

SPORE: A Sequential Personalized Spatial Item Recommender System

Weiying Wang[†] Hongzhi Yin^{†*} Shazia Sadiq[†] Ling Chen[§] Min Xie[‡] Xiaofang Zhou[†]

[†]*University of Queensland, School of Information Technology and Electrical Engineering*

[§]*QCIS, University of Technology, Sydney*

[‡]*Institute of Software, Chinese Academy of Sciences*

[†]{weiyingwang, h.yin1, uqssadiq, uqxzhou}@uq.edu.au [§]ling.chen@uts.edu.au [‡]xiemin@iscas.ac.cn

Abstract—With the rapid development of location-based social networks (LBSNs), spatial item recommendation has become an important way of helping users discover interesting locations to increase their engagement with location-based services. Although human movement exhibits sequential patterns in LBSNs, most current studies on spatial item recommendations do not consider the sequential influence of locations. Leveraging sequential patterns in spatial item 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 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 in a principled manner.

In light of the above challenges, we propose a sequential personalized spatial item 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. Furthermore, we design an asymmetric Locality Sensitive Hashing (ALSH) technique to speed up the online top-*k* recommendation process by extending the traditional LSH. We evaluate the performance of *SPORE* on two real datasets and one large-scale synthetic dataset. The results demonstrate a significant improvement in *SPORE*’s ability to recommend spatial items, in terms of both effectiveness and efficiency, compared with the state-of-the-art methods.

I. INTRODUCTION

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, Brightkite and Loopt, where users can check in at different venues and share life experience in the physical world via mobile devices [1]. A personalized *location recommendation service*, which encourages users to explore new locations [2], is an essential function of LBSNs. Therefore, developing personalized recommender systems for LBSNs to provide users with spatial items (e.g., a venue or an event associated with a geographic location) has recently attracted increased research attention [3].

Existing research on personalized spatial item 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 [3]. Recently, there are works that further integrate social influence in LBSNs to recommend items as common interests shared by social friends [1]. In terms of the temporal effect of user check-in activities in LBSNs, to our knowledge, only the temporal cyclic patterns of check-ins have been investigated [4]. However, it has been observed by recent research on human mobility that, in reality, human movement exhibits sequential patterns, which serve as the basis for mobility prediction [5]. In particular, an analysis has been conducted on three publicly available real-world datasets, Foursquare, Gowalla and Brightkite in [6], which calculates the probabilities of each of the next spatial items immediately visited by a user after visiting a given spatial item. The results show that each selected spatial item 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 spatial items indicates there are underlying sequential patterns between spatial items 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 [6]), geographical proximity (e.g., tourists often sequentially visit London Eye, Big Ben and Downing Street [7]) 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 [8]). Therefore, exploiting sequential patterns is important to improve recommendation in LBSNs.

Nevertheless, leveraging sequential information for spatial item 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 [9], [10]. 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 [9]. Thus, it is difficult to model the dependency between two check-in locations in LBSNs using the location prediction techniques

* This author is the corresponding author.

on GPS trajectories.

Huge prediction space. Sequential recommendation methods have been proposed in the literature [12], [13], [14], most of which are based on Markov chains. Suppose there are a collection of V spatial items 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 n th order Markov chain model, which is extremely computational-expensive. To reduce the size of the prediction space, most related studies [12], [14] 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 spatial item recommendation methods focusing on personalization [1], [2], [5] make recommendations according to users' personal interests, but neglect the sequential orders between spatial items. 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.

In light of the aforementioned challenges, we propose a **Sequential PersOnalized spatial items REcommender** system, called SPORE, which 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) [15]. 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 n th order Markov Chain into linear complexity $|V| \times K$ (K is the number of topic-regions).

To unify personal interests and sequential effect in a principled way, traditional mixture models, such as LCA-LDA [16], combine multiple facets (e.g., personal interests and temporal effect) by introducing additional latent variables that act as "switches", to control which facet is currently active. In reality, it is computationally expensive and difficult to learn these variables accurately, given sparse datasets. Again, SPORE follows the SAGE model [15] to add the effect of personal interests and sequential influence in the exponential space to avoid the inference of latent "switching" variables, with the aim of achieving improved robustness and predictive accuracy.

To support real-time recommendation scenario, we further design an asymmetric Locality Sensitive Hashing (ALSH), extending the classical LSH technique, to significantly reduce the search space and produce top- k recommendations without examining all available spatial items.

The main contributions of our work are summarized as follows.

- We design a novel sequential personalized spatial item 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 an efficient top- k query processing technique to speed up the process of online recommendation 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 SPORE.

The remainder of this paper is organized as follows: Section II presents related works on sequential recommendation and spatial item recommendation. Section III formulates our personalized spatial item recommendation problem. The detailed design of SPORE is presented in Section IV. We deploy SPORE to spatial item recommendation in Section V. Section VI discusses the experimental evaluation of SPORE. Section VII concludes this paper.

II. RELATED WORK

In this section, we discuss existing research related to our work, including sequential recommendation and spatial item

recommendation.

Existing works on sequential recommendation mostly utilize the Markov chain property to predict the next check-ins. Cheng et al. [12] 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 [13]. 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.

