

Adapting to User Interest Drift for POI Recommendation

Hongzhi Yin, Xiaofang Zhou, Bin Cui, Hao Wang, Kai Zheng, and Quoc Viet Hung Nguyen

Abstract—Point-of-Interest recommendation is an essential means to help people discover attractive locations, especially when people travel out of town or to unfamiliar regions. While a growing line of research has focused on modeling user geographical preferences for POI recommendation, they ignore the phenomenon of user interest drift across geographical regions, i.e., users tend to have different interests when they travel in different regions, which discounts the recommendation quality of existing methods, especially for out-of-town users. In this paper, we propose a latent class probabilistic generative model Spatial-Temporal LDA (ST-LDA) to learn region-dependent personal interests according to the contents of their checked-in POIs at each region. As the users' check-in records left in the out-of-town regions are extremely sparse, ST-LDA incorporates the crowd's preferences by considering the public's visiting behaviors at the target region. To further alleviate the issue of data sparsity, a social-spatial collective inference framework is built on ST-LDA to enhance the inference of region-dependent personal interests by effectively exploiting the social and spatial correlation information. Besides, based on ST-LDA, we design an effective attribute pruning (AP) algorithm to overcome the curse of dimensionality and support fast online recommendation for large-scale POI data. Extensive experiments have been conducted to evaluate the performance of our ST-LDA model on two real-world and large-scale datasets. The experimental results demonstrate the superiority of ST-LDA and AP, compared with the state-of-the-art competing methods, by making more effective and efficient mobile recommendations.

Index Terms—POI recommendation, user interest drift, collective inference, social-spatial correlation, user modeling

1 INTRODUCTION

RECENT years have witnessed the increased development and popularity of location-based social networks (LBSNs), such as Yelp, Foursquare and Facebook Places, due to the advances in location-acquisition and wireless communication technologies. In these LBSNs, users can post their physical locations in the form of “check-in”, and share their life experiences in the physical world. It is crucial to utilize user check-in data to make personalized POI recommendation in LBSNs, which helps users know new POIs and explore new regions (e.g., cities), facilitate advertisers to launch mobile advertisements to targeted users. This application becomes more important and useful when a user travels to an unfamiliar area, where she has little knowledge of the neighbourhood. In this scenario, the recommender system is proposed as *recommendation for out-of-town users* in [1]. In this paper, we aim to offer accurate recommendations for both home-town and out-of-town users by mining their check-in data in LBSNs.

One of the most important problems for POI recommendation is how to deal with a severe challenge stemming from extreme sparsity of user-POI interaction matrix. There are millions of POIs in LBSNs, but a user can only visit a limited number of them. Moreover, the observation of travel locality exacerbates this problem. The observation of travel locality [2] 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 percent of the ones generated in their home cities [3]. 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) [1], [4].

A promising way for alleviating the data sparsity, especially in the out-of-town recommendation scenario, is to exploit and integrate the content information of POIs. Some recent literatures [4], [5], [6], [7] exploited the content information of checked-in POIs (e.g., categories or tags) to infer the users' interests, which were then used to make POI recommendation. By integrating the content information, these methods can alleviate the data sparsity issue for out-of-town recommendation to some extent, since they can transfer users' interests inferred at their home towns to out-of-town regions by the medium of contents. However, all these studies do not consider the phenomenon of *user interest drift across geographical regions*, i.e., users tend to have different interests when they travel in different regions which have different urban compositions. For example, a user u never goes gambling when she lives in Beijing, China, but when she travels in Macao or Las Vegas she is most likely to visit

- H. Yin, X. Zhou, and Q. Nguyen are with the School of ITEE, University of Queensland, QLD 4072, Australia. E-mail: db.hongzhi@gmail.com, zxf@itee.uq.edu.au, q.nguyen@uq.edu.au.
- B. Cui is with the Key Lab of High Confidence Software Technologies (MOE), School of EECS, Peking University, Beijing 100871, China. E-mail: bin.cui@pku.edu.cn.
- H. Wang is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. E-mail: cashenry@126.com.
- K. Zheng is with the School of Computer Science and Technology, Soochow University, Suzhou 215000, China. E-mail: zhengkai@suda.edu.cn.

Manuscript received 22 Nov. 2015; revised 4 June 2016; accepted 8 June 2016.
Date of publication 14 June 2016; date of current version 7 Sept. 2016.

Recommended for acceptance by J. Xu.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TKDE.2016.2580511

TABLE 1
Illustration of User Interest Drift

City	Top POI Types	Percentage of Check-ins(%)
Gold Coast (AU)	Beach	71.36%
	Surf Spot	14.82%
	Theme Park	9.60%
	Scenic Lookout	3.36%
Las Vegas (US)	Casino	80.32%
	Nightlife	10.61%
	Outlet	5.82%
	Hotel	3.23%
Istanbul (Turkey)	Mosque	68.32%
	Museum	15.45%
	Cafe	7.65%
	Art Gallery	5.83%

casinos. Through Foursquare API,¹ we derive top four categories of a group of users' visited POIs in three different cities, as shown in Table 1. The group size is 3000 and each member of this group has check-in records in Gold Coast (AU), Las Vegas and Istanbul (Turkey). Diverse POI types can be observed: users traveling in Las Vegas are interested in Casino (80.32 percent), Nightlife (10.61 percent), and Outlet (5.82 percent), while users traveling in Gold Coast prefer Beach (71.36 percent), Surf Spot (14.82 percent), and Theme Park (9.60 percent). Thus, users' interests inferred at one region may not be applied to the POI recommendation at another region.

In light of this, we propose a latent class probabilistic generative model, called Spatio-Temporal LDA (ST-LDA), to adapt to user interest drift and improve the effectiveness of POI recommendation, especially for out-of-town users. Inspired by recent works about user interest modeling [4], [8], [9], ST-LDA adopts latent topics to characterize users' interests to overcome the sparsity of user-word matrix. Specifically, we first divide the geographical space into multiple regions, and then infer individual user's interest distribution over a set of topics w.r.t. each region according to the contents (e.g., tags and categories) of her visited POIs at that region. As the quality of the learnt topics is very important for user interest modeling, we exploit the temporal patterns of the public visiting POIs to improve the process of topic discovery. For example, the POIs frequently visited at lunch and dinner time are more likely to be restaurants, while the ones visited around midnight are more likely to be nightlife spots. To distinguish user interests in different regions, the method of region division is essential for ST-LDA. As topics and regions are interdependent, ST-LDA combines geographical clustering and topic modeling into a unified process.

Due to travel locality, users' footprints left in each non-home region are extremely sparse, which makes it very challenging to infer users' interests in these regions. To alleviate the data sparsity of individual users, ST-LDA leverages the preferences of the crowd who have the same role with the target user when they travel in the target region. Specifically, ST-LDA recognizes two roles of an individual user in a geographical region: a local or a tourist. Given a region, check-in records from local users will be mined to

learn *native preferences*. Similarly, check-in records from tourists will be used to learn *tourist preferences*. Users with the same role at a region are more likely to have similar preferences and behavior patterns. Thus, to recommend POIs to a target user u at a target region r , we consider not only u 's personal interests at that region, but also the preferences of the crowd who have the same role with u at region r . Moreover, we propose a social-spatial collective inferring framework to enhance the inference of users' interests in each region, especially out-of-town regions, by exploiting the social and spatial correlation information. Recent research [10] showed that there is a strong correlation between users' friendship and their mobility, especially when users travel out of town. Besides, if two regions are geographically proximate, the user's interests in these two regions should be similar, due to their similar urban compositions and cultures. To the best of our knowledge, this is the first work to adapt to *user interest drift across geographical regions*.

To demonstrate the applicability of ST-LDA, we investigate how it supports two recommendation scenarios in a unified way: 1) home-town recommendation that assumes that the target user is located in her home town, i.e., to meet users' information need in their daily life, and 2) out-of-town recommendation that aims to meet users' information need when they travel out of town, especially in unfamiliar regions. It is worth mentioning that both of the recommendation scenarios should be time-aware [11], location-based [1] and personalized, which requires producing recommendation results in a real-time manner. To speed up the process of online recommendation, we design an effective attribute pruning algorithm to overcome the curse of dimensionality and support real-time POI recommendation.

The main contributions of our work are summarized as:

- We investigate the phenomenon of user interest drift across geographical regions and argue that the ability to adapt to users' drifting interests is important for POI recommendation, especially for out-of-town users.
- We propose a latent class probabilistic generative model (ST-LDA) to learn region-dependent personal interests and crowd's preferences to adapt to user interest drift. Moreover, we design a social-spatial collective inferring framework to further alleviate the data sparsity in non-home regions by effectively exploiting the social and spatial correlation.
- We develop an attribute-pruning algorithm to speed up the online recommendation by exploiting the distributions of both query and POI vectors.
- We conduct extensive experiments to evaluate the performance of our ST-LDA model and AP algorithm on two real-life and large-scale datasets. The experimental results show the superiority of our proposals in both recommendation effectiveness and efficiency.

Roadmap of this paper. Section 2 reviews the related work. Section 3 details our ST-LDA model. Section 4 presents a social-spatial smoothing framework. We propose an efficient online recommendation algorithm in Section 5. We

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

report the experimental results in Section 6 and conclude the paper in Section 7.

2 RELATED WORK

To improve POI recommendation, many recent work has tried to explore and integrate geo-social, temporal and semantic information associated with users' check-in activities.

Geo-Social Influence. Many recent studies [1], [10], [12], [13], [14], [15] 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 work mainly focuses on leveraging the geographical and social influences to improve recommendation accuracy. For example, Ye et al. [13] delved into POI recommendation by investigating the geographical influences among locations and proposed a framework that combines user preferences, social influence and geographical influence. Cheng et al. [16] investigated the geographical influence through combining a multi-center Gaussian model, matrix factorization and social influence together for location recommendation. Lian et al. [17] incorporated spatial clustering phenomenon resulted by geographical influence into a weighted matrix factorization framework to deal with the challenge from matrix sparsity.

