However, according to the analysis in [49], the efficiency of the techniques based on a tree index structure suffers from the curse of the dimension K which is $O(K^{12})$. To find the exact top-k results by examining the minimum number of items, we extend the Threshold-based Algorithm (TA) [17, 81, 80] in both ST-SAGE and SPORE. This is because, given a query, the proposed ranking functions in Equation 3.19 and 4.18 are both monotonic, so it can easily be expressed in the form of a linear weighting function. As the application of TA in these two models are similar, we only present how to accelerate the on-line recommendation using TA technique in ST-SAGE.

5.2.2 Algorithm Description

In the Threshold based query processing technique, for each location at each time, we precompute K sorted lists of spatial items for K topics. For each topic z, we precompute a sorted list of spatial items according to $F(z, v)$ defined in Equation 3.19. The top spatial item in each list is the item most related to the corresponding topic. When receiving a query $q = (u, l, t)$, we first obtain K ranked lists $L_z, z \in \{1, 2, ..., K\}$. Then, Algorithm 3 is run to compute the top-k spatial items from all the spatial items in the target region and return them in the result list L. In the algorithm, PL, L and L_z are priority lists in which elements are automatically sorted according to their priorities. They have six operations: $insert$(element,priority) inserts an element into the list with a specific priority; get() returns the head element; $remove$() removes the head element; get(k) returns the k-th element; $remove$(k) removes the k-th element; $hasMore$() returns true if the list is non-empty. The result list L contains the ID of the spatial items and their ranking scores, denoted as $S(q, v)$ in Equation 3.19.

As is presented in Algorithm 3, the process of the extended TA algorithm can be generalized as follows:

ALGORITHM 3: Threshold-based algorithm

Input: A query $q = (u, l, t)$; the query preference weights $W(q, z)$, $z \in \{1, 2, ..., K\}$; priority lists $(L_1, ..., L_K)$;

Output: List L with all the k highest ranked spatial items;

```
PL, L = ∅, S_Ta = max;
for z = 1 to K do
    v = L_z.get();
    Compute S(q, v) according to Equation (3.19);
    PL.insert(z, S(q, v));
end
S_Ta = Compute_Threshold();
while true do
    nextListToCheck = PL.get();
    PL.remove();
    v' = L_nextListToCheck.get();
    L_nextListToCheck.remove();
    if v' ∉ L then
        if L.size() < k then
            L.insert(v', S(q, v'));
        end
        else
            v = L.get(k);
            if S(q, v) ≥ S_Ta then
                break;
            end
            if S(q, v) < S(q, v') then
                L.remove(k);
                L.insert(v', S(q, v'));
            end
        end
    end
    if L_nextListToCheck.hasMore() then
        v' = L_nextListToCheck.get();
        Compute S(q, v') according to Equation (3.19);
        PL.insert(nextListToCheck, S(q, v'));
        S_Ta = Compute_Threshold();
    end
    else
        break;
    end
end
return L;
```

- For each topic, maintain a ranked list L_z of spatial items according to $F(z, v)$;
- For the K ranked lists, maintain a priority list PL where the priority of a list L_z is the ranking score of the first item v in L_z (i.e., $S(q, v)$) (Lines 3-7);
- In each iteration, select the most promising item (i.e., the first item) from the list that has the highest priority in PL and add it to the result list L (Lines 10-17);
- When the size of the result list is no less than k (i.e., $|L| \geqslant k$), check the $k-$th item v in L. If its ranking score is no less than the $thresholdscore$ which is computed in Algorithm 4 (i.e., $S(q, v) \geqslant S_{T_a}$), terminate the process (Lines 19-22); else the k-th item is either replaced by the current item if its ranking score is lower than that of the current one, or reserved if otherwise (Lines 23-26).
- At the end of each iteration, update the priority of the current list as well as the threshold score. (Lines 29-34).

ALGORITHM 4: Function Compute_Threshold()

Input: A priority list PL; priority lists $(L_1, ..., L_K)$; the query preference weights $W(q, z)$, $z \in \{1, 2, ..., K\}$;

Output: The threshold score S_{Ta};

```
S_Ta = 0;
for z = 1 to K do
    v = L_z.get();
    S_Ta = S_Ta + W(q, z)F(v, z);
end
return S_Ta;
```

Algorithm 4 describes the computation of the threshold score. Given the ranking lists of the remaining unexamined items, the threshold score is obtained by aggregating $F(z, v)$ of the first item v in each list. Clearly, the threshold score is the maximum possible ranking score that can be achieved by the remaining items. It can be achieved only if the first item in all the ranking lists are the same. As a result, if the ranking score of the k-th item in the result list L is no less than the threshold score, the process can be terminated immediately because no remaining item will have a higher ranking score than the k-th item.

From the computation of the threshold score, we can see that the TA algorithm is able to find the exact top-k spatial items without computing ranking scores for all spatial items. Additionally, this algorithm is instance optimal, scanning the minimum number of spatial items, and there is no other deterministic algorithm that has a lower optimality ratio [17]. Here, instance optimality corresponds to optimality in every instance, as opposed to just worst case or average case.

5.2.3 Approximate Algorithm for Top-*k* Recommendation

The TA algorithm returns the exact top-k spatial items with the highest scores, as shown in Algorithm 3. However, in reality, the users may be more interested in a quicker response with an approximate top-k result. Specifically, for a query $q = (u, l)$ in Geo-SAGE-RS, an approximate top-k recommendation returns k spatial items with approximate highest scores ($S(q, v)$). This problem is formulated in Definition 11.