There are two main lines of research in spatial item recommendation. The first one focuses on GPS trajectory data. GPS trajectory data usually consists of a small number of users, but has dense location records [17], [18]. The other line of research is conducted on LBSN data, which has a low sampling rate in both space and time compared to GPS trajectories [11], [9]. Many recent studies have shown that there is a strong correlation between user check-in activities and geographical distance as well as social connections [3]. Consequently, most of the current spatial item recommendation methods focus on leveraging geographical and social influences to improve recommendation accuracy. Recently, the temporal effect of user check-in activities in LBSNs has been noted in spatial item recommendation [4], [19], [20]. However, only the temporal cyclic patterns of user check-ins to capture the temporal non-uniformness and temporal consecutiveness have been studied and leveraged in recommending spatial items. While it has been shown in multiple studies that human movement in LBSNs clearly demonstrates sequential patterns [6], [7], [8], leveraging sequential patterns for spatial item recommendation has not been well-studied.

Our work in this paper distinguishes itself from previous research in several aspects. Firstly, to the best of our knowledge, this is the first effort that automatically integrates sequential effect and personalization in a unified model. Secondly, although existing research [13] 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. Thirdly, 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.

Variable	Interpretation
U, V	the set of users and spatial items
W	the vocabulary set
D_u	the profile of user u
S_u	the sequence of user u
$P_{u,t}$	the predecessor set of user u at time t
$v_{u,i}$	the spatial item of i^{th} record in D_u
$W_{v_{u,i}}$	the set of words describing spatial item $v_{u,i}$
$w_{v_{u,i},n}$	the n^{th} word in $W_{v_{u,i}}$
$p_{u,t,n}$	the n^{th} spatial item in $P_{u,t}$
$t_{u,i}$	the time stamp of i^{th} record in D_u

TABLE I. NOTATIONS OF THE INPUT DATA

III. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we formally define the data in LBSNs that we are interested in and formulate the problem accordingly. For ease of presentation, Table I lists the notations.

Definition 1: (Spatial Item) A spatial item is an item associated with a geographical location (e.g., a restaurant or a cinema).

Definition 2: (Sequence) A sequence of user u , denoted by $S_u = \{(v_1, t_1), (v_2, t_2), \dots, (v_n, t_n)\}$, 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 ($1 \leq i \leq n$ and $t_1 \leq t_2 \leq \dots \leq t_n$).

Definition 3: (Predecessor, Successor) Given a sequence $S_u = \{(v_1, t_1), (v_2, t_2), \dots, (v_n, t_n)\}$ and a time period threshold ΔT , if v_i and v_j are two items in this sequence and $0 < t_i - t_j \leq \Delta T$, we say v_j is a **predecessor** of v_i . Conversely, v_i is a **successor** of v_j .

Definition 4: (Predecessor Set) Given a target user u and time t , the predecessor set, denoted as $P_{u,t} = \{v_i | v_i \in S_u, 0 < t - t_i \leq \Delta T\}$, is a set of spatial items visited before t in the given time threshold ΔT .

Following the previous works [13], [6], 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 ΔT , it is assumed that there is no influence between the two items. We will study the impact of ΔT on the quality of spatial item recommendation in Section VI.

Definition 5: (User Activity) A user activity is a five tuple $(u, v, W_v, t, P_{u,t})$, which indicates that the user u visits the spatial item v described as W_v at time t . $P_{u,t}$ is the predecessor set of spatial items that user u has visited before t .

Definition 6: (User Profile) For each user u , we create a user profile D_u , which contains a set of user activities associated with u . Given a set of users U , the dataset D used in our model is a collection of user profiles, $D = \{D_u : u \in U\}$.

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 1: (Spatial Item Recommendation) 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.

As discussed before, traditional spatial item recommendation methods do not exploit the sequential patterns between items but focus on learning user interests. On the other hand, existing sequential recommendation algorithms using Markov chain or association rule mining techniques ignore the personalization, which is the key to the success of a recommender system. In our research, we aim to integrate personal interests and sequential effects into a unified framework. In particular, there are three main tasks involved in our problem.

Task 1: Extracting users’ personal interests. This task models the users’ personal interests.

Task 2: Extracting sequential influence. This task models the influence of visited spatial items in a sequence.

Task 3: Fusing users' personal interests and sequential influence into a unified framework.

All three tasks are very challenging in the context of LBSN data. In **Task 1**, the user-item matrix is very sparse in LBSNs, which makes it difficult for traditional recommendation models (e.g., matrix factorization or collaboration filtering) to accurately infer users’ interests from the data. As analyzed in Section I, **Task 2** faces the severe challenges of low-sampling rate and huge prediction space, which render the classical n th order Markov Chain inefficient because its complexity increases exponentially w.r.t. n (corresponding to $|P_{u,t}|$) [13]. In **Task 3**, no framework exists that simultaneously integrates the two components into a unified model. Zhang et al. combine the sequential influence, geographical influence and social influence in [13]. However, they fuse the three components by simply multiplying them together while we try to unify them in a principled manner rather than an ad-hoc way.

IV. THE SPORE MODEL

In this section, we first present some preliminaries about the SAGE model [15], and then describe our sequential personalized recommendation model based on it.

A. Preliminaries about SAGE

Our model is inspired by the Sparse Additive Generative Model (SAGE) [15], which is an effective generative model without explicit “switching” variables. The basic idea of the model is that, if a variable is affected by several factors, it can be generated by the mixture of these factors without any explicit indicator variable. The key difference between SAGE and traditional mixture models is that the mixture occurs in terms of 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 distinguishing the set of factors.