Temporal Effect. The temporal effect of user check-in activities in LBSNs has also attracted much attention from researchers. POI recommendation with temporal effect mainly leverage temporal cyclic patterns and temporal sequential patterns on LBSNs. Gao et al. [18] investigated the temporal cyclic patterns of user check-ins in terms of temporal non-uniformness and temporal consecutiveness. Yuan et al. [11] incorporated the temporal cyclic information into a user-based collaborative filtering framework for time-aware POI recommendation. [19], [20], [21], [22], [23] focused on the task of successive personalized POI recommendation in LBSNs by embedding the sequential influences.

As described above, while there are many studies to improve POI recommendation by exploiting geographical-social influence and temporal effect, they did not address the challenges (e.g., data sparsity) arising from either *travel locality* or *interest drift* for the out-of-town recommendation. Most of the above work assumed that users are in their home towns, they did not consider users' real-time locations, nor their interest drift across regions.

Semantic Information. Most recently, researchers explored the content information of POIs to alleviate the problem of data sparsity. Hu et al. [8] proposed a spatial topic model for POI recommendation considering both spatial aspect and textual aspect of user posts from Twitter. Zhao et al. [24] studied both POI-associated contents and user sentiment information into POI recommendation. To alleviate the data sparsity issue for out-of-town recommendation, Yin et al. [4], [7] developed LCA-LDA and Geo-SAGE models to exploit the content information of checked-in POIs to infer both personal interests and local preferences.

Compared with ST-LDA, there is only one latent variable, topic z , to model personal interests in LCA-LDA and Geo-SAGE, and they did not consider the dependence

between personal interests and geographical regions. Besides, the geographical regions used in LCA-LDA and Geo-SAGE were predefined (e.g., cities or states) instead of automatically learnt from users' check-in activity data. Moreover, they did not consider the temporal effect, nor users' real-time locations, thus they cannot support POI recommendation in the mobile scenario. In addition, Geo-SAGE is a sparse additive generative model, and its time cost for model inference is much more expensive than LDA-like models.

ST-LDA is also related to both JIM [6] and TRM [25], [26] models developed in our previous work that considered the geographical influence, temporal effect, wisdom of the crowds and the content information of POIs in a unified way. Actually, all ST-LDA, JIM and TRM are LDA-like models (i.e., the variants of topic model LDA), but ST-LDA distinguishes from them in the following several points. First, JIM and TRM models assume that user interests are stable across geographical regions and ignores the spatial dynamics of user interests, while our ST-LDA assumes that users interests are region-dependent. Second, there are two latent variables in all these three models, topic z and region r . They are assumed to be conditionally independent given user u in JIM and TRM, thus there is not an edge between them in the graphical representations of JIM and TRM. Actually, the geographical clustering and topic modeling in JIM and TRM are two independent processes. In contrast, ST-LDA assumes that topics and regions are interdependent, and it combines geographical clustering and topic modeling into a unified process where they can influence and enhance each other. Third, although JIM and TRM also leveraged the wisdom of the crowds (i.e., local crowd's preferences) to overcome the sparsity of individuals' check-ins generated at non-home regions, they ignored the crowds' roles and did not distinguish local preferences from tourist preferences. Fourth, we develop a social-spatial smoothing technique in ST-LDA to further alleviate the data sparsity in non-home regions by effectively exploiting the social and spatial correlation. Fifth, we propose an attribute pruning algorithm to speed up the online recommendation based on ST-LDA model. To the best of our knowledge, this is the first work to study recommendation efficiency in large-scale and high-dimensional data.

3 ST-LDA MODEL

In this section, we first formulate the problem definition, and then present our proposed ST-LDA model.

3.1 Problem Definition

Notation. Throughout this paper, all vectors are column vectors and are denoted by bold lower case letters (e.g., θ and ϕ). We use calligraphic letters to represent sets (e.g., the POI set \mathcal{V}). For simplicity, we use their corresponding normal letters to denote their cardinalities (e.g., $V = |\mathcal{V}|$).

Definition 1 (POI). A POI is defined as a uniquely identified specific site (e.g., a restaurant or a cinema). In our model, a POI has three attributes: identifier, location and content. We use v to represent a POI identifier and l_v to denote its corresponding location attribute in terms of longitude and latitude

TABLE 2
A POI and Its Associated Information

Name: Darling Harbour
Location: Longitude: 151.200, Latitude: -33.877
Categories: Harbor, Marina, Park, and Plaza
Tags: scenic views, harbourside, fireworks, cafe

coordinates. Besides, there is textual semantic information associated with a POI, such as the category and tag words. We use the notation \mathcal{W}_v to denote the collection of words describing POI v . Table 2 shows an example of a POI.

Definition 2 (User Home Location). Following the recent work of [27], given a user u , we define the user’s home location as the place where she lives, denoted as l_u .

For a user u whose home location is not explicitly given, we adopt the method developed by [3]. This method discretizes the world into $25km$ by $25km$ cells and finds the cell with most of her check-ins. Then, her home location is defined as the average position of all her check-ins within the cell.

Definition 3 (Check-in Activity). A check-in activity is made up of a six tuple $(u, v, l_v, \mathcal{W}_v, t, s)$ that means user u visits POI v at time t with 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 is a tourist when visiting v .

Definition 4 (User Profile). For each user u , we create a user profile \mathcal{D}_u , which is a collection of check-in activities generated by u . The dataset \mathcal{D} used in our model consists of user profiles, i.e., $\mathcal{D} = \{\mathcal{D}_u : u \in \mathcal{U}\}$ where \mathcal{U} is the set of users.

Definition 5 (Topic). Given a collection of words \mathcal{W} , a topic z is defined as a multinomial distribution over \mathcal{W} , i.e., $\phi_z = \{\phi_{z,w} : w \in \mathcal{W}\}$ where each component $\phi_{z,w}$ denotes the probability of topic z generating word w .

Given a dataset \mathcal{D} as a collection of user profiles, we aim to provide POI recommendation for both home-town and out-of-town users. We formulate our problem that takes into account both of the two scenarios in a unified fashion as follows.

Problem 1 (POI Recommendation). Given a user check-in activity dataset \mathcal{D} and a querying user u_q with her current location l_q , time t_q and role s_q (that is, the query is $q = (u_q, t_q, l_q, s_q)$), our goal is to recommend top- k new POIs that u_q would be interested in. Given a distance threshold d , the problem becomes an out-of-town recommendation if the distance between her current location and her home location (that is, $|l_q - l_{u_q}|$) is greater than d . Otherwise, the problem is a home-town recommendation.

3.2 Model Structure

For the ease of presentation, we first list the notations in Table 3. Fig. 1 shows the graphical representation of ST-LDA. It is a generative model jointly over the geographical location, text, check-in time and user role in the check-in record. It discovers latent topics and regions, and learns region-dependent users’ interests, spatial mobility patterns and the crowd’s preferences in a

TABLE 3
Notations of Model Parameters

Variable	Interpretation
ϑ_u	the spatial patterns of user u , expressed by a multinomial distribution over a set of regions
$\theta_{u,r}$	the interests of user u at region r , expressed by a multinomial distribution over a set of topics
ϕ_z	a multinomial distribution over words specific to topic z
ψ_z	a multinomial distribution over time specific to topic z
$\varphi_{s,r}$	the preferences of the crowd with role s for region r , expressed by a multinomial distribution over set of POIs; that is $\varphi_{0,r}$ denotes the native preferences for region r , and $\varphi_{1,r}$ represents the tourist preferences for region r
μ_r	the mean location of region r
Σ_r	the location covariance of region r
K, R	the number of topics and regions, respectively
$\gamma, \alpha, \beta, \eta, \tau$	Dirichlet priors to multinomial distributions $\vartheta_u, \theta_{u,r}, \phi_z, \varphi_r$ and ψ_z respectively

unified way. Below, we will describe each component in ST-LDA.

Time-Aware Topic Discovery. Inspired by recent studies on user interest modeling [4], [9], ST-LDA adopts latent topics to characterize users’ interests to overcome the sparsity of user-word matrix. Thus, the quality of topics is very important for accurately modeling users’ interests. To improve the topic discovery process, we exploit the temporal patterns of the general public visiting POIs, or more exactly daily patterns. Intuitively, different types of POIs have different temporal patterns of check-ins, and two POIs exhibiting similar temporal patterns are more likely to have the same/similar functions and categories than two random ones, which is validated in Section 6.6. For example, the POIs frequently visited at lunch and dinner time are more likely to be restaurants, while the ones visited around midnight are more likely to be nightlife spots. Based on this intuition, a topic z in ST-LDA is responsible for simultaneously generating semantic words \mathcal{W}_v and check-in time t . Thus, each topic z in ST-LDA is not only associated with a word distribution $\phi_{\cdot, z}$, but also with a distribution over time

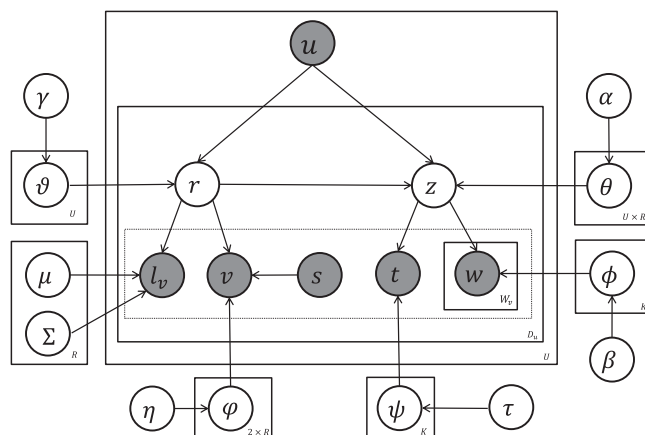


Fig. 1. The graphical representation of ST-LDA.

ψ_z . This design enables ϕ_z and ψ_z to be mutually influenced and enhanced during the topic discovery process, facilitating the clustering of the words of POIs with similar temporal patterns into the same topic with high probability. To integrate the check-in time information to the topic discovery process, we adopt the discretization method in [11], [18] to split a day into hourly-based slots.