Definition 11 ***(ρ-approximate top-k recommendation)*** *Following the recent work of [80], assume $\rho > 1$, ρ-approximate top-k recommendations for a query q is defined as a list of k items such that for each item v in this list and each item v' not in this list, $\rho * S(q, v) \geqslant S(q, v')$.*

Note that the approximate recommendation with $\rho = 1$ gives the exact top-k answers.

Theorem 1 *If at least k items have been seen whose ranking scores are no less than S_{T_a}/ρ, then these k items are ρ-approximate top-k recommendation.*

PROOF to Theorem 1: "At least k items have been seen whose ranking scores are no less than S_{T_a}/ρ" indicates that for each item v in this list, $\rho * S(q, v) \geqslant S_{T_a}$. As for every v' not in L, $S_{T_a} \geqslant S(q, v')$, thus for each item v in this list and each item v' not in this list, $\rho * S(q, v) \geqslant S_{T_a} \geqslant S(q, v')$. According to Definition 11, these k items are ρ-approximate top-k recommendations.

ρ-approximate top-k recommendation can be achieved by modifying the rule in Line 19 in Algorithm 3 to "if $S(q, v) \geqslant S_{T_a}/\rho$". This new rule indicates that as soon as k items, whose ranking score is no less than S_{T_a}/ρ have been seen, then halt (based on Theorem 1).

Furthermore, we can easily extend Algorithm 3 to an interactive process so that the recommender system shows the querying user its current view of the top-k recommendations along with a guarantee of ρ-approximation to the accurate result. Specifically, we can modify the rule in Line 19 to be an interactive process, as described in Algorithm 5. In this algorithm, after the condition of ρ-approximate top-k recommendation has been satisfied by L, the current top-k result list L is shown to the querying user u. If u is satisfied with the current result, this interactive process is terminated; otherwise, we reduce ρ with a predefined decay rate and present a more accurate result to u. This process is terminated either when u is satisfied with L or $\rho = 1$ which means that L is an accurate result list.

ALGORITHM 5: Interactive Process in Recommendation

```
if S(q, v) ⩾ S_{T_a}/ρ then
    Show the current view of the top-k result list L to querying user u;
    Get feedback from u;
    if u is satisfied with L or ρ == 1 then
        break;
    end
    else
        /* η is a predefined decay rate with the range (0...1)   */
        ρ = ηρ;
    end
end
```

An example of this interactive process of ρ-approximate top-k recommendation is given in Figure 5.1. Given a query from a user u traveling to Brisbane and a predefined ρ, the recommender recommends "Little Greek Taverna" and "Cafe O-mai" to u. However, u is not satisfied with these two recommendations and chooses "dislike". Receiving this response, the recommender recomputes the result by reducing ρ with a predefined decay rate and produces more accurate recommendations: "Lady Marmalade Cafe" and "John Mills Himself". If u still does not like this result, this process will continue until u is satisfied or $\rho = 1$, which means that the exact top-k recommendations are produced.

5.2.4 Experiments

In this section, we evaluate the recommendation efficiency in Foursquare and a synthetic dataset. There are 4163 users, 121142 spatial items and 483813 check-ins in the Foursquare dataset. Every user has approximately 116 check-ins and every spatial item is associated with 4 check-ins on average. To keep the sparsity property on the synthetic dataset, we generate a dataset with 10 million check-ins, 2.5 million spatial items and 86 thousand users to simulate the distribution of the check-ins on the Foursquare dataset.