To provide a clearer explanation of SAGE, we take its application to our **Task 3** as an example. Given a query (u, t) , we retrieve the predecessor set P which contains items visited by u before t . A traditional probabilistic mixture generative model combines the two factors, u 's personal interests θ^{user} and the sequential influence of all the predecessors θ^{seq} , through a linear combination as Equation 1.

$$P(v|\theta_u^{user}, \theta_P^{seq}) = \lambda_u P(v|\theta_u^{user}) + (1 - \lambda_u) P(v|\theta_P^{seq}) \quad (1)$$

where λ_u is the personalized mixture weight (i.e., the “switching” variable) that needs to be inferred for each user. Obviously, it is very difficult to infer the variable accurately when

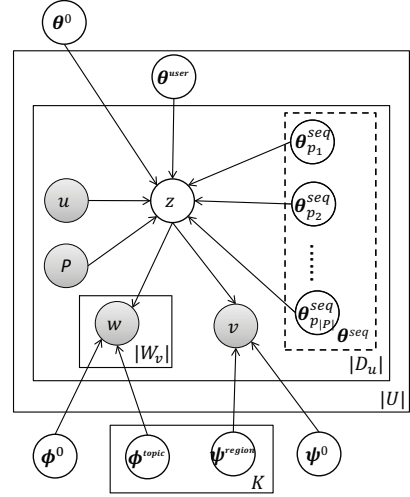


Fig. 1. The Graphical Representation of Our Model

the training data for the individual user is sparse. By contrast, SAGE combines the two generative facets through simple addition in an exponential space as illustrated in Equation 2. Note that SAGE avoids the need for computing latent switching variables.

$$\begin{aligned} P(v|\theta_u^{user}, \theta_P^{seq}) &= P(v|\theta_u^{user} + \theta_P^{seq}) \\ &= \frac{\exp(\theta_{u,v}^{user} + \theta_{P,v}^{seq})}{\sum_{v'} \exp(\theta_{u,v'}^{user} + \theta_{P,v'}^{seq})} \end{aligned} \quad (2)$$

B. 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 t is influenced by three factors: 1) her personal interests θ_u^{user} ; 2) the influence of the items visited before t , $\theta_P^{seq} = \{\theta_{p_1}^{seq}, \theta_{p_2}^{seq}, \dots, \theta_{|P|}^{seq}\}$; 3) the general public's preferences θ^0 . Figure 1 shows the graphical representation of SPORE, and Table II 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 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 [3]. 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 [21], 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 θ_u^{user} . To capture both the semantic and spatial information, a topic-region z in SPORE is associated with a word distribution ϕ_z^{topic} and a distribution over spatial items ψ_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 [22]. 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 ϕ_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 $\theta_{p_1}^{seq}, \theta_{p_2}^{seq}, \dots, \theta_{p_{|P_{u,t_{u,i}}|}}^{seq}$. To reduce the exponential complexity of classical n th-order Markov Chain to the linear complexity, an intuitive solution is to add the influence of each predecessor, inspired by the idea of n th additive Markov Chain in [13], [6]. However, using the n th 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 n th 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 3. 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. In this way, SPORE also reduces the exponential complexity $|V|^{n+1}$ of classical n th order Markov Chain into $|V| \times K$ (n corresponds to $|P_{u,t}|$ in SPORE).

$$P(v|\theta_{P_{u,t_{u,i}}}^{seq}) = \sum_{j=1}^{|P_{u,t_{u,i}}|} \lambda_{u,j} P(v|\theta_{p_j}^{seq}) \quad (3)$$

$$\sum_{j=1}^{|P_{u,t_{u,i}}|} \lambda_{u,j} = 1 \quad (\lambda_{u,j} \geq 0)$$

$$P(v|\theta_{P_{u,t_{u,i}}}^{seq}) = P(v|\theta_{p_1}^{seq} + \theta_{p_2}^{seq} + \dots + \theta_{p_{|P_{u,t_{u,i}}|}}^{seq})$$

$$= \frac{\exp(\sum_{j=1}^{|P_{u,t_{u,i}}|} \theta_{p_j,v}^{seq})}{\sum_{v'} \exp(\sum_{j=1}^{|P_{u,t_{u,i}}|} \theta_{p_j,v'}^{seq})} \quad (4)$$

Variable	Interpretation
K	the number of topic-regions
$z_{u,i}$	the topic-region assigned to spatial item $v_{u,i}$
θ^0	the topic-region distribution of the background
θ_u^{user}	the topic-region distribution, representing the intrinsic interest of user u
$\theta_{p_j,v}^{seq}$	the topic-region distribution of v^{th} spatial item in p_j , representing the influence of the v^{th} item
ϕ_z^{topic}	content word distribution of topic-region z
ψ_z^{region}	spatial item distribution of topic-region z

TABLE II. NOTATIONS OF MODEL PARAMETERS

Public Preference Component. We incorporate the public's preference θ^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, θ^0 plays the role of a background model to make the learned users' interests more discriminative and personalized, since θ^0 captures the common topics among users. Similarly, we introduce background models for words and spatial items as ϕ^0 and ψ^0 , respectively. The background models make the relevant parameters learned for topic-regions more discriminative and meaningful, since ϕ^0 and ψ^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.