Geographical Region Modeling. To distinguish user interests in different regions, the method of region division is essential for our ST-LDA. The spatial pyramid methods (such as R-Tree) have been widely used to divide, index and manage the geographical space, but it considers only the geographical location information of POIs and ignores the collaborative check-in behaviors of users. Thus, the geographical regions indexed by the spatial pyramid do not necessarily correspond to the activity regions of users in reality. Although administrative divisions such as countries, states and cities can be used as activity regions, we are more interested in different region segmentations corresponding to different topics or themes. In other words, we expect that the divided geographic regions are coherent in topics or themes. However, existing studies either overlook interdependence between topics and regions or employ state/city as the fixed configuration [4], [9]. ST-LDA combines geographical clustering and topic modeling into a unified process to capture their interdependence. Specifically, we cluster the geographical locations of check-ins into R regions. Following the literatures [6], [28], we assume a Gaussian distribution for each region r , and the geographical location of POI v is characterized by $l_v \sim \mathcal{N}(\mu_r, \Sigma_r)$, as follows:

$$P(l_v | \mu_r, \Sigma_r) = \frac{1}{2\pi\sqrt{|\Sigma_r|}} \exp\left(-\frac{(l_v - \mu_r)^T \Sigma_r^{-1} (l_v - \mu_r)}{2}\right), \quad (1)$$

where μ_r and Σ_r denote the mean vector and covariance matrix.

$$\begin{aligned} P(v, l_v, w, t, z, r, \psi, \theta, \phi, \varphi | \alpha, \beta, \gamma, \eta, \tau, \mu, \Sigma, s) \\ = P(r | \vartheta) P(\vartheta | \gamma) P(\theta | \alpha) P(z | r, \theta) P(\phi | \beta) P(w | z, \phi) \\ P(\psi | \tau) P(t | z, \psi) P(\varphi | \eta) P(v | r, s, \varphi) P(l_v | r, \mu, \Sigma) \end{aligned} \quad (2)$$

Region-Dependent User Interest Modeling. Based on the latent topics and regions, this component is developed to uncover region-dependent users' interests, which also serves to seamlessly unify *geographical clustering* and *topic modeling*. Being different from existing user interest models [4], [6], [9], we distinguish a user's interests from region to region. Specifically, we associate a user with R region-dependent topic distributions, i.e., $\{\theta_{u,r}\}_{r=1}^R$ where $\theta_{u,r}$ represents user u 's interests w.r.t. region r . We infer $\theta_{u,r}$ according to the contents of u 's visited POIs at region r .

Individual's Spatial Pattern Modeling. Being different from users' online behaviors in the virtual world, users' check-in activities in the physical world are limited by travel distance. So, it is also important to capture users' spatial patterns (or activity ranges) according to the location distributions of their historical checked-in POIs. The spatial clustering phenomenon indicates that users tend to visit

POIs around several centers (e.g., "home" and "office"), and thus these POIs are usually limited to some specific geographical regions [13]. We apply a multinomial distribution over regions ϑ_u to model a user's spatial patterns.

Role-Aware Crowd's Preference Modeling. The word-of-mouth opinions from the crowd who have also visited the target region also affect the users' check-in behaviors to a great extent, especially out-of-town users. To adapt to user interest drift across regions, we exploit the preferences of the crowd who share the same role with the target user u . For example, the preferences of the tourists will be leveraged if the target user is currently out-of-town. Technically, we introduce two model parameters: *native preferences* and *tourist preferences*. Given a region r , the native preferences represent the preferences of people living at region r , denoted as $\varphi_{0,r}$. In contrast, the tourist preferences represent the preferences of tourists travelling in region r , denoted as $\varphi_{1,r}$. Both $\varphi_{0,r}$ and $\varphi_{1,r}$ are represented by a multinomial distribution over POIs. Intuitively, $\varphi_{s,r,v}$ reflects the popularity of POI v among the local people ($s = 0$) or the tourists ($s = 1$) at region r . Note that, distinguishing native preferences from tourist preferences is one of the fundamental differences between our model and LCA-LDA model [4] which also exploits the crowd's preferences at the target region.

3.3 Generative Process

The generative process of ST-LDA is summarized in Algorithm 1. To avoid overfitting, we place a Dirichlet prior [29] over each multinomial distribution (i.e., $\phi_z, \psi_z, \varphi_{s,r}, \vartheta_u$ and $\theta_{u,r}$).

Algorithm 1: Probabilistic generative process in ST-LDA

```

for each topic  $z$  do
  Sample a distribution over words:  $\phi_z \sim \text{Dirichlet}(\cdot | \beta)$ ;
  Sample a distribution over time:  $\psi_z \sim \text{Dirichlet}(\cdot | \tau)$ ;
end
for each region  $r$  do
  for each role  $s$  do
    Sample a distribution over POIs:  $\varphi_{s,r} \sim \text{Dirichlet}(\cdot | \eta)$ ;
  end
end
for each user  $u$  do
  Sample her distribution over regions:  $\vartheta_u \sim \text{Dirichlet}(\cdot | \gamma)$ ;
  for each region  $r$  do
    Sample her distribution over topics at region  $r$ :
     $\theta_{u,r} \sim \text{Dirichlet}(\cdot | \alpha)$ ;
  end
end
for each  $\mathcal{D}_u$  in  $\mathcal{D}$  do
  for each check-in  $(u, v, l_v, \mathcal{W}_v, t, s) \in \mathcal{D}_u$  do
    Sample region indicator  $r \sim \text{Multi}(\vartheta_u)$ ;
    Sample topic indicator  $z \sim \text{Multi}(\theta_{u,r})$ ;
    Sample POI indicator  $v \sim \text{Multi}(\varphi_{s,r})$ ;
    Sample geographical coordinate  $l_v \sim \mathcal{N}(\mu_r, \Sigma_r)$ ;
    for each token  $w \in \mathcal{W}_v$  do
      Sample word  $w \sim \text{Multi}(\phi_z)$ ;
    end
    Sample time  $t \sim \text{Multi}(\psi_z)$ ;
  end
end

```

Given a user u and her current role s , when she plans to visit a POI v , she first selects a region r according to her region distribution ϑ_u , then chooses a topic z by her topic distribution $\theta_{u,r}$ at the chosen region r . With the chosen

region r , POI indicator v is generated by the crowd's preferences $\varphi_{s,r}$, and the POI's geographical coordinate l_v is generated by the chosen region's spatial distribution $\mathcal{N}(\mu_r, \Sigma_r)$. With the chosen topic z , words \mathcal{W}_v are generated from the topic's word distribution ϕ_z , and time t is generated from the topic's temporal distribution ψ_z .

3.4 Model Inference

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables w, t, s, v and l_v , and the marginalization is performed with respect to the latent random variables r and z . However, it is difficult to be maximized directly. Therefore, we apply a mixture between EM and a Monte Carlo sampler, called Gibbs EM algorithm [30], to maximize the complete data likelihood in Equation 2. In the E-step, we sample latent regions and topics by fixing all other parameters (e.g., μ and Σ) by collapsed Gibbs sampling. As a widely used Markov chain Monte Carlo algorithm, Gibbs sampling iteratively samples latent variables (i.e., $\{r, z\}$ in ST-LDA) from a Markov chain, whose stationary distribution is the posterior. The samples can therefore be used to estimate the distributions of interest (i.e., $\{\vartheta, \theta, \phi, \psi, \varphi\}$). As for hyperparameters $\alpha, \beta, \gamma, \eta$ and τ , for simplicity, we take a fix value, i.e., $\alpha = 50/K, \gamma = 50/R$ and $\beta = \eta = \tau = 0.01$, following the studies [4], [9]. In the M-step, we optimize model parameters μ and Σ by fixing all region and topic assignments. We iterate this until convergence.

More specifically, in the E-step, we iteratively draw latent region assignments and topic assignments for all check-in records. For i -th check-in record of user u , a latent region r is firstly drawn from the following distribution, conditioned on the old topic assignments:

$$P(r_{u,i} = r | r_{-}, z, v, l_v, \mathcal{W}_v, t, s, u, \cdot) \propto (n_{u,r}^{-} + \gamma) \times \frac{n_{u,r,z}^{-} + \alpha}{\sum_{z'} (n_{u,r,z'}^{-} + \alpha)} \frac{n_{s,r,v}^{-} + \eta}{\sum_{v'} (n_{s,r,v'}^{-} + \eta)} P(l_v | \mu_r, \Sigma_r), \quad (3)$$

where r_{-} represents region assignments for all check-in records except the current one; $n_{u,r}$ is the number of times that region r is sampled from user u ; $n_{s,r,v}$ is the number of times the POI v is sampled from the native preferences ($s = 0$) or tourist preferences ($s = 1$) at region r ; $n_{u,r,z}$ is the number of times that latent topic z is sampled from user u at region r ; and the number n^{-} with superscript $-$ denotes a quantity excluding the current instance.

After r is sampled, we sample the topic assignment z for the same check-in record, conditioned on the newly sampled r :

$$P(z_{u,i} = z | z_{-}, r, v, l_v, \mathcal{W}_v, t, s, u, \cdot) \propto (n_{u,r,z}^{-} + \alpha) \times \frac{n_{z,t}^{-} + \tau}{\sum_{t'} (n_{z,t'}^{-} + \tau)} \prod_{w \in \mathcal{W}_v} \frac{n_{z,w}^{-} + \beta}{\sum_{w'} (n_{z,w'}^{-} + \beta)}, \quad (4)$$

where r is the new region index, and z_{-} represents topic assignments for all check-in records except the current one; $n_{z,w}$ is the number of times that word w is generated from topic z , and $n_{z,t}$ is the number of times that time t is generated from topic z .