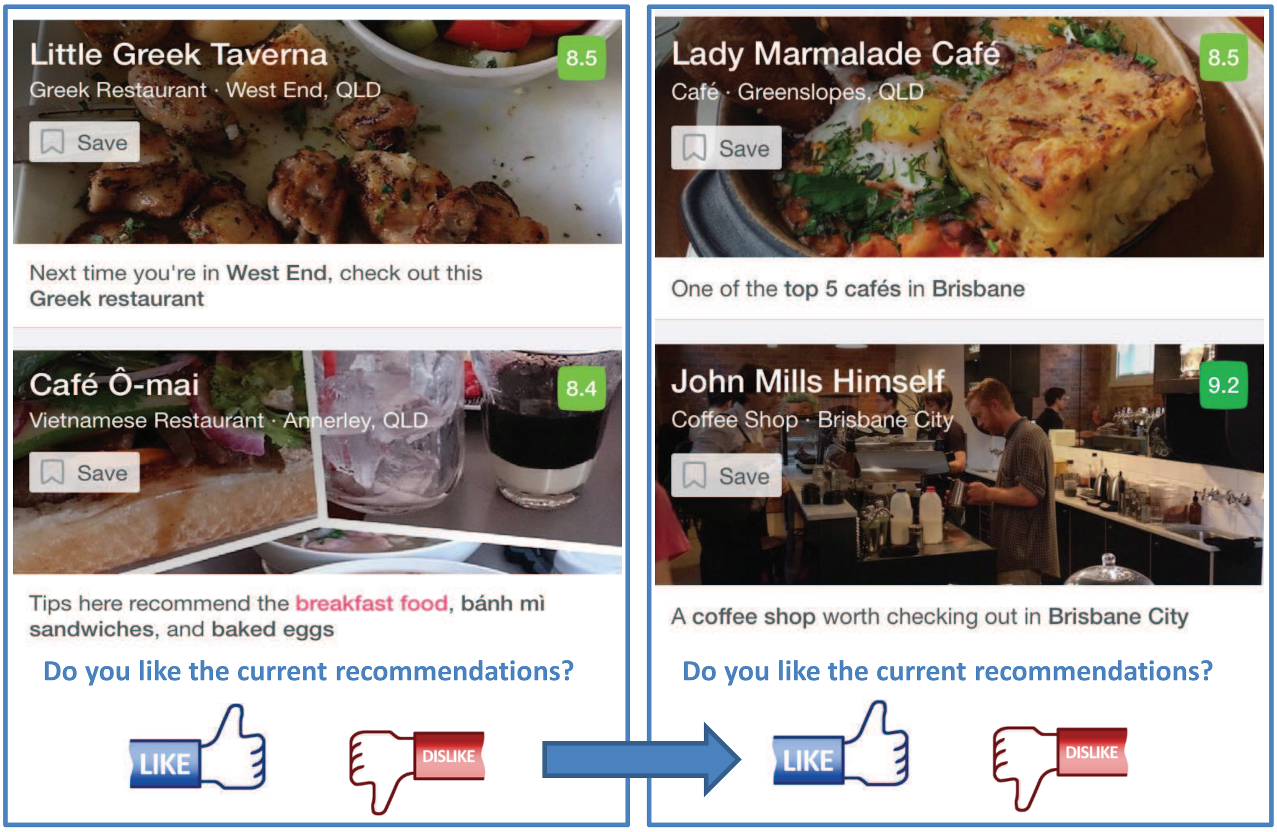
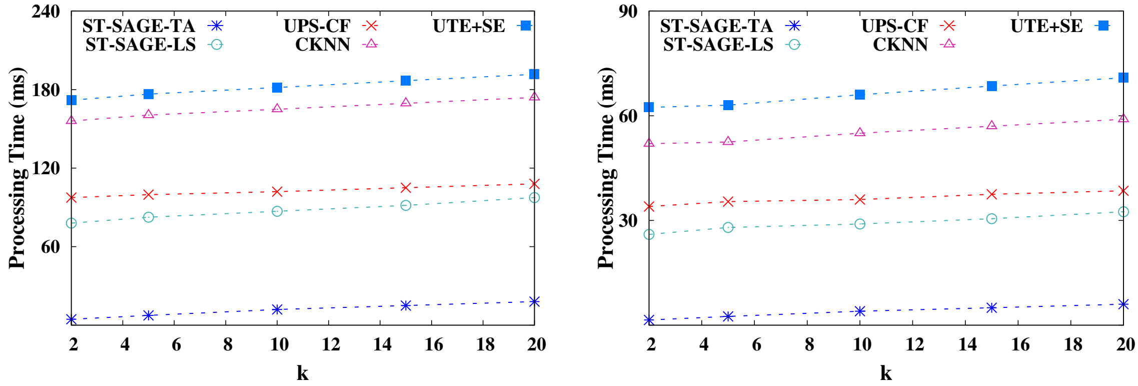


FIGURE 5.1: An Example of Interactive Process

In the efficiency study on the Foursquare dataset, we tested top-k recommendations for the target region with $50km \times 50km$ and $100km \times 100km$ respectively. As there are different numbers of candidate spatial items in different sizes of target regions, there are more spatial items in a target region with $100km \times 100km$ than in a target region with $50km \times 50km$. All the recommendation algorithms are implemented in Java 1.7 and run on a Windows Server with "Intel E5-2690" CPU and 256G RAM.

For the online recommendation efficiency test, we compare ST-SAGE with UPS-CF, CKNN and UTE+SE. We did not compare our model with LCA-LDA and JIM due to the fact that the TA algorithm can also be employed to speed up the online recommendation in these two methods, as the ranking functions in these two methods are also monotonous. For the online recommendation of ST-SAGE, we adopt two methods to utilize the knowledge learned offline to produce recommendations. The first method extends the TA algorithm to produce top-k recommendations and is denoted as "ST-SAGE-TA". The second method linearly scans all the spatial items in the target region, computes their ranking scores and then recommends the top-k items with the highest scores. This method is called "ST-SAGE-LS".



(a) Target Region with $100km \times 100km$ (b) Target Region with $50km \times 50km$

FIGURE 5.2: Recommendations Efficiency using TA on Foursquare

Figure 5.2 presents the average online efficiency of the four different methods. On average, our proposed "ST-SAGE-TA" produces top-10 recommendations in 12.1ms and 3.9ms for the target region with $100km \times 100km$ and $50km \times 50km$ respectively. From the figures, we observe that 1) "ST-SAGE-TA" outperforms "ST-SAGE-LS" significantly in all querying regions, which demonstrates that the TA-based query processing technique is efficient; 2) "ST-SAGE-TA" and "ST-SAGE-LS" consistently outperform "CKNN", "UPS-CF" and "UTE+SE" significantly in all querying regions, showing that the model-based methods produce faster responses to querying users than memory-based methods once the model parameters have been learned offline; 3) the time costs of all algorithms in the target region with $100km \times 100km$ are higher than a target region with $50km \times 50km$ due to the increase of the number of the candidate spatial items; 4) the time costs of all algorithms increase slowly with the increasing number of recommendations (k).

To evaluate the scalability of ST-SAGE, on the synthetic dataset, we vary the number of candidate spatial items from 0.5 million to 2.5 million based on the fact that, given a query and the number of recommendations, the efficiency of online recommendation largely depends on the number of available spatial items. Figure 5.3 presents the time cost of producing top-10 recommendations with varying numbers of available items from 0.5 million to 2.5 million. From Figure 5.3, we can see that both "ST-SAGE-TA" and "ST-SAGE-LS" exhibit highly desirable scaling characteristics which are linear to the number of available spatial items. On the other hand, the result also demonstrates that "ST-SAGE-TA" is much faster than "ST-SAGE-LS" (30ms vs. 268ms) when the number of available items is 2.5 million (improvement by a factor of 8.93).