C. 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 $z_{u,i}$
 $z_{u,i} \sim P(z_{u,i}|P_{u,t_{u,i}}, \theta^0, \theta_u^{user}, \theta^{seq})$
- For each content word $w_{v_{u,i},n}$ in $W_{v_{u,i}}$, draw
 $w_{v_{u,i},n} \sim P(w_{v_{u,i},n}|\phi^0, z_{u,i}, \phi^{topic})$
- Draw a spatial item $v_{u,i}$
 $v_{u,i} \sim P(v_{u,i}|\psi^0, z_{u,i}, \psi^{region})$

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 5.

$$P(z_{u,i}|P_{u,t_{u,i}}, \theta^0, \theta_u^{user}, \theta^{seq}) = P(z_{u,i}|\theta^0 + \theta_u^{user} + \theta_{P_{u,t_{u,i}}}^{seq}) \quad (5)$$

where $\theta_{P_{u,t_{u,i}}}^{seq}$ is the sum of the influences of all the visited spatial items in $P_{u,t_{u,i}}$. It can be expanded as $\sum_{n=1}^{|P_{u,t_{u,i}}|} \theta_{p_{u,t_{u,i},n}}^{seq}$. Once the topic-region z is generated, the spatial item v and the associated content words are generated as expressed in Equations 6 and 7, respectively.

$$P(v_{u,i}|\psi^0, z_{u,i}, \psi^{region}) = P(v_{u,i}|\psi^0 + \psi_{z_{u,i}}^{region}) \quad (6)$$

$$P(w_{v_{u,i},n}|\phi^0, z_{u,i}, \phi^{topic}) = P(w_{v_{u,i},n}|\phi^0 + \phi_{z_{u,i}}^{topic}) \quad (7)$$

D. Model Inference

Our goal with model inference is to learn the parameters that maximize the marginal log-likelihood of the observed random variables \mathbf{w} and \mathbf{v} . Marginalization is performed with respect to the latent random variable \mathbf{z} . However, it is difficult to achieve maximization directly. Therefore, we apply a mixture between EM and a Monte Carlo sampler, called the Gibbs

$$\alpha_{u,P,z} = \frac{\exp(\theta_z^0 + \theta_{u,z}^{user} + \sum_{i=1}^{|P|} \theta_{P_i,z}^{seq})}{\sum_{zz} \exp(\theta_{zz}^0 + \theta_{u,zz}^{user} + \sum_{i=1}^{|P|} \theta_{P_i,zz}^{seq})}, \beta_{z,w} = \frac{\exp(\phi_w^0 + \phi_{z,w}^{topic})}{\sum_{ww} \exp(\phi_{ww}^0 + \phi_{z,ww}^{topic})}, \gamma_{z,v} = \frac{\exp(\psi_v^0 + \psi_{z,v}^{region})}{\sum_{vv} \exp(\psi_{vv}^0 + \psi_{z,vv}^{region})} \quad (8)$$

$$\begin{aligned} P(\mathbf{z}, \mathbf{w}, \mathbf{v} | \ominus, \mathbf{u}, \mathbf{P}) &= P(\mathbf{z} | \mathbf{u}, \mathbf{P}, \theta^0, \theta^{user}, \theta^{seq}) P(\mathbf{w} | \mathbf{z}, \phi^0, \phi^{topic}) P(\mathbf{v} | \mathbf{z}, \psi^0, \psi^{region}) \\ &= \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \alpha_{u,P_u,t_{u,i},z_{u,i}} \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{v_{u,i},n}} \prod_{u=1}^{|U|} \prod_{i=1}^{|D_u|} \gamma_{z_{u,i},v_{u,i}} \end{aligned} \quad (9)$$

EM algorithm [23], to maximize the complete data likelihood in Equation 9, where \ominus 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 \ominus by fixing all topic-region assignments. The two steps are iterated until convergence.

More specifically, we iteratively draw the latent topic-region \mathbf{z} for all check-in activities in the E-step. According to the Gibbs Sampling, when sampling $z_{u,i}$ as expressed in Equation 10, we assume all other variables are fixed. $\mathbf{z}_{-u,i}$ represents the topic-region assignments for all user activities except the i^{th} activity for user u .

$$\begin{aligned} P(z_{u,i} | \mathbf{z}_{-u,i}, \mathbf{w}, \mathbf{v}, \mathbf{u}, \mathbf{P}, \ominus) \\ \propto \alpha_{u,P_u,t_{u,i},z_{u,i}} \times \prod_{n=1}^{|W_{v_{u,i}}|} \beta_{z_{u,i},w_{v_{u,i},n}} \times \gamma_{z_{u,i},v_{u,i}} \end{aligned} \quad (10)$$

In the M-step, we optimize the parameters \ominus 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) [24], 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 [24] updates for the quasi-Newton method, the gradients of model parameters θ^0 , θ^{user} and θ^{seq} are provided as follows.

$$\frac{\partial L}{\partial \theta_z^0} = d(z) - \sum_{u=1}^{|U|} \sum_{i=1}^{|D_u|} \alpha_{u,P_u,t_{u,i},z} \quad (11)$$

$$\frac{\partial L}{\partial \theta_{u,z}^{user}} = d(u, z) - \sum_{i=1}^{|D_u|} \alpha_{u,P_u,t_{u,i},z} \quad (12)$$

$$\frac{\partial L}{\partial \theta_{v,z}^{seq}} = d(v, z) - \sum_{j=1}^{|D_v|} \alpha_{u_j,P_j,z} \quad (13)$$