In the M-step, we maximize the log likelihood of the model with respect to model parameters μ and Σ by fixing

all region assignments obtained in the E-step. The maximum likelihood estimation (MLE) can be obtained in the closed form as follows:

$$\mu_r = E(r) = \frac{1}{|\mathcal{S}_r|} \sum_{v \in \mathcal{S}_r} l_v \quad (5)$$

$$\Sigma_r = D(r) = \frac{1}{|\mathcal{S}_r| - 1} \sum_{v \in \mathcal{S}_r} (l_v - \mu_r)(l_v - \mu_r)^T, \quad (6)$$

where \mathcal{S}_r denotes the collection of POIs assigned with region r .

Time Complexity. We analyze the time complexity of our model inference algorithm. Suppose the process needs I iterations to reach convergence. In each iteration, it requires to go through all user check-in records. For each check-in record, it first requires $\mathcal{O}(R)$ operations to compute the posterior distribution for sampling latent region, and then needs $\mathcal{O}(K)$ operations to compute the posterior distribution for sampling latent topic. In addition, it requires constant cost to update μ and Σ in the M-step. Thus, the whole time complexity is $\mathcal{O}(I(K+R)D)$, which is linear to the number D of check-in records in the dataset. To ensure the scalability of our model, we implemented a parallel ST-LDA inference algorithm on the GraphLab framework, following the implementation method developed in [31]. GraphLab is a vertex centric programming framework, expressing computational dependencies with a distributed graph. It has demonstrated superior performance over popular parallel systems such as Map-Reduce, for many machine learning algorithms.

4 SOCIAL-SPATIAL SMOOTHING

Due to travel locality, users' footprints left in each non-home region are extremely sparse, which makes it very challenging to infer users' real interests in these regions. The estimated $\theta_{u,r}$ in the out-of-town regions tends to be an uniform distribution over topics. To alleviate the data sparsity, we exploit the social-spatial correlation information to enhance the inference of user interests at these non-home regions. Our enhancements make use of two intuitions. First, if two users u and u' are close in the social network, their interests at the region r (i.e., $\theta_{u,r}$ and $\theta_{u',r}$) are similar to each other. Second, if two geographical regions, r and r' , are close in the geographical space, the user u 's interests at these two regions (i.e., $\theta_{u,r}$ and $\theta_{u,r'}$) should approximate to each other due to the similar urban compositions and cultures. Thus, the user's interest $\theta_{u,r}$ changes smoothly along the social and the spatial dimensions, respectively. So, given a target user u and a region r , both the interests of u 's friends at region r and u' interests at other adjacent regions provide clues for the enhancement of inferring u 's interest at region r . Based on the two intuitions, we propose a social-spatial collective inferring framework to enhance the inference of the user's region-dependent interests. This framework consists of two components: social smoothing component and spatial smoothing component.

Social Smoothing. Given a social network $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \Pi)$ where \mathcal{U} is the set of users (i.e., nodes), \mathcal{E} is the set of edges linking nodes, and Π is the collection of weights on the edges. For example, given $u, u' \in \mathcal{U}$, there is a social link between

them, and the weight $\pi_{u,u'} \in \Pi$ denotes the relation strength between them. Given a user u and a region r , we adopt the relaxation labeling method [32] to improve the inference of her interests at region r , i.e., $\theta_{u,r}$, from her neighbors. Specifically, the inference of $\theta_{u,r}$ can be recursively enhanced by the set of u 's friends $\mathcal{S}_{u,r}$ whose interest entropies w.r.t. region r are smaller than that of $\theta_{u,r}$. The *interest entropy* of user u w.r.t. region r is computed as follows:

$$Entropy(\theta_{u,r}) = - \sum_z \theta_{u,r,z} \log \theta_{u,r,z}. \quad (7)$$

We use $Entropy(\theta_{u,r})$ to quantify the informativeness and reliability of u 's interests inferred at region r . When u has few check-in records in region r , $\theta_{u,r}$ tend to be a uniform distribution at the beginning of iteration. Given a region r , the local people's interest entropies are much smaller than that of the out-of-town users (e.g., travelers). Thus, $\mathcal{S}_{u,r}$ initially consists of u 's friends living in region r or its nearby regions.

Relaxation labeling freezes the current estimations of $\theta_{u,r}$, so that, at round $i+1$, all $\theta_{u,r}$ will be updated based on the estimations from round i . As shown below, $\theta_{u,r}$ is calculated by considering both the weighted average of the interest distributions of users in $\mathcal{S}_{u,r}$ and the current estimation of $\theta_{u,r}$ itself.

$$\theta_{u,r}^{i+1} = \lambda^{i+1} \frac{\sum_{u' \in \mathcal{S}_{u,r}} \pi_{u,u'} \theta_{u',r}^i}{\sum_{u' \in \mathcal{S}_{u,r}} \pi_{u,u'}} + (1 - \lambda^{i+1}) \theta_{u,r}^i, \quad (8)$$

where $\theta_{u,r}^i$ denotes the estimation of $\theta_{u,r}$ at the i -th round. As users' region-dependent interests (i.e., $\theta_{u,r}$) are dynamically updated in the iteration process, thus, $\mathcal{S}_{u,r}$ is also automatically updated. As for the mixture parameter λ , it is also dynamically updated, as follows:

$$\lambda^{i+1} = \xi \lambda^i, \quad (9)$$

where λ^0 is a constant between 0 and 1, and ξ is a decay factor, i.e., $0 < \xi < 1$. The calculation of $\theta_{u,r}$ with larger λ^0 setting converges slower than the one with smaller λ^0 [33]. More importantly, a larger λ^0 implies that user u 's interest at region r (i.e., $\theta_{u,r}$) should be estimated not only according to herself, but also influenced by her friends and users who are in multi-hops away as there are multiple rounds of calculation. A smaller λ^0 suggests that the user's interest is mainly affected by herself and smoothed only by her close-by friends as there are very few rounds of calculation.

Spatial Smoothing. Similar to the social smoothing, we also adopt the relaxation labeling method [32], [33] for spatial smoothing. Specifically, $\theta_{u,r}$ is calculated by considering both the weighted average of u 's interests at other close regions $\mathcal{R}_{u,r}$ and the current estimation of $\theta_{u,r}$ itself.

$$\theta_{u,r}^{i+1} = \lambda^{i+1} \frac{\sum_{r' \in \mathcal{R}_{u,r}} \pi_{r,r'} \theta_{u,r'}^i}{\sum_{r' \in \mathcal{R}_{u,r}} \pi_{r,r'}} + (1 - \lambda^{i+1}) \theta_{u,r}^i, \quad (10)$$

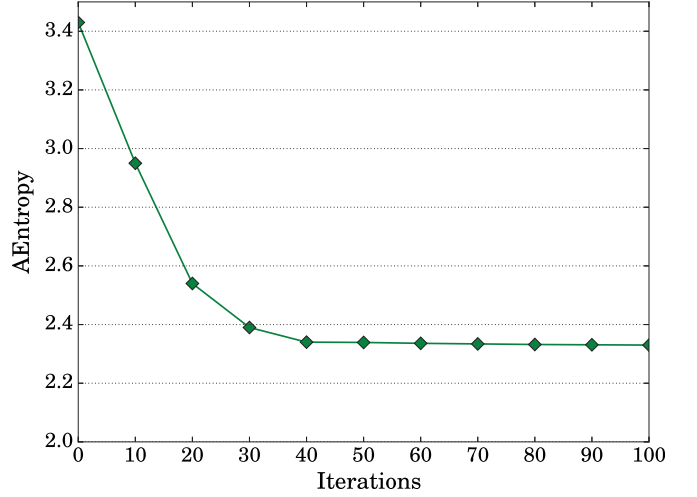


Fig. 2. Analysis of variation trend of "AEntropy".

where $\mathcal{R}_{u,r}$ denotes a set of chosen regions, and for any $r' \in \mathcal{R}_{u,r}$, $Entropy(\theta_{u,r'})$ is smaller than $Entropy(\theta_{u,r})$. During the iteration process, $\mathcal{R}_{u,r}$ is dynamically updated.

Social-Spatial Collective Inferring Framework. Once we have inferred users' region-dependent interests (i.e., $\theta_{u,r}$) by running the Gibbs EM algorithm presented in Section 3.4, we alternatively iterate over Social Smoothing and Spatial Smoothing until convergence. There are many possible ways to compute user-user similarity score (i.e., $\pi_{u,u'}$) and region-region similarity score (i.e., $\pi_{r,r'}$). Goyal et al. [34] proposed a method to compute the influence probability from u' to u in a social network, and we adopt the social influence probability from u' to u as the weight on the edge from u' to u . The method proposed in [34] not only considers the number of their common check-in POIs, but also takes their check-in orders into account. For the computation of region-region similarity score, we employ the widely used Gaussian kernel function which takes the distance (i.e., $\|\mu_r - \mu_{r'}\|$) between regions r and r' as the input. Thus, our proposed spatial smoothing can be also called Gaussian Smoothing. In fact, both social smoothing and spatial smoothing are a K -dimension convolution operator.² As for the parameters ξ and λ^0 , we set them to 0.99 and 0.8 based on the empirical results in [32].

Fig. 2 shows the variation trend of the average user interest entropy on the Foursquare dataset as the number of iterations increases, and the *average user interest entropy* is defined as follows.

$$AEntropy = \frac{1}{U} \sum_u \frac{1}{R} \sum_r Entropy(\theta_{u,r}).$$

From the figure, we observe that the average user interest entropy decreases with the increasing iterations. The result shows that 1) our proposed Social-Spatial Collective Inferring Framework is effective to improve the inference of region-dependent user interests; and 2) the framework is efficient since it only needs about 50 iterations to converge.