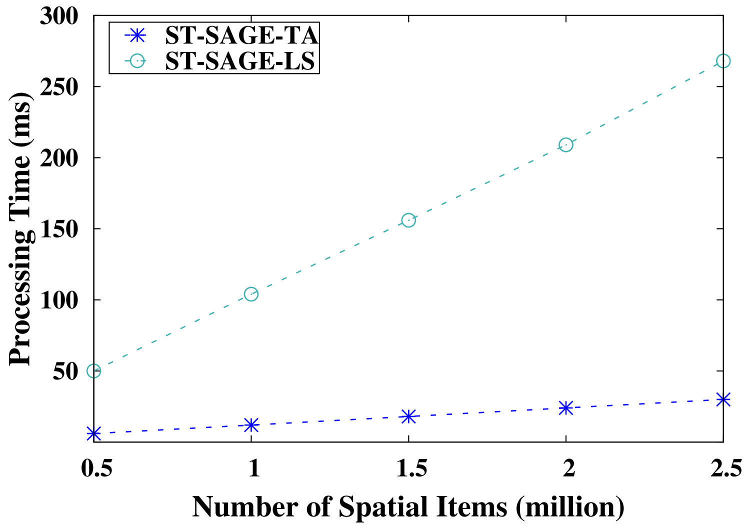


FIGURE 5.3: Recommendations Efficiency using TA on the synthetic dataset

5.3 ALSH-Based Query Processing Technique

5.3.1 Motivation

In reality, the users may be more interested in a quicker response with an approximate top-k result. On the other hand, the Threshold-based algorithm needs to maintain and access K sorted lists of items and frequently update the threshold, which makes it slow when K is large.

5.3.2 Algorithm Description

Locality Sensitive Hashing (LSH) [15] based techniques are common and successful in industrial practice for solving the KNN problem efficiently. Both the running time and the accuracy guarantee of LSH based KNN are in a way independent of the dimensionality of the data. Furthermore, LSH is massively parallelizable, which makes it ideal for large modern datasets. Although LSH is popular in both Euclidean distance and Cosine similarity, there are few work extending LSH to the Maximum Inner-Product Search (MIPS). LSH involves constructing hashing functions h such that the probability of $h(\vec{q}) = h(\vec{v})$ is equivalent to the similarity between the query $\vec{q}$ and the item $\vec{v}$, denoted as $S(\vec{q}, \vec{v})$. For any similarity function to admit a locality sensitive hash function family, the distance function (e.g., $D(\vec{q}, \vec{v}) = 1 - S(\vec{q}, \vec{v})$) must satisfy the triangle inequality [59]. However, inner-product similarity does not satisfy this condition. Thus, LSH cannot be directly applied to MIPS.

Inspired by [64], we apply two different hash functions to the spatial items and the queries respectively (i.e., h_1 for each spatial item $\vec{v}$ and h_2 for each query $\vec{q}$), which is called asymmetric LSH

ALGORITHM 6: The Algorithm of ALSH

Input: all the spatial items V and a given query $\vec{q}$ (both $\vec{q}$ and each item $\vec{v}$ are represented by a vector over K dimensions);

Output: k spatial items with largest S in Equation 4.19;

Preprocessing;

Scale each $\vec{v} \in V$ to have $||\vec{v}||_2 \leq I < 1$;

Append m scalars to each $\vec{v}$ as: $h_1(\vec{v}) = [\vec{v}; ||\vec{v}||_2^2; ||\vec{v}||_2^4; \ldots; ||\vec{v}||_2^{2^m}]$;

Use hash function 5.2 to create hash tables for V;

Querying;

Append m 0.5 to the query $\vec{q}$: $h_2(\vec{q}) = [\vec{q}; 0.5; 0.5; \ldots; 0.5]$;

Apply hash function 5.2 on the transformed query to probe buckets to find top-k items ;

Return the found top-k items;

(ALSH). The main idea of ALSH is to transform the MIPS into classic nearest neighbor search by introducing two hashing functions so that the probability of new collision event $h_2(\vec{q}) = h_1(\vec{v})$ satisfies the conditions in the definition of KNN for $S(\vec{q}, \vec{v}) = \vec{q}^T\vec{v}$.

As the application of ALSH in ST-SAGE are similar, we only present how to accelerate the on-line recommendation using ALSH technique in SPORE.

We present the ALSH algorithm in Algorithm 6, where we apply two hash functions $h_1(\vec{v})$ and $h_2(\vec{q})$ to spatial items and queries respectively. In particular, $h_1(\vec{v})$ appends m scalers of the form $||\vec{v}||_2^{2^i}$ at the end of the vector $\vec{v}$, while $h_2(\vec{q})$ simply appends m "0.5" to the end of the vector $\vec{q}$. According to [64], we have

$$\underset{\vec{v} \in \{\vec{v}\}}{\operatorname{argmax}}\, \vec{q}^T\vec{v} \backsimeq \underset{v \in V}{\operatorname{argmin}}\, ||h_2(\vec{q}) - h_1(\vec{v})||_2 \tag{5.1}$$

This operation connects MIPS with approximate nearest neighbor search. Therefore, the LSH can then be applied to solve the problem. For a vector $\vec{x}$, the hash function proposed in [15] is applied in SPORE, as follows.

$$h_{a,b}(x) = \lfloor \frac{\vec{a}^T\vec{x} + b}{r} \rfloor \tag{5.2}$$

where r is a fixed real number. There are three parameters in Algorithm 6: I, m and r. According to empirical analysis in [64], we set $I = 0.83$, $m = 3$. and $r = 2.5$. a is a random vector with each component generated from i.i.d. normal, i.e., $a_i \sim N(0, 1)$, and b is a scalar generated uniformly at random from $[0, r]$.