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 ϕ^0 , ϕ^{topic} , ψ^0 , and ψ^{region} are computed as follows:

$$\frac{\partial L}{\partial \phi_w^0} = d(w) - \sum_{z=1}^K d(z) \times \beta_{z,w} \quad (14)$$

$$\frac{\partial L}{\partial \phi_{z,w}^{topic}} = d(z, w) - d(z) \times \beta_{z,w} \quad (15)$$

$$\frac{\partial L}{\partial \psi_v^0} = d(v) - \sum_{z=1}^K d(z) \times \gamma_{z,v} \quad (16)$$

$$\frac{\partial L}{\partial \psi_{z,v}^{region}} = d(z, v) - d(z) \times \gamma_{z,v} \quad (17)$$

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 .

V. 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 18. Then, we return the top- k spatial items with the highest probabilities as recommendations.

$$\begin{aligned} P(v, W_v | \ominus, u_q, P_q) &= \sum_{z=1}^K P(v, W_v, z | \ominus, u_q, P_q) \\ &= \sum_{z=1}^K P(z | u_q, P_q, \theta^0, \theta^{user}, \theta^{seq}) P(W_v | z, \phi^0, \phi^{topic}) P(v | z, \psi^0, \psi^{region}) \\ &= \sum_{z=1}^K \alpha_{u_q, P_q, z} \left(\prod_{n=1}^{|W_v|} \beta_{z, w_{v,n}} \right)^{\frac{1}{|W_v|}} \gamma_{z,v} \end{aligned} \quad (18)$$

In Equation 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 19 which separates the offline computation from the online calculation to the maximum extent.

$$\begin{aligned} S(q, v) &= \sum_{z=1}^K F(z, v) W(q, z) \\ F(z, v) &= \left(\prod_{n=1}^{|W_v|} \beta_{z, w_{v,n}} \right)^{\frac{1}{|W_v|}} \times \gamma_{z,v} \\ W(q, z) &= \alpha_{u_q, P_q, z} \end{aligned} \quad (19)$$

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.

A. Efficient Top- k Recommendation

Although the time cost of online recommendation for each query is largely reduced by the offline pre-computation, the online computation is still inefficient when there are a large number of spatial items. This is because the online computation needs to calculate a ranking score for every single candidate item with respect to the query. Thus, we aim to further improve the efficiency of top- k recommendation by reducing the search space.

One straightforward solution for pruning the item search space is to index spatial items using a tree structure such as R-Tree or Metric Tree, which is widely used in nearest-neighbor search problem in metric spaces. However, according to Equation 19, we transform both queries and items into K -dimensional vectors \vec{q} and \vec{v} (with $W(q, z)$ and $F(z, v)$ on each dimension respectively) and measure the similarities between \vec{q} and \vec{v} using the inner-product, which is different from distance functions in metric spaces (e.g., Euclidean distance and Cosine similarity). For instance, the inner-product lack the basic property of *coincidence*. In particular, the Euclidean distance of a point to itself is 0 while the inner-product of a point \vec{v} to itself is $\|\vec{v}\|^2$, which may be high or low depending on the length of \vec{v} . Therefore, it is infeasible to directly apply the techniques for nearest-neighbor search in our problem.

Ram et al. proposed a technique for maximum inner-product search in [25] based on an adapted metric tree and Yin et al. proposed a solution by extending the Threshold-based Algorithm (TA) in [16], [26]. However, these two solutions suffer from the curse of the dimension K . According to the analysis in [27], the efficiency of the techniques based on a tree index structure is $O(K^{12})$. 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) [28] 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 [25]. However, if similarity is measured using inner-product, the distance measure does not satisfy the condition. That is, LSH cannot be directly applied to MIPS.

Algorithm 1: 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 19;

1 Preprocessing:

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

3 Append m scalars to each \vec{v} as:

$$h_1(\vec{v}) = [\vec{v}; \|\vec{v}\|_2^2; \|\vec{v}\|_2^4; \dots; \|\vec{v}\|_2^{2m}] ;$$

4 Use hash function 21 to create hash tables for V ;

5 Querying:

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

7 Apply hash function 21 on the transformed query to probe buckets to find top- k items ;

8 Return the found top- k items;

Inspired by [29], 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 (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 $S(\vec{q}, \vec{v}) = \vec{q}^T \vec{v}$.

We present the ALSH algorithm in Algorithm 1, 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 scalars 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 [29], we have

$$\arg \max_{\vec{v} \in \{\vec{v}\}} \vec{q}^T \vec{v} \simeq \arg \min_{\vec{v} \in V} \|h_2(\vec{q}) - h_1(\vec{v})\|_2 \quad (20)$$

This operation connects MIPS with approximate near neighbor search. Therefore, the LSH can then be applied to solve the problem. For a vector \vec{x} , the hash function proposed in [28] is applied in SPORE, as follows.

$$h_{a,b}(x) = \lfloor \frac{\vec{a}^T \vec{x} + b}{r} \rfloor \quad (21)$$

where r is a fixed real number. There are three parameters in Algorithm 1: I , m and r . According to empirical analysis in [29], 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 $[0, r]$.