2. <https://en.wikipedia.org/wiki/Convolution>

5 POI RECOMMENDATION USING ST-LDA

Once we have estimated the model parameter set $\hat{\Psi} = \{\hat{\theta}, \hat{\vartheta}, \hat{\phi}, \hat{\psi}, \hat{\mu}, \hat{\Sigma}\}$, given a target user u_q with the current time t_q and location l_q , we first compute the indicator s_q (i.e., the role of target user u_q) according to the distance between her home location l_{u_q} and l_q . Thus, a query $q = (u_q, t_q, l_q, s_q)$ is formed. Then, we compute the probability of user u_q choosing each unvisited POI v as follows:

$$P(v|q, \hat{\Psi}) = \frac{P(v, t_q|u_q, l_q, s_q, \hat{\Psi})}{\sum_{v'} P(v', t_q|u_q, l_q, s_q, \hat{\Psi})} \propto P(v, t_q|u_q, l_q, s_q, \hat{\Psi}), \quad (11)$$

where $P(v, t_q|u_q, l_q, s_q, \hat{\Psi})$ is calculated as follows:

$$P(v, t_q|u_q, l_q, s_q, \hat{\Psi}) = \sum_r P(r|l_q, \hat{\Psi}) P(v, t_q|u_q, r, s_q, \hat{\Psi}), \quad (12)$$

where $P(r|l_q, \hat{\Psi})$ denotes the probability of u_q choosing region r given her current location l_q , and it is computed as in Equation (13) according to Bayes rule, in which the prior probability of latent region r can be estimated using Equation (18).

$$P(r|l_q, \hat{\Psi}) = \frac{P(r)P(l_q|r, \hat{\Psi})}{\sum_{r'} P(r')P(l_q|r', \hat{\Psi})} \propto P(r)P(l_q|r, \hat{\Psi}) \quad (13)$$

$$P(r) = \sum_u P(r|u)P(u) = \sum_u \frac{n_u + \kappa}{\sum_{u'} (n_{u'} + \kappa)} \vartheta_{u,r}, \quad (14)$$

where n_u denotes the number of check-ins generated by user u . In order to avoid overfitting, we introduce the Dirichlet prior parameter κ to play the role of pseudocount. Note that to support dynamic real-time recommendation, we compute the probability of u_q choosing region r according to her real-time location l_q instead of the spatial patterns (i.e., $\vartheta_{u,r}$) learnt from her historical check-in records, which distinguishes this work from the static recommendation scheme adopted by most POI recommendation work [8], [13], [17], [18], [24].

$P(v, t_q|u_q, r, s_q, \hat{\Psi})$ in Equation (12) is defined as in Equation (15) where we adopt geometric mean for the probability of topic z generating word set \mathcal{W}_v , i.e., $P(\mathcal{W}_v|z, \hat{\Psi}) = \left(\prod_{w \in \mathcal{W}_v} P(w|z, \hat{\Psi}) \right)^{\frac{1}{|\mathcal{W}_v|}}$, considering that the number of words W_v may be different for different POIs.

Based on Equations (12-15), the original Equation (11) can be reformulated as in Equation (16).

$$P(v, t_q|u_q, r, s_q, \hat{\Psi}) = P(l_v|r, \hat{\Psi}) P(v|r, s_q, \hat{\Psi}) \sum_z P(z|u_q, r, \hat{\Psi}) P(t_q|z, \hat{\Psi}) \left(\prod_{w \in \mathcal{W}_v} P(w|z, \hat{\Psi}) \right)^{\frac{1}{|\mathcal{W}_v|}} \quad (15)$$

$$\begin{aligned} P(v|q, \hat{\Psi}) &\propto \sum_r \left[P(r) P(l_q|r, \hat{\Psi}) P(l_v|r, \hat{\Psi}) P(v|r, s_q, \hat{\Psi}) \right. \\ &\quad \left. \sum_z P(z|u_q, r, \hat{\Psi}) P(t_q|z, \hat{\Psi}) \left(\prod_{w \in \mathcal{W}_v} P(w|z, \hat{\Psi}) \right)^{\frac{1}{|\mathcal{W}_v|}} \right] \\ &= \sum_r \left[P(r) P(l_q|\hat{\mu}_r, \hat{\Sigma}_r) P(l_v|\hat{\mu}_r, \hat{\Sigma}_r) \hat{\phi}_{s_q, r, v} \right. \\ &\quad \left. \sum_z \hat{\theta}_{u_q, r, z} \hat{\psi}_{z, t_q} \left(\prod_{w \in \mathcal{W}_v} \hat{\phi}_{z, w} \right)^{\frac{1}{|\mathcal{W}_v|}} \right] \\ &= \sum_r \sum_z \left[P(r) P(l_q|\hat{\mu}_r, \hat{\Sigma}_r) P(l_v|\hat{\mu}_r, \hat{\Sigma}_r) \hat{\phi}_{s_q, r, v} \hat{\theta}_{u_q, r, z} \hat{\psi}_{z, t_q} \right. \\ &\quad \left. \left(\prod_{w \in \mathcal{W}_v} \hat{\phi}_{z, w} \right)^{\frac{1}{|\mathcal{W}_v|}} \right] \end{aligned} \quad (16)$$

5.1 Fast Top- k Recommendation Framework

In this subsection, we first propose a ranking framework according to Equation (16), as follows:

$$S(q, v) = \sum_{a=(s, r, z)} W(q, a) F(v, a) \quad (17)$$

$$W(q, a) = (s \odot s_q) \hat{\theta}_{u_q, r, z} \hat{\psi}_{z, t_q} P(l_q|\hat{\mu}_r, \hat{\Sigma}_r) \quad (18)$$

$$F(v, a) = P(r) P(l_v|\hat{\mu}_r, \hat{\Sigma}_r) \hat{\phi}_{s, r, v} \left(\prod_{w \in \mathcal{W}_v} \hat{\phi}_{z, w} \right)^{\frac{1}{|\mathcal{W}_v|}}, \quad (19)$$

where $S(q, v)$ represents the ranking score of POI v w.r.t. query q . Each role-region-topic triple (s, r, z) can be seen as an attribute (i.e., $a = (s, r, z)$), and $W(q, a)$ represents the weight of query q on attribute a , and $F(v, a)$ represents the score of POI v with respect to attribute a . \odot denotes the xnor operation. This ranking framework separates the offline computation from the online computation. Since $F(v, a)$ is independent of queries, it is computed offline. Although the query weight $W(q, a)$ is computed online, its main time-consuming components (i.e., $\hat{\psi}_{z, t_q}$, $\hat{\theta}_{u_q, r, z}$ and $(\hat{\mu}_r, \hat{\Sigma}_r)$) are also computed offline, the online computation is just a simple combination process. This design enables maximum pre-computation for the problem considered, and in turn minimizes the query time. At query time, the offline scores $F(v, a)$ only need to be aggregated over $A = R \times K$ attributes by a simple weighted sum function, once we obtain the role of the querying user (s_q).

5.2 Attribute Pruning Algorithm

The straightforward method of generating the top- k POIs needs to compute the ranking scores for all POIs according to Equation (17) and select the k ones with highest ranking scores, which is computationally inefficient, especially when the number of POIs or POI attributes becomes large [9], [35]. To speed up the online process of producing recommendations, some efficient online recommendation techniques [9], [35], [36] have been developed to prune the object (e.g., POI) search space, but they cannot be applied to our problem due to the curse of dimensionality. Specifically,

the number of attributes (i.e., $A = R \times K$) is very large, generally beyond thousands of dimensions in our problem. The high dimensionality disables existing tree index structures and tree-based search algorithms, such as Metric-Tree [36]. Another choice is the threshold algorithm (TA) [9] that may save the computation for some POIs. However, it also needs to maintain and access a large number of sorted lists and frequently update the *threshold*, which makes it slower. Hashing has been widely utilized to speed up large scale similarity search (e.g., near neighbor search), but the top- k recommendation problem is not equivalent to similarity search in traditional hashing. For our recommendation problem, queries' preferences over POIs are calculated as the inner product between query and POI vectors, as shown in Equation (17). Inner product between two vectors is fundamentally different from their cosine similarity unless all POI vectors have the same length.

To speed up the online recommendation, we propose an efficient algorithm to prune the attribute space and facilitate fast computation of the ranking score for a single POI, inspired by TA algorithm [9] and Region Pruning strategy [24]. Our algorithm is based on three observations that 1) a query q only prefers a small number of attributes (i.e., the sparsity of query preferences) and the query weights on most attributes are extremely small; 2) POIs with high values on these preferred attributes are more likely to become the top- k recommendation results; and 3) the attribute values of most POIs also exhibit sparsity, i.e., each POI has significant values for only a handful of attributes.

The above three observations indicate that only when a query prefers an attribute and the POI has a high value on that attribute, will the score $W(q, a)F(v, a)$ contribute significantly to the final ranking score. Thus, we first pre-compute ordered lists of POIs, where each list corresponds to a latent attribute learned by ST-LDA model. For example, given A latent attributes, we will compute A lists of sorted POIs, L_{a_j} , $1 \leq j \leq A$, where POIs in each list L_{a_j} are sorted according to $F(v, a_j)$ as defined in Equation (19). Different from the Threshold Algorithm developed in [9], each sorted list L_{a_j} only stores k POIs with highest $F(v, a_j)$ values instead of all POIs. Hence, it is space-saving. Besides, for each POI v , its attributes are pre-ranked offline according to the value of $F(v, a_j)$. Given an online query q , we develop a branch and bound algorithm, as shown in Algorithm 2, to prune the search space of the attributes in the computation of the ranking score, i.e., after we have scanned a small number of significant attributes for a POI, it may not be necessary to examine the remaining attributes. The algorithm is called Attribute Pruning and contains two components: *initialization* and *pruning*.

In the initialization component (Lines 1-18), we select k candidate POIs that are potentially good for recommendation. Specifically, we pick top m attributes which cover most the query's preferences with smallest m , i.e., $\sum_{j=1}^m W(q, a_j) > \rho \sum_{j=1}^A W(q, a_j)$, where ρ is a predefined constant between 0 and 1 (Line 3). In our experiment, AP achieves its best performance for $\rho = 0.9$. For each of the top m attributes a , we choose top ranked POIs from L_a as candidates (Lines 4-18).