Methods	Online Recommendation Time Cost (ms)				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
ALSH	2.28	2.46	2.74	3.16	3.4
TA	4.64	5.71	7.21	8.32	9.23
LS	20.20	21.26	22.46	23.11	24.12

TABLE 5.1: Recommendation Efficiency using ALSH on Foursquare Dataset

Bound Analysis. Inspired by the bound analysis of L2LSH in [15], we can conclude that:

1): if $S(\vec{q}, \vec{v}) \geqslant S_0$, then $Pr(h_{a,b}(h_1(\vec{v}) = h_2(\vec{q}))) \geqslant F_r(\sqrt{1 + m/4 - 2S_0 + I^{2^{m+1}}})$, which means that the probability that $\vec{v}$ is placed in the same bucket as $\vec{q}$ is larger than a specific fraction;

2): if $S(\vec{q}, \vec{v}) \leqslant cS_0 (0 < c < 1)$, then $Pr(h_{a,b}(h_1(\vec{v}) = h_2(\vec{q}))) \leqslant F_r(\sqrt{1 + m/4 - 2cS_0})$;

where the function F_r is defined in Equation 5.3 and $\Phi(x)$ is the cumulative density function of standard normal distribution. $S_0 > 0$ and it corresponds to the constant in S_0-near neighbor search of query $\vec{q}$ in LSH.

$$F_r(d) = 1 - 2\Phi(-r/d) - \frac{2}{\sqrt{2\pi}(r/d)}(1 - e^{-(r/d)^2/2}) \tag{5.3}$$

In this way, we can construct data structures with worst case $O(n^\rho \log n)$ query time guarantees, where ρ is computed with Equation 5.4. According to the analysis in [64], for any given $c < 1$, there always exist $I < 1$ and m such that $\rho < 1$. This way, we obtain a sublinear query time algorithm.

$$\rho = \frac{\log F_r(\sqrt{1 + m/4 - 2S_0 + I^{2^{m+1}}})}{\log F_r(\sqrt{1 + m/4 - 2cS_0})} \tag{5.4}$$

5.3.3 Experiments

This experiment is to evaluate the efficiency of our proposed online recommendation algorithm ALSH on both the real-life and large-scale synthetic datasets which are the same as the datasets used in Section 5.2.4. We compare ALSH with two algorithms. The first algorithm is the threshold algorithm (TA) developed for online recommendation in Section 5.2. The other algorithm is the linear-scanning method (LS) that linearly scans all spatial items by computing a ranking score for each item according to Equation 4.19 and selects top-k ones with highest ranking scores. All the online recommendation algorithms were implemented in Java (JDK 1.7) and ran on a Windows Server 2012 with 256G RAM.

Table 5.1 presents the average online efficiency of the three different methods on the Foursquare dataset. On average, our proposed ALSH produces top-10 recommendations in 2.74ms. From these

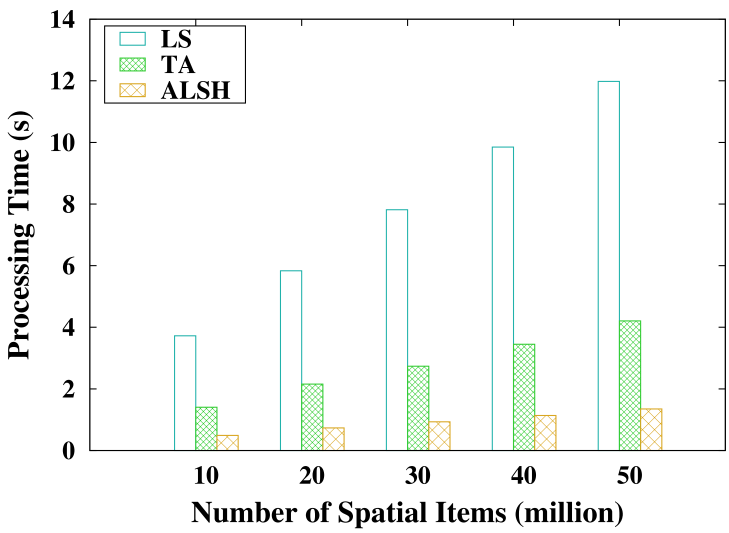


FIGURE 5.4: Recommendation Efficiency using ALSH on the synthetic dataset

results, we observe that 1) ALSH outperforms LS and TA significantly, which demonstrates that ALSH-based query processing technique is much more efficient; 2) the time costs of all methods increase with the increasing number of recommendations (k).

To further evaluate the scalability of SPORE, another experiment is conducted on the synthetic dataset. We vary the number of candidate spatial items from 10 million to 50 million based on the fact that, given a query and the number of recommendations (k), the efficiency of producing online recommendation mainly depends on the number of the available spatial items. Figure 5.4 presents the time costs for producing top-10 recommendations by varying number of available items from 10 millions to 50 millions. From Figure 5.4, we can see that ALSH reduces the processing time for each online query significantly compared with both LS (1.35s vs 11.98s) and TA (1.35s vs 4.21s) when the number of spatial items is 50 million. This improvement is crucial to enhancing the online users' experience in the real-life scenario where the number of spatial items is very large.