Bound Analysis. Inspired by the bound analysis of L2LSH in [28], we can conclude that:

- 1): if $S(\vec{q}, \vec{v}) \geq S_0$, then $Pr(h_{a,b}(h_1(\vec{v})) = h_2(\vec{q})) \geq F_r(\sqrt{1 + m/4 - 2S_0 + I^{2m+1}})$, 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 $S(\vec{q}, \vec{v}) \leq cS_0$ ($0 < c < 1$), then $Pr(h_{a,b}(h_1(\vec{v})) = h_2(\vec{q})) \leq F_r(\sqrt{1 + m/4 - 2cS_0})$;

where the function F_r is defined in Equation 22 and $\Phi(x)$ is the cumulative density function of standard normal distribution.

	Foursquare	Twitter	Synthetic
# of users	4,163	114,508	1.72 million
# of items	121,142	62,462	50 million
# of check-ins	483,813	1,434,668	200 million
time span	12/2009–07/2013	09/2010–01/2011	12/2009–07/2013

TABLE III. STATISTICS OF THE THREE DATASETS

$S_0 > 0$ and it corresponds to the constant in S_0 -near neighbor search of query \vec{q} in LSH.

$$F_r(d) = 1 - 2\Phi(-r/d) - \frac{2}{\sqrt{2\pi}(r/d)}(1 - e^{-(r/d)^2/2}) \quad (22)$$

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 23. According to the analysis in [29], 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.

$$\rho = \frac{\log F_r(\sqrt{1 + m/4 - 2S_0 + I^{2^{m+1}}})}{\log F_r(\sqrt{1 + m/4 - 2cS_0})} \quad (23)$$

VI. 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.

A. Experimental Settings

1) *Datasets*: We conducted our experiments on two real datasets (Foursquare and Twitter) and one large synthetic dataset. The basic statistics of them are shown in Table III. The two real datasets are publicly available¹.

Foursquare. This dataset contains the check-in history of 4,163 users who live in California, USA, between Dec 2009 and Jul 2013. Each check-in activity contains the *user-ID*, *item-ID*, *item-location*, *item-content* and a *check-in time*.

Twitter. This dataset is based on the publicly available Twitter dataset in [30] which does not contain the category and tag information about spatial items. Twitter supports third-party location sharing services like Foursquare and Gowalla (where users of these services choose to share their check-ins on Twitter). Thus, we crawled the content information from Foursquare with the publicly available API². Each record has the same format as the Foursquare dataset.

Synthetic Dataset. To evaluate the online recommendation efficiency of ALSH when the number of spatial items is large, a large synthetic dataset was created following the distribution characteristics of the Foursquare dataset. There are 4,163 users, 121,142 spatial items and 483,813 check-ins on the Foursquare dataset. Approximately, every user has 116 check-ins and every spatial item is associated with 4 check-ins on average. To mimic the data sparsity of the real data, we generate a synthetic dataset with 200 million check-ins, 50 million spatial items and 1.72 million users.

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 [12] and category based MC [11]. As the complexity of classical n th 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 [13] 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 dataset.

GT-BNMF: GT-BNMF [31] is a Geographical-Topical Bayesian non-negative Matrix Factorization framework, which integrates all the information except the sequential information, such as geographical influence, personal interest, popularity effect and content effect in a joint manner.

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** is the simplified version which unifies personal interests with only the sequential influence of the latest visited items.

To evaluate the online recommendation efficiency of ALSH in SPORE, we compared it with two baseline methods. The first method is the linear-scan (LS) algorithm without any accelerating scheme. The second one is the threshold algorithm (TA) [26]. This algorithm is able to find the exact top- k results without scanning all spatial items.

3) *Evaluation Methods*: Given a user profile, namely a collection of user activities, we first extracted the activity

¹<https://sites.google.com/site/dbhongzhi/>

²<http://developer.foursquare.com>

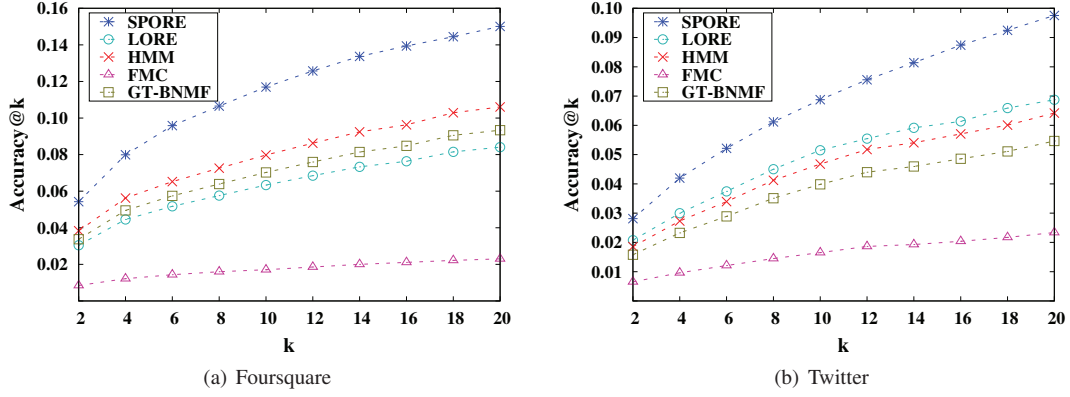


Fig. 2. Performance on Foursquare and Twitter Datasets

sequence of each user. Then, we used 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 adopted the evaluation methodology and measurement $Accuracy@k$ proposed in [21], [32]. Specifically, for each test case (u, v, W_v, t, P) in D_{test} as well as its corresponding query q :

- (1) We computed the ranking score for item v and all other unvisited spatial items.
- (2) We formed a ranked list by ordering all of these items according to their ranking scores. Let p denote the position of v within this list. The best result is that v precedes all the other unvisited spatial items which means that $p = 1$.
- (3) We formed a top- k recommendation list by picking the k top ranked items from the list. If $p \leq k$ (i.e., the ground truth item v appears in the top- k recommendation results), we have a hit. Otherwise, we have a miss.