Algorithm 2: Attribute Pruning Algorithm

Input: A POI set V , a query q and A ranked lists L_a ;
Output: Result list L with k highest ranking scores;

```

1 Initialize  $L$  as  $\emptyset$ ;
2 Sort the query attributes by  $W(q, a)$ ;
3 Choose top  $m$  attributes satisfying:
    $\sum_{j=1}^m W(q, a_j) > 0.9 \sum_{j=1}^A W(q, a_j)$ ;
4 for  $j = 1$  to  $m$  do
5   for  $v \in L_{a_j}$  and  $v \notin L$  do
6     Compute  $S(q, v)$  according to Equation 18;
7     if  $L.size() < k$  then
8        $L.add(< v, S(q, v) >)$ ;
9     end
10    else
11       $v' = L.top()$ ;
12      if  $S(q, v) > S(q, v')$  then
13         $L.removeTop()$ ;
14         $L.add(v, S(q, v))$ ;
15      end
16    end
17  end
18 end
19 for  $v \in V$  and  $v \notin L$  do
20    $PS = 0$ ,  $PW = 0$ ,  $Skip = false$ , and  $v' = L.top()$ ;
21   while there exists attribute  $a$  not examined for  $v$  do
22      $a = v.nextAttribute()$ ;
23      $PS = PS + W(q, a)F(v, a)$ ;
24      $PW = PW + W(q, a)$ ;
25     if  $PS + (\sum_{j=1}^A W(q, a_j) - PW)F(v, a) \leq S(q, v')$  then
26        $Skip = true$ ;
27       break;
28     end
29   end
30   if  $Skip == false$  then
31     if  $S(q, v) > S(q, v')$  then
32        $L.removeTop()$ ;
33        $L.add(< v, S(q, v) >)$ ;
34     end
35   end
36 end
37  $L.Reverse()$ ;
38 Return  $L$ ;
```

In the pruning component (Lines 19-36), we check whether we can avoid traversing unnecessary attributes for POI v according to the descending order of $F(v, a)$. Suppose we have traversed attributes $\{a_1, \dots, a_{i-1}\}$. The partial score we have computed for the traversed attributes is

$$PS(q, v) = \sum_{j=1}^{i-1} W(q, a_j)F(v, a_j).$$

When we explore the i -th attribute, we compute the upper bound of ranking score for the POI v as:

$$UB(q, v) = PS(q, v) + \sum_{j=i}^A W(q, a_j)F(v, a_i). \quad (20)$$

Because we check the attributes in the descending order of $F(v, a)$, the actual value of $F(v, a)$ for the remaining attributes should be less than the value for the current attribute, i.e., $F(v, a_i)$. Therefore, we have a partial ranking score for the rest of the attributes, which is at most

TABLE 4
Features of Different Methods

Methods \ Features	Geographical Influence	Temporal Effect	Content Effect	Social Influence	Sequential Effect	Local Preference
ST-LDA	•	•	•	•		•
GT-BNMF	•		•			•
Geo-SAGE	•		•			•
Rank-GeoFM	•	•				
LTSCR	•	•		•	•	
STM			•	•		
TopPop						•

$$\sum_{j=i}^A W(q, a_j) F(v, a_i), \quad (21)$$

where $\sum_{j=i}^A W(q, a_j)$ is the portion of query preferences for the rest attributes. The upper bound of $\sum_a W(q, a) F(v, a)$ for all attributes is $PS(q, v) + \sum_{j=i}^A W(q, a_j) F(v, a_i)$, which results in Equation 20.

We employ a binary min-heap to implement L so that the top POI v' has the smallest ranking score in L (Line 20). If the upper bound is smaller than the ranking score of v' , we skip the current POI (no need to check the remaining attributes) (Lines 25-28). Otherwise, we continue to check the remaining attributes. If all attributes are examined for the POI and the POI is not pruned by the aforementioned upper bound, we obtain the full score of the POI to compare with v' (Lines 30-35). We remove the POI v' and add the current POI to the list if its full score is larger than v' (Lines 31-34).

Time Complexity Analysis. Actually, the time cost of AP is dependent on the distribution of the query preference vectors \vec{q} and the POI vectors \vec{v} learnt in our ST-LDA model. If the query preference vector \vec{q} and POI vectors \vec{v} were dense and uniform distributions, AP would lose its powerful pruning ability and degenerate to the naive straightforward method in the worst case. However, after analyzing the learnt query and POI vectors in both Foursquare and Yelp datasets, we find that both query preference vectors and POI vectors are extremely sparse, i.e., each POI or query has significant values for only a small number of attributes, and most of its attribute values are extremely small. As shown in Section 6.5, AP exhibits strong attribute pruning ability in the real datasets. It only needs to access very few attributes for each POI to compute its partial score, e.g., less than 10 percent of attributes for $k = 10$ and less than 5 percent for $k = 5$.

6 EXPERIMENTS

In this section, we first describe the settings of experiments and then demonstrate the experimental results.

TABLE 5
Basic Statistics of Foursquare and Twitter Datasets

	Yelp	Foursquare
# of users	366,000	114,508
# of POIs	61,000	62,462
# of check-ins	1,569,264	1,434,668
time span	Jan 2006-Jan 2015	Sep 2010-Jan 2011

6.1 Experimental Settings

6.1.1 Datasets

Our experiments are conducted on two real-world datasets: Yelp and Foursquare. Their basic statistics are shown in Table 5.

Foursquare. Foursquare is one of the most popular online LBSNs. We collected its public check-in data from Sep. 2010 to Jan. 2011 through Twitter with the same crawling strategy as proposed in [12], [37]. This dataset contains the check-in history of 114,508 users who live in the USA. Each check-in record is stored as *user-ID*, *POI-ID*, *POI-location*, *POI-content*, *check-in time*. Note that this dataset does not contain user social network.

Yelp. The dataset is Yelp's Challenge Dataset,³ which contains 366,000 users and 61,000 POIs from 10 cities across four countries, including Edinburgh of United Kingdom, Karlsruhe of Germany, Montreal and Waterloo of Canada, and Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison of U.S.. There are 1.6 million check-in records, and each check-in record is stored as *user-ID*, *POI-ID*, *POI-location*, *POI-content*, *check-in date*. Each record in social networks is stored as *user-ID*, *friend-ID* and the total number of social relationship is 2.9 million. Note that this dataset does not contain the exact check-in time, and only provides the coarse check-in date (e.g., "2014-08-01").

6.1.2 Comparative Approaches

Recommendation Effectiveness. We first compare our ST-LDA with the following four state-of-the-art recommendation methods. Table 4 lists the characteristics of these methods.

GT-BNMF. GT-BNMF [28] is a Geographical-Topical Bayesian non-negative Matrix Factorization framework, which models the joint effect of users' interests, geographical influence and region-level POI popularity for POI recommendation.

Geo-SAGE. Geo-SAGE [7] is a geographical sparse additive generative model for predicting user check-in behaviors. This model considers both users' personal interests and the local preferences or attractions of the target region, by exploiting both the co-occurrence pattern of POIs and their contents.

Rank-GeoFM. Rank-GeoFM [38] is a ranking based geographical factorization method for POI recommendation, which incorporates both geographical and temporal influences.

LTSCR. LTSCR [21] is a location-time-aware social collaborative retrieval model for next POI recommendation,

3. http://www.yelp.com.sg/dataset_challenge/

which jointly considers the sequential effect, time-aware user preferences and temporal popularity of POIs. To make a fair comparison, we train two LTSCR models: one for home-town and one for out-of-town, since the information of home locations is not used in LTSCR.

STM. STM is a social-topic model proposed in [39] that recommends POIs according to the topic interests of his friends.

TopPop. TopPop is a strong baseline that recommends the most popular POIs to users within the target region.

To further validate the benefits brought by social-spatial collective inferring framework, distinguishing users' personal interests for different regions, discriminating between tourist preferences and native preferences, exploiting the temporal patterns of visiting POIs, and unifying topic discovery and region division (i.e., soft spatial clustering), respectively, we design five variant versions. *ST-LDA-V1* is the first variant version of ST-LDA where we do not use the social-spatial collective inferring framework to enhance the inference of users' region-dependent interests. *ST-LDA-V2* assumes that users' interests are region-independent, i.e., users' interests are stable across regions. This variant version is equivalent to the JIM model developed in [6]. *ST-LDA-V3* does not distinguish between tourist preferences and native preferences. *ST-LDA-V4* does not consider the check-in time information, and the latent variable z only generates the word set \mathcal{W}_v for each check-in record; and as the last variant, *ST-LDA-V5* adopts city-level administrative divisions as regions, and then infer city-dependent personal interests and the crowd's preferences.

Recommendation Efficiency. We compare our AP algorithm with two competitor algorithms and one baseline. The first competitor, developed in [36] to speed up the online recommendation, first uses a binary spatial partitioning metric tree (MT) to index the POI vectors and then utilizes a branch-and-bound algorithm to prune the POI search space. The second competitor is the threshold algorithm [9] developed for online recommendation. It pre-computes a sorted list for each latent attribute a in which POIs are sorted according to their values on attribute a , and also maintains a priority queue of the A sorted lists to control which sorted list to access in the next. The algorithm has the nice property of terminating early without scanning all POIs. The baseline is a brute-force algorithm (BF) that needs to compute a ranking score for each POI and selects top- k ones with highest ranking scores.

6.1.3 Evaluation Methods

Since ST-LDA is designed to support both home-town and out-of-town recommendation, we evaluate the recommendation effectiveness of our model under the two scenarios respectively. Given a user profile, namely a collection of check-in records, we first rank the check-in records in \mathcal{D}_u according to their check-in timestamps, and divide the user's check-ins into a training set and a test set. We adopt two different dividing strategies with respect to the two recommendation settings. For the out-of-town setting, we first choose the last non-home city visited by the user as the target city. Then, we select all POIs visited by the user in the target city as the test set, and use all of her check-in records generated before visiting the target city as the training set. For the home-town setting, we use the 80th percentile as the

cut-off point so that check-ins before this point are used for training, and the rest generated at her home city are chosen for testing. Besides, to simulate a real-time POI recommendation scenario, we have to choose a location as the user's current standing position before visiting v . Specifically, for each test case $(u, v, l_v, \mathcal{W}_v, t, s)$, the location l which is visited by u at time $t - 1$ is chosen. Thus, a query $q = (u, l, t, s)$ is formed for the test case.