Discussion about the Accuracy of ALSH: LS returns the exact top-k spatial items with highest ranking scores. However, ALSH finds the approximate top-k spatial items with highest ranking scores. To evaluate the accuracy of ALSH in making recommendations, we compared the $Accuracy@k$ values of ALSH with the ones of LS. Figure 5.5 presents comparison results on the Foursquare dataset. From Table 5.1 and Figure 5.5, we can see that ALSH reduces the online query time significantly (about 87.80%) at the cost of a minor accuracy lose (8.69%) in producing top-10 recommendations, compared with LS.

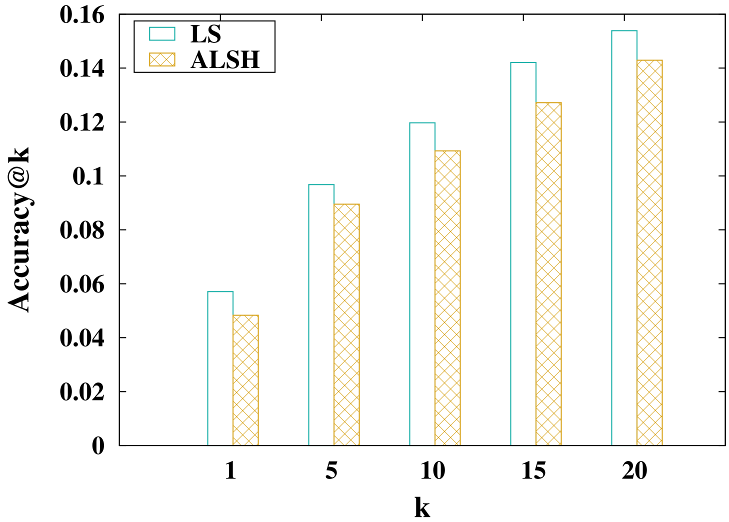


FIGURE 5.5: Recommendation Accuracy of ALSH and LS on Foursquare Dataset

5.4 Conclusion

In this chapter, we proposed two methods to accelerate the online recommendation, motivated by the fact that the online computation in both models proposed in Chapter 3 and 4 is inefficient when the number of spatial items is large. Both methods aim at reducing the number of POIs to be scanned in online recommendation. To prune the item search space, we first proposed an instance optimal algorithm: the Threshold based algorithm (TA). This algorithm is able to return the exact top-k items by scanning the minimum number of spatial items. Considering that, in reality, the users may be more interested in a quicker response with an approximate top-k result, we also demonstrated an approximate algorithm based on TA. As the Threshold based algorithm needs to maintain and access K (K is the number of topics) sorted lists of items and frequently update the threshold, which makes it slow especially when K is large, we proposed another algorithm based on Locality Sensitive Hashing (LSH). This new algorithm is called as asymmetric LSH (ALSH). The main idea of ALSH is that it transform the Maximum Inner-Product Search into classic nearest neighbor search by introducing two hashing functions to the spatial items and queries respectively.

Chapter 6

Conclusion and Future Work

We mainly focused on how to build effective models for POI recommendation in LBSNs in two cases:

- In Chapter 3, we proposed a spatial-temporal sparse additive generative model, ST-SAGE, to cope with the problem in out-of-town spatial item recommendation, which effectively overcomes the challenges arising from travel locality and spatial-temporal dynamics of user behavior. Specifically, to combat travel locality, ST-SAGE exploits both the co-occurrence pattern of spatial items and their content to infer and transfer user interests. To address spatial dynamics of user behavior, ST-SAGE incorporates the native or tourist preference at the target location. To combat the data sparsity in modeling the temporal dynamics of user behavior, we enhanced the model with smoothing by taking advantage of the temporal preferences of the other similar users. To alleviate the data sparsity confronted by the inference of native preference and tourist preference for each region, ST-SAGE employs an additive framework to smooth the preferences over a well-designed spatial pyramid. Besides, we developed a scalable and parallel learning algorithm for ST-SAGE based on the GraphLab to improve the efficiency of the model training. We conducted extensive experiments to evaluate the performance of our ST-SAGE model on two real datasets. The experimental results reveal the advantages of ST-SAGE over other spatial item recommendation methods, for both out-of-town and home-town recommendations in terms of both recommendation effectiveness and efficiency, which demonstrate the effectiveness of ST-SAGE in facilitating travel for users in their home towns as well as in regions they are not familiar with.

- In Chapter 4, we proposed a novel sequential personalized spatial item recommendation framework (*SPORE*). To effectively overcome the challenges arising from *low-sampling rate* and *huge prediction space*, SPORE introduces a novel latent variable *topic-region* to model and fuse the sequential influence and personal interests in the latent space. A topic-region corresponds to both a semantic topic (i.e., a soft cluster of words) and a geographical region (i.e., a soft cluster of locations). The advantages of modeling sequential effects at the topic-region level include a significantly reduced prediction space, an effective alleviation of data sparsity and a direct expression of the semantic meaning of users' spatial activities. To seamlessly fuse sequential effects and personal interests in a unified and principled way, we adopted the sparse additive modeling technique to add them to exponential space thus avoiding the inference of mixture weights for each factor. Extensive experiments were conducted to evaluate the performance of SPORE on two real datasets. The results demonstrate the advantages of SPORE.