$Accuracy@k$ is computed as shown in Equation 24. We define $hit@k$ for a single test case as either the value 1, if we have a hit, or the value 0 if we have a miss. The overall $Accuracy@k$ is defined by averaging over all test cases. $\#hit@k$ denotes the number of hits in the test set and $|D_{test}|$ is the number of all test cases.

$$Accuracy@k = \frac{\#hit@k}{|D_{test}|} \quad (24)$$

Note that, 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 low accuracy values [33]. 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, the low accuracy values are common and reasonable. In this paper, we focus on the relative improvements we achieved, instead of the absolute values.

B. 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 (ΔT) 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 2(a) and 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 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

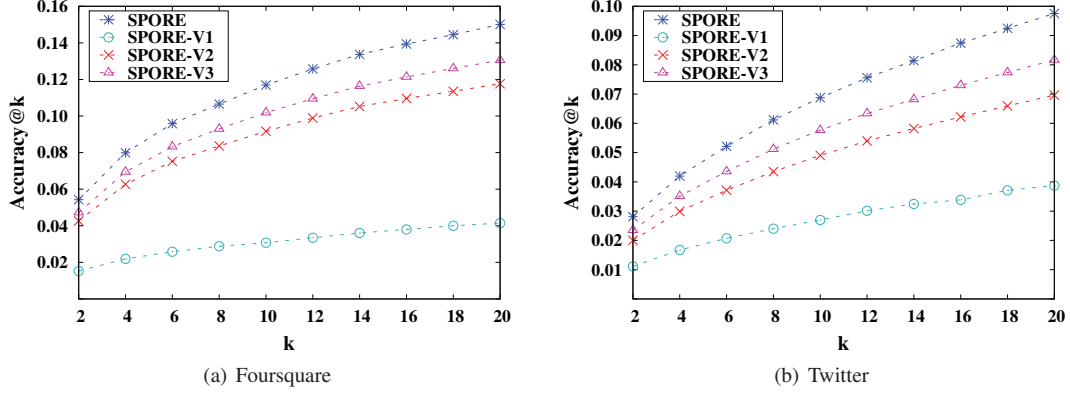


Fig. 3. Impact of Different Factors on Both Datasets

considering sequential influence in personalized spatial item recommendation.

C. 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 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 ΔT .

The results of comparing SPORE with the three variant versions on both datasets are shown in Figure 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. What is also worth noting is 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.

ΔT	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 IV. IMPACT OF PARAMETERS

To study the impact of the two parameters in SPORE, i.e. K and ΔT , 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 ΔT from 0.2 days to 0.7 days. The results are presented in Table IV. 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 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 ΔT becomes larger according to Definition 3. Thus, the performance improves with the increase of ΔT . However, when ΔT is larger than a threshold (e.g., $\Delta T = 0.5$ day), the added predecessors have little influence on the recommendation performance. As a result, we chose $K = 100, \Delta T = 0.5$ day as the best trade-off between accuracy and efficiency on the Twitter dataset.

D. Recommendation Efficiency

This experiment is to evaluate the efficiency of our proposed online recommendation algorithm ALSH on both the real-life and large-scale synthetic datasets. We compared ALSH with two algorithms. The first algorithm is the threshold algorithm (TA) [26] developed for online recommendation. It pre-computes a sorted list for each dimension z (i.e., $F(z, v)$), and also maintains a priority queue of the K sorted lists that controls which sorted list to access in the next. The algorithm has the nice property of terminating early without scanning all spatial items. Specifically, it terminates when the ranking score of the k -th item in the result list is higher than the threshold score. 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 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 V 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 this 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

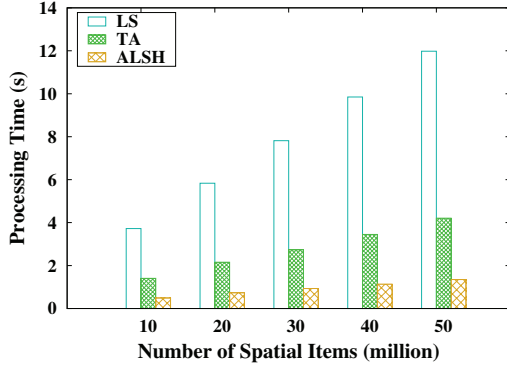


Fig. 4. Scalability of SPORE on the synthetic dataset

costs of all methods increase with the increasing number of recommendations (k).

Methods	Online Recommendation Time Cost (<i>ms</i>)				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
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 V. RECOMMENDATION EFFICIENCY ON FOURSQUARE DATASET

To further evaluate the scalability of SPORE, another experiment was conducted on the synthetic dataset. We varied 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 4 presents the time costs for producing top-10 recommendations by varying number of available items from 10 millions to 50 millions. From Figure 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 presents comparison results on the Foursquare dataset. From Table V and Figure 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.