According to the above dividing strategies, we split the dataset \mathcal{D} into the training set \mathcal{D}_{train} and the test set \mathcal{D}_{test} . To evaluate the recommendation methods, we adopt the metric Accuracy@ k proposed in [4], [7]. Specifically, for each check-in record $(u, v, l_v, \mathcal{W}_v, t, s)$ in \mathcal{D}_{test} :

- (1) We compute the ranking score for POI v and all other POIs unvisited by user u previously.
- (2) We form a ranked list by ordering all of these POIs according to their ranking scores. Let p denote the position of the POI v within this list. The best result corresponds to the case where v precedes all the unvisited POIs (that is, $p = 1$).
- (3) We form a top- k recommendation list by picking the k top ranked POIs from the list. If $p \leq k$, we have a hit (i.e., the ground truth v is recommended to the user). Otherwise, we have a miss.
- (4) The computation of Accuracy@ k proceeds as follows. We define hit@ k for a single test case as either the value 1, if the ground truth POI 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:

$$Accuracy@k = \frac{\#hit@k}{|\mathcal{D}_{test}|},$$

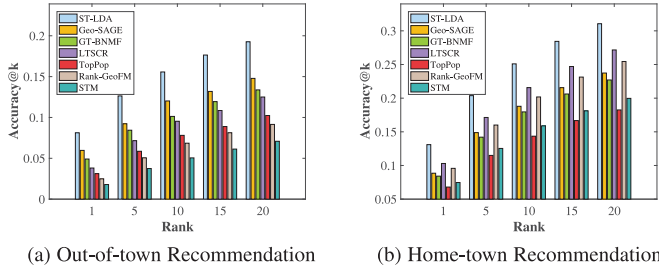
where $\#hit@k$ denotes the number of hits in the test set, and $|\mathcal{D}_{test}|$ is the number of all test cases.

Both the Yelp and Foursquare datasets have a low density (i.e., the densities of user-item matrix are 0.007 percent and 0.02 percent for Yelp and Foursquare datasets, respectively), which usually results in relatively low accuracy values [11]. In addition, the POIs in the test data of each user may represent only a small portion of POIs that the user is truly interested in. Thus, the relatively low accuracy values are common and reasonable. In this paper, we focus on the relative improvements we achieved, instead of the absolute values.

6.2 Recommendation Effectiveness

In this subsection, we report the comparison results between our proposed model ST-LDA and other competitor methods with well-tuned parameters. Figs. 3 and 4 report the performance of the recommendation methods on the Foursquare and Yelp datasets, respectively. We only show the performance where k is set to 1, 5, 10, 15, 20, as a greater value of k is usually ignored for a typical top- k recommendation task.

Out-of-Town Recommendation On Foursquare. Fig. 3(a) presents the recommendation accuracy in the scenario of out-of-town recommendation, where the accuracy of ST-LDA is about 0.148 when $k = 10$, and 0.183 when $k = 20$ (i.e., the model has a probability of 14.8 percent of placing an appealing POI in the top-10 and 18.3 percent of placing it

Fig. 3. Top- k performance on Foursquare dataset.

in the top-20). Clearly, our proposed ST-LDA model outperforms other competitor models significantly in the out-of-town setting, and the improvements, in terms of $Accuracy@10$, are 27.79, 38.42, 72.13, 97.51, 110.68, and 180.87 percent compared with Geo-SAGE, GT-BNMF, LTSCR, TopPop, Rank-GeoFM and STM, respectively. Several observations are made from the results: 1) ST-LDA achieves much higher recommendation accuracy than Geo-SAGE and GT-BNMF, showing the benefits of distinguishing users' interests for different regions due to user interest drift across geographical regions. 2) Both ST-LDA and Geo-SAGE outperform GT-BNMF. This may be because only ST-LDA and Geo-SAGE further discriminate between native preferences and tourist preferences, although all the three methods exploit the preferences of the crowds w.r.t. a specific region to adapt to the drift of users' interests. 3) Rank-GeoFM and LTSCR drop behind other three methods, showing the advantages of exploiting the content information of users' visited POIs to capture their interests. Through the medium of content, these methods can transfer the users' general interests inferred at home town to out-of-town regions, which alleviates the data sparsity at out-of-town regions to some extent. 4) Users do not have any activity history in most of their out-of-town regions, and their check-ins generated at home town are not helpful for LTSCR and Rank-GeoFM to predict their behaviors at out-of-town regions. As the co-occurrence frequency of two POIs (i.e., co-visited by the same users) located at two different regions is much lower than that of two POIs at the same region, LTSCR and Rank-GeoFM are not capable of linking POIs located at different regions together.

Home-Town Recommendation On Foursquare. In Fig. 3b, we report the performance of all recommendation models for the home-town scenario, and our ST-LDA achieves the highest recommendation accuracy. From the results, we observe that the recommendation accuracies of all methods are higher in Fig. 3b than that in Fig. 3a. Besides, Geo-SAGE, and GT-BNMF outperform LTSCR and Rank-GeoFM in Fig. 3a while

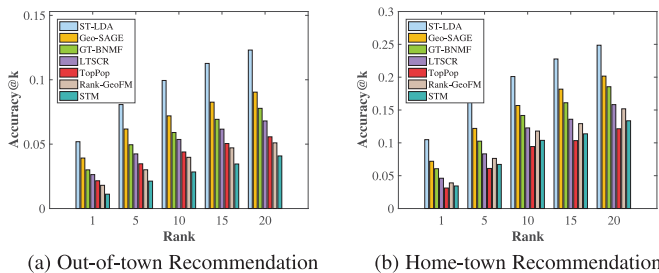
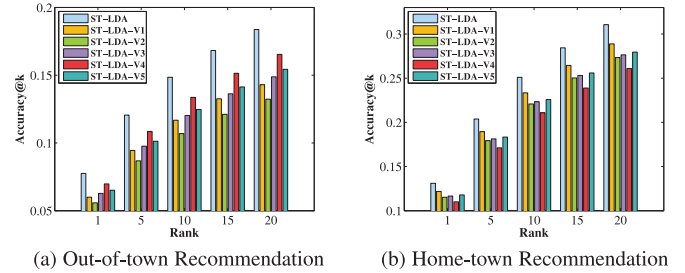
Fig. 4. Top- k performance on Yelp dataset.

Fig. 5. Impact of different factors on Foursquare dataset.

LTSCR and Rank-GeoFM slightly exceed Geo-SAGE and GT-BNMF in Fig. 3b, showing that the temporal and sequential influences play a more important role than the content effect in the home-town recommendation setting where the user-POI matrix is not sparse. Our proposed ST-LDA distinguishes users' interests for different regions and has the comprehensive modeling ability to take advantage of different dimension information to alleviate the issue of data sparsity, thus it performs well consistently in both recommendation settings. The comparison between Fig. 3a and Fig. 3b also reveals that the two recommendation scenarios are intrinsically different, and should be separately evaluated.

Recommendation on Yelp. Fig. 4 reports the performance of the recommendation models on the Yelp dataset. From the figure, we can see that the trend of comparison result is similar to that presented in Fig. 3, and the main difference is that all recommendation methods achieve lower accuracy. This may be because users' check-in data on this dataset is more sparse than the Foursquare dataset (0.007 versus 0.02 percent). Another difference is that LTSCR and Rank-GeoFM perform better than Geo-SAGE and GT-BNMF in Fig. 3b while Geo-SAGE and GT-BNMF outperform LTSCR and Rank-GeoFM in 4(b), because the Yelp dataset does not contain the exact check-in time, and only the check-in date is provided. This seriously impedes the discovery of both temporal and sequential patterns of users' check-in activities.

6.3 Impact of Different Factors

In this subsection, we carry out an ablation study showing the benefits of each factor in ST-LDA, i.e., social-spatial collective inferring of users' interests (F1), distinguishing users' personal interests for different regions (F2), discriminating between tourist preferences and native preferences (F3), exploiting the temporal patterns of visiting POIs (F4), and unifying topic discovery and region division (or soft spatial clustering) (F5). We compare ST-LDA with its five variant versions proposed in Section 6.1.2, and the comparison results are shown in Figs. 5 and 6. Note that we do not

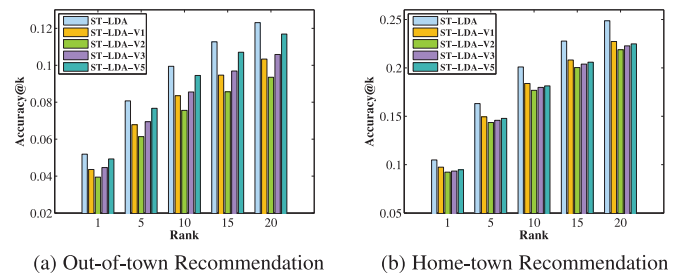


Fig. 6. Impact of different factors on Yelp dataset.

TABLE 6
Out-of-Town Recommendation Accuracy@10

$R \backslash K$	K=30	K=40	K=50	K=60	K=70	K=80
R=50	0.115	0.123	0.128	0.131	0.131	0.131
R=60	0.123	0.131	0.137	0.140	0.140	0.140
R=70	0.127	0.135	0.141	0.144	0.144	0.144
R=80	0.131	0.140	0.146	0.148	0.148	0.148
R=90	0.131	0.140	0.146	0.148	0.148	0.149
R=100	0.131	0.140	0.146	0.148	0.148	0.149

compare with ST-LDA-V4 on the Yelp dataset since the exact check-in time (e.g., the hour) is not available in this dataset.