To accelerate the on-line recommendation process in both cases, we proposed two on-line query processing techniques.

- TA. TA algorithm is able to find the exact top-k spatial items without computing ranking scores for all spatial items. Additionally, this algorithm is instance optimal, scanning the minimum number of spatial items, and there is no other deterministic algorithm that has a lower optimality ratio.

- ALSH. In reality, the users may be more interested in a quicker response with an approximate top-k result. On the other hand, the TA algorithm needs to maintain and access K sorted lists of items and frequently update the threshold, which makes it slow when K is large. Thus, we designed an asymmetric Locality Sensitive Hashing (ALSH) technique to speed up online top-k recommendations by extending the traditional LSH to the Maximum Inner-Product Search (MIPS) by applying two different hash functions to the spatial items and the queries respectively.

- We have evaluated the two techniques on one real-life dataset and one large scale synthetic dataset. The results show the superiority of these accelerating methods in on-line recommendation.

Future work could adapt these models to the stream data or concentrate on making recommendation for the cold-start users.

- Stream Data. All these models we proposed are static, which means that they are one-time trained and then used to predict the users' behavior. However, as we all know, the social data is generated all the time. For example, about half a billion tweets are generated every day [52]. Thus it's essential that these models are able to adapt to the new generated data dynamically. In recent years, there are increasing literatures exploiting stream settings in both general recommender systems [6] and social media [54, 73]. However, to the best of our knowledge, there is very few work studying POI recommendation in stream settings on location-based social networks.

- Cold-Start Users. All the models in this thesis aim at providing recommendations for the users with history activities, which are known as warm-start users. However, recommendation for the cold-start users are crucial as providing them with a good initial experience is essential to growing the user base of the systems. This problem is more acute for newly-launched recommender systems. Even for relatively mature recommender systems, a vast majority of the customers are actually cold-start users and the participation frequency of customers follows a power-law distribution [99]. Recommendation for cold-start users is challenging as traditional methods, such as the well-known collaborative filtering (CF) approaches, would fail for these users, because the system knows very little about the target users' preferences.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, June 2005.

[2] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. G. Pueyo, and J. Yuan. Latent factor models with additive and hierarchically-smoothed user preferences. In *WSDM*, pages 385–394, 2013.

[3] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL*, pages 199–208, 2012.

[4] J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel. A survey on recommendations in location-based social networks. *GeoInformatica*, 2014.

[5] A. Beeharee and A. Steed. Exploiting real world knowledge in ubiquitous applications. *Personal Ubiquitous Comput.*, 11(6):429–437, 2007.

[6] S. Chang, Y. Zhang, J. Tang, D. Yin, Y. Chang, M. A. Hasegawa-Johnson, and T. S. Huang. Streaming recommender systems. In *WWW*, pages 381–389, 2017.

[7] W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang. Collaborative filtering for orkut communities: Discovery of user latent behavior. In *Proceedings of the 18th International Conference on World Wide Web*, WWW, pages 681–690, 2009.

[8] C. Cheng, H. Yang, I. King, and M. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, pages 17–23, 2012.

[9] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.

[10] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, pages 2605–2611, 2013.

[11] H. Cheng, J. Ye, and Z. Zhu. What's your next move: User activity prediction in location-based social networks. In *SDM*, pages 171–179, 2013.

[12] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

[13] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[14] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys, pages 39–46, 2010.

[15] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, pages 253–262, 2004.

[16] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *ICML*, 2011.

[17] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS, pages 102–113, 2001.

[18] G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *CIKM*, pages 721–726, 2013.

[19] J. L. Finkel, R. A.and Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

[20] S. Gale and G. Olsson. *Philosophy in geography*. Dordrecht ; Boston : D. Reidel Pub. Co., 1979.

[21] H. Gao and H. Liu. Data analysis on location-based social networks. In *Mobile Social Networking*, pages 165–194, 2014.

[22] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, pages 93–100, 2013.

[23] H. Gao, J. Tang, X. Hu, and H. Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, 2015.

[24] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *AAAI*, 2012.

[25] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware travel tour recommendation. In *KDD*, pages 983–991, 2011.

[26] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. An energy-efficient mobile recommender system. In *KDD*, pages 899–908, 2010.

[27] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237, 1999.

[28] T. Horozov, N. Narasimhan, and V. Vasudevan. Using location for personalized poi recommendations in mobile environments. In *SAINT*, pages 124–129, 2006.

[29] H.-P. Hsieh, C.-T. Li, and S.-D. Lin. Measuring and recommending time-sensitive routes from location-based data. *TIST*, 5(3):45:1–45:27, 2014.

[30] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32, 2013.

[31] B. Hu and E. M. Social topic modeling for point-of-interest recommendation in location-based social networks. In *ICDM*, pages 845–850, 2014.

[32] Z. Hu, J. Yao, B. Cui, and E. Xing. Community level diffusion extraction. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD, pages 1555–1569, 2015.

[33] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, J. Yuan, and L. Garcia-Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *VLDB*, 5(10):956–967, June 2012.

[34] E.-y. Kang, H. Kim, and J. Cho. Personalization method for tourist point of interest (poi) recommendation. 4251:392–400, 2006.

[35] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[36] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *CIKM*, pages 579–588, 2010.

[37] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A mobility model for human walks. INFOCOM, pages 855–863, 2009.

[38] K. W.-T. Leung, D. L. Lee, and W.-C. Lee. Clr: A collaborative location recommendation framework based on co-clustering. In *SIGIR*, pages 305–314, 2011.

[39] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *ICDE*, pages 450–461, 2012.