E. 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 $P(w|z, \phi^{topic})$ for each topic-region z . Based on the top words and their corresponding generation probabilities,

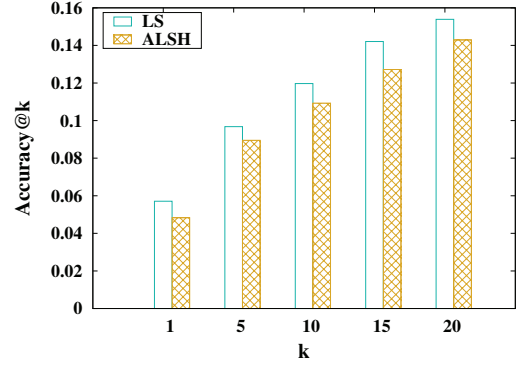


Fig. 5. Recommendation Accuracy of ALSH and LS on Foursquare Dataset

we employ wordle³ 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 $P(v|z, \psi^{region})$ on Google Maps for each topic-region z . We present four example topic-regions in Figure 6.

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 6(c) and 6(d) are both located around Los Angeles. However, they focus on different subjects. The topic-region in Figure 6(c) mainly focuses on the tourist hot spots in Los Angeles, such as the universities, Beverly Hills and Hollywood while the one in Figure 6(d) mainly focuses on the food and nightlife related spots. Thus, the topic-regions discovered by SPORE can be interpreted both semantically and spatially.

VII. CONCLUSION

In this paper, 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 adopt the sparse additive modeling technique to add them to exponential space thus avoiding the inference of mixture weights for each factor. Further, we designed an asymmetric Locality Sensitive Hashing (ALSH) technique to speed up online top- k recommendations by extending the traditional LSH. Extensive experiments were conducted to evaluate the performance of SPORE on two real datasets and one large-scale synthetic dataset. The results demonstrate the advantages of SPORE in terms of both recommendation effectiveness and efficiency.

VIII. ACKNOWLEDGEMENT

This work is partially supported by ARC Discovery Early Career Researcher Award (DE160100308), National Ba-

³<http://www.wordle.net/>



REFERENCES

- [1] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *SIGSPATIAL*, 2012, pp. 199–208.
- [2] G. Ference, M. Ye, and W.-C. Lee, "Location recommendation for out-of-town users in location-based social networks," in *CIKM*, 2013, pp. 721–726.
- [3] 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*, 2014, pp. 831–840.
- [4] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *RecSys*, 2013, pp. 93–100.
- [5] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *KDD*, 2011, pp. 1082–1090.
- [6] J.-D. Zhang and C.-Y. Chow, "Spatiotemporal sequential influence modeling for location recommendations: A gravity-based approach," *TIST*, 2015.
- [7] Z. Yin, L. Gao, J. Han, J. Luo, and T. S. Huang, "Diversified trajectory pattern ranking in geo-tagged social media," in *SIAM*, 2011, pp. 980–991.
- [8] H.-P. Hsieh, C.-T. Li, and S.-D. Lin, "Measuring and recommending time-sensitive routes from location-based data," *TIST*, vol. 5, no. 3, pp. 45:1–45:27, 2014.
- [9] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *WWW*, 2009, pp. 791–800.
- [10] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive wi-fi mobility data," in *INFOCOM*, vol. 2, 2004, pp. 1414–1424 vol.2.
- [11] H. Cheng, J. Ye, and Z. Zhu, "What's your next move: User activity prediction in location-based social networks," in *SDM*, 2013, pp. 171–179.
- [12] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *IJCAI*, 2013, pp. 2605–2611.
- [13] J.-D. Zhang, C.-Y. Chow, and Y. Li, "Lore: Exploiting sequential influence for location recommendations," in *SIGSPATIAL*, 2014, pp. 103–112.
- [20] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang, "A temporal context-aware model for user behavior modeling in social media systems," in *SIGMOD*, 2014, pp. 1543–1554.
- [21] B. Hu and M. Ester, "Spatial topic modeling in online social media for location recommendation," in *RecSys*, 2013, pp. 25–32.
- [22] 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*, 2011, pp. 338–349.
- [23] H. M. Wallach, "Topic modeling: Beyond bag-of-words," in *ICML*, 2006, pp. 977–984.
- [24] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, 1989.
- [25] P. Ram and A. G. Gray, "Maximum inner-product search using cone trees," in *KDD*, 2012, pp. 931–939.
- [26] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, "Lcars: A spatial item recommender system," *TOIS*, vol. 32, no. 3, pp. 11:1–11:37, 2014.
- [27] T. Liu, A. W. Moore, A. Gray, and K. Yang, "An investigation of practical approximate nearest neighbor algorithms," in *MIT Press*, 2004, pp. 825–832.
- [28] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SCG*, 2004, pp. 253–262.
- [29] A. Shrivastava and P. Li, "Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips)," in *NIPS*. Curran Associates, Inc., 2014, pp. 2321–2329.
- [30] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services," in *ICWSM*, 2011.
- [31] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, "A general geographical probabilistic factor model for point of interest recommendation," *TKDE*, vol. 27, no. 5, pp. 1167–1179, 2015.
- [32] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, "Geosage: A geographical sparse additive generative model for spatial item recommendation," in *KDD*, 2015.
- [33] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *SIGIR*, 2013, pp. 363–372.