From the results, we first observe that ST-LDA consistently outperforms the five variant versions for both out-of-town and home-town recommendation, indicating the benefits brought by each factor, respectively. For instance, the performance gap between ST-LDA and ST-LDA-V2 validates the benefit of distinguishing users' interests for different regions, especially in the out-of-town recommendation. Second, we find that the contribution of each factor to improving recommendation accuracy is different. Besides, the contributions of the same factor are different in the two different recommendation scenarios. Specifically, according to the importance of the five factors in the out-of-town recommendation scenario, they can be ranked as follows: $F2 > F1 \geq F3 > F5 > F4$, while in the home-town recommendation scenario they can be ranked as $F4 > F2 \geq F3 \geq F5 > F1$. This is because the two recommendation scenarios have different characteristics: 1) most of users have enough check-in records in their home towns while few check-in activities are left in out-of-town regions; 2) the limitation of travel distance (i.e., the geographical influence) in the out-of-town scenario does not matter as much as that in home town; 3) users' daily routines may change when they travel out of town. Obviously, distinguishing users' interests w.r.t. regions and social-spatial collective inferring play dominant roles in improving out-of-town recommendation. Another observation is that the performance gap between our ST-LDA model and the five variants in the task of home-town recommendation is smaller than that in the task of out-of-town recommendation, showing that the performance differences among these methods become less significant when people travel in home town.

6.4 Analysis of Parameter Sensitivity

In this experiment, we study the impact of varying parameters in ST-LDA on the Foursquare dataset, e.g., the number of topics (K) and the number of regions (R), and report the results in Tables 6 and 7. As for the hyperparameters α , β , γ , η and τ , for simplicity, we take a fix value, i.e., $\alpha = 50/K$, $\gamma = 50/R$ and $\beta = \eta = \tau = 0.01$, following the studies [4], [31]. We try different setups and find that the performance of ST-LDA model is not sensitive to these hyperparameters, but its performance is sensitive to the number of topics and regions.

From the results, we observe that the recommendation accuracy of ST-LDA first increases with the increasing number of topics, and then it does not change significantly when the number of topics is larger than 60. Similar observation is

TABLE 7
Home-Town Recommendation Accuracy@10

$R \backslash K$	K=30	K=40	K=50	K=60	K=70	K=80
R=50	0.194	0.208	0.216	0.221	0.221	0.221
R=60	0.208	0.222	0.231	0.236	0.236	0.236
R=70	0.214	0.229	0.239	0.243	0.244	0.244
R=80	0.221	0.236	0.246	0.251	0.251	0.251
R=90	0.221	0.236	0.246	0.251	0.251	0.252
R=100	0.221	0.236	0.246	0.251	0.252	0.252

made for increasing the number of regions (i.e., R): the recommendation accuracy of ST-LDA increases with the increasing number of regions, and then it does not change much when the number of regions is larger than 80. The reason is that K and R represent the model complexity. Thus, when K and R are too small, the model has limited ability to describe the data. On the other hand, when K and R exceed a threshold, the model is comprehensive enough to handle the data. At this point, it is less helpful to improve the model performance by increasing K and R .

We conducted another experiment to test the recommendation accuracy by varying K and R in a larger scale (i.e., from 100 to 1,000), and observed that when K and R are larger than 300, the recommendation accuracy of ST-LDA begins to decrease with further increasing K and R . The reason is that when R and K are too large, the check-in data used to infer or estimate the model parameters associated with a specific region or topic becomes extremely sparse, which easily leads to the overfitting and makes the learnt model parameters unreliable. Due to the space constraint, we do not present this result.

6.5 Recommendation Efficiency

This experiment is to evaluate the efficiency of our proposed online POI recommendation algorithm AP on the Foursquare dataset. We compare it with two competitor algorithms and one baseline proposed in Section 6.1.2. All the online recommendation algorithms were implemented in Java (JDK 1.7) and run on a Windows Server 2008 with 256G RAM.

Table 8 presents the average online efficiency of the four different methods over all queries created for \mathcal{D}_{test} on the Foursquare dataset. We show the performance with 4800 latent dimensions (i.e., $A = 4800$) and k set to 1, 5, 10, 15, and 20. A greater value of k is not necessary for the top- k recommendation task. Obviously, our proposed AP algorithm outperforms others significantly and consistently for different number of recommendations. For example, on average our proposed AP algorithm finds the top-10

TABLE 8
Recommendation Efficiency on Foursquare Dataset

Methods	Online Recommendation Time Cost (ms)				
	k=1	k=5	k=10	k=15	k=20
AP	50.91	61.75	67.57	72.13	75.32
BF	110.83	111.40	111.81	111.95	112.94
MT	130.90	131.46	131.87	132.39	132.99
TA	880.92	1055.68	1144.82	1221.49	1265.58

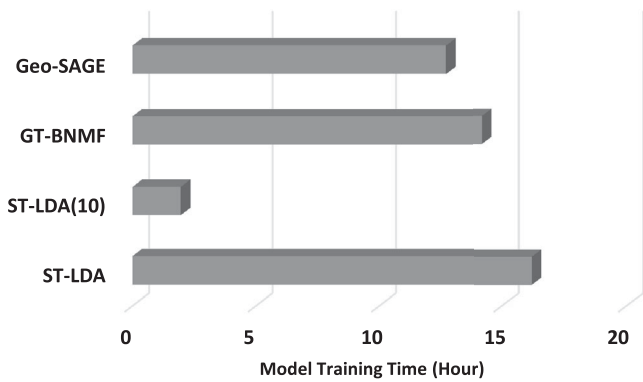


Fig. 7. Training time. “ST-LDA (10)” is the distributed implementation on 10 nodes.

recommendations from about 62,000 POIs in 67.57 ms, and achieves 1.65 times faster than the brute-force algorithm.

Specifically, from the results, we observe that: 1) AP outperforms BF significantly, justifying the benefits brought by pruning attribute space. It only needs to access very few attributes for each POI to compute its partial score, about 465 attributes on average (that is less than 10 percent) for $k = 10$, and 220 attributes for $k = 5$, since AP algorithm takes full advantage of the sparsity of both query and POI vectors. 2) Although the time cost of AP increases with the increasing number of recommendations (i.e., k), it is still much lower than that of BF in the recommendation task even when $k = 20$. 3) The time cost of MT is higher than that of the naive linear scan method (BF) in our task, although Koenigstein et al. [36] reported the good performance of the Metric Tree in the setting with 50 attributes. This is because MT loses its ability to prune POI search space and needs to scan all POIs in the leaf nodes when the dimensionality is high. 4) The threshold algorithm performs worse than the brute-force algorithm, since it still needs to access many POIs (around 40 percent of the POIs on average for $k = 10$ and 35 percent for $k = 5$). Moreover, TA needs to frequently update the threshold for each access of sorted lists and to maintain the dynamic priority queue of sorted lists. These extra computations reduce down the efficiency of TA when the dimensionality is high. In summary, although both TA and MT can achieve better performance than BF due to their ability of pruning POI search space when the dimensionality is not very high (e.g., less than 500), they cannot overcome the curse of dimensionality when the objects have thousands of attributes. In contrast, our AP algorithm is designed for pruning attribute space, thus it can still achieve superior performance for the setting of high dimensionality.

We also study the efficiency of model training on the Foursquare dataset. We deploy the inference algorithm of ST-LDA on the distributed GraphLab system to tackle the challenge of large data size. Fig. 7 shows the running time of different model training. Note that while ST-LDA jointly models text, geographical location, time and network information, other models generally only accounts for a limited portion of the data. Though the basic implementation of ST-LDA is costly, the parallel implementation guarantee the efficiency of model training in the actual deployment. We reduce the training time for 1.4 million check-ins from 16.2 hours to 1.9 hours. This clearly shows the advantage of parallel processing. The model structure of ST-LDA is loosely coupled enough to guarantee the parallel processing, and it is feasible in the actual deployment.

6.6 Analysis of Latent Topics

In this section, we qualitatively compare the latent topics discovered by ST-LDA and Twitter-LDA [40] which is designed to discover topics from short documents.

Table 9 shows four latent topics discovered by ST-LDA and Twitter-LDA. For each topic, we present the top six words with the highest generation probabilities. By comparing the topics in Table 9, we observe that the words in each latent topic learned by ST-LDA are semantically coherent, indicating the category or function of POIs. In contrast, topics estimated by Twitter-LDA look confusing and noisy, and several different kinds of semantics (e.g., categories) are mixed up in one topic. For example, Topic 1 is about “home”; Topic 2 is about “shopping”; Topic 3 describes “eating” and “restaurant” and Topic 4 indicates “night life” activity. In contrast, topics estimated by Twitter-LDA look confusing and noisy, and several different kinds of semantics (e.g., categories or concepts) are mixed up in one topic. For example, “home” and “eating” are mixed up in Topic 1, and “restaurant” and “entertainment” are mixed up in Topic 3. The comparison results show that it actually improves the discovered topic quality by introducing the check-in time to the topic discovery process, since the POIs with the same categories and functions are more likely to have the same or similar temporal patterns of check-ins.

To analyze the temporal patterns of people’s activities captured by the discovered topics in ST-LDA, we plot the time distribution of these four topics in the 24-hours time scale in Fig. 8, where horizontal axis denotes the time slots and vertical axis represents the normalized check-in number. From the figure, we observe four different time distribution patterns that correspond to four kinds of POIs, which reflect human’s daily activity patterns. For example,

TABLE 9
Latent Topic Analysis on the Foursquare Dataset

Topics Discovered by ST-LDA				Topics Discovered by Twitter-LDA			
Topic 1	Topic 2	Topic 3	Topic 4	Topic 1	Topic 2	Topic 3	Topic 4
home	shopping	restaurant	pub	home	shopping	restaurant	entertainment
residence	supermarket	eating	bar	eating	food	bakery	pub
apartment	store	dinner	nightclub	lunch	zoo	club	gym
condo	mall	breakfast	nightlife	office	park	shop	playground
private	market	pizza	entertainment	tea	store	bakery	drink
building	plaza	sandwich	dancestudio	coffee	pharmacy	pub	aquarium