[40] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *GIS*, pages 34:1–34:10, 2008.

[41] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, pages 1023–1031, 2012.

[42] 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 *KDD*, pages 831–840, 2014.

[43] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, KDD, pages 1043–1051, 2013.

[44] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, pages 396–404, 2013.

[45] B. Liu, H. XIONG, S. Papadimitriou, Y. Fu, and Z. Yao. A general geographical probabilistic factor model for point of interest recommendation. *TKDE*, (99), 2014.

[46] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528, Dec. 1989.

[47] H. Liu, J. Caverlee, G. Xue, and J. Ye. Personalized poi recommendation on location-based social networks. 2014.

[48] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong. Personalized travel package recommendation. In *ICDM*, pages 407–416, 2011.

[49] T. Liu, A. W. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. In *MIT Press*, pages 825–832, 2004.

[50] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, 2012.

[51] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[52] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, pages 1155–1158, 2010.

[53] D. Mok, B. Wellman, and J. Carrasco. Does distance matter in the age of the internet? *Urban Studies*, 47(13):2747–2783, 2010.

[54] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming API with twitter's firehose. In *ICWSM*, 2013.

[55] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *ICDM*, pages 1038–1043, 2012.

[56] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, ICWSM, 2011.

[57] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *iTrust*, pages 224–239, 2005.

[58] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *UIC'07*, pages 1130–1139, 2007.

[59] P. Ram and A. G. Gray. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 931–939, 2012.

[60] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[61] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. In *ICWSM*, 2011.

[62] M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l1-regularization paths. In *AAAI*, pages 1278–1283, 2007.

[63] S. Shekhar, P. Zhang, Y. Huang, and R. Vatsavai. *Data Mining: Next Generation Challenges and Future Directions*. AAAI/MIT Press, 2004.

[64] A. Shrivastava and P. Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *NIPS*, pages 2321–2329. Curran Associates, Inc., 2014.

[65] R. Simon and P. Fröhlich. A mobile application framework for the geospatial web. In *WWW*, pages 381–390, 2007.

[66] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[67] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM*, volume 2, pages 1414–1424 vol.2, 2004.

[68] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009.

[69] Y. Takeuchi and M. Sugimoto. Cityvoyager: An outdoor recommendation system based on user location history. In *UIC*, pages 625–636, 2006.

[70] H. M. Wallach. Topic modeling: Beyond bag-of-words. In *ICML*, pages 977–984, 2006.

[71] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *KDD*, pages 1100–1108, 2011.

[72] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou. Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1255–1264, 2015.

[73] D. Yang, B. Li, and P. Cudré-Mauroux. Poisketch: Semantic place labeling over user activity streams. In *IJCAI*, pages 2697–2703, 2016.

[74] D. Yang, D. Zhang, Z. Yu, and Z. Wang. A sentiment-enhanced personalized location recommendation system. In *HT*, pages 119–128, 2013.

[75] M. Ye, K. Janowicz, C. Mülligann, and W.-C. Lee. What you are is when you are: The temporal dimension of feature types in location-based social networks. In *GIS*, pages 102–111, 2011.

[76] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *GIS*, pages 458–461, 2010.

[77] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[78] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *MDM*, pages 1–10, 2009.

[79] H. Yin and B. Cui. *Spatio-Temporal Recommendation in Social Media*. Springer Briefs in Computer Science. Springer, 2016.

[80] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen. Lcars: A spatial item recommender system. *TOIS*, 32(3):11:1–11:37, 2014.

[81] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *KDD*, pages 221–229, 2013.

[82] H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *the 24th ACM International Conference on Information and Knowledge Management*, CIKM, pages 1631–1640, 2015.

[83] Z. Yin, L. Gao, J. Han, J. Luo, and T. S. Huang. Diversified trajectory pattern ranking in geo-tagged social media. In *SIAM*, pages 980–991, 2011.

[84] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.

[85] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei. Evaluating geo-social influence in location-based social networks. In *CIKM*, pages 1442–1451, 2012.

[86] J.-D. Zhang and C.-Y. Chow. igslr: Personalized geo-social location recommendation: A kernel density estimation approach. In *SIGSPATIAL*, pages 334–343, 2013.

[87] J.-D. Zhang and C.-Y. Chow. Spatiotemporal sequential influence modeling for location recommendations: A gravity-based approach. *TIST*, 2015.

[88] J.-D. Zhang, C.-Y. Chow, and Y. Li. Lore: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*, pages 103–112, 2014.

[89] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *ECIR*, pages 338–349, 2011.

[90] B. Zheng, K. Zheng, M. Sharaf, X. Zhou, and S. Sadiq. Efficient retrieval of top-k most similar users from travel smart card data. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 1 of *MDM*, pages 259–268, 2014.

[91] K. Zheng, S. Shang, N. Yuan, and Y. Yang. Towards efficient search for activity trajectories. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 230–241, 2013.

[92] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038, 2010.

[93] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI, pages 236–241, 2010.

[94] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Towards mobile intelligence: Learning from gps history data for collaborative recommendation. *Artif. Intell.*, 184-185:17–37, 2012.

[95] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Trans. Intell. Syst. Technol.*, 2(1):2:1–2:29, 2011.

[96] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Trans. Web*, 5(1):5:1–5:44, 2011.

[97] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.

[98] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua. Mining travel patterns from geotagged photos. *TIST*, 3(3):56:1–56:18, 2012.

[99] K. Zhou, S. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *SIGIR*, pages 315–324, 2011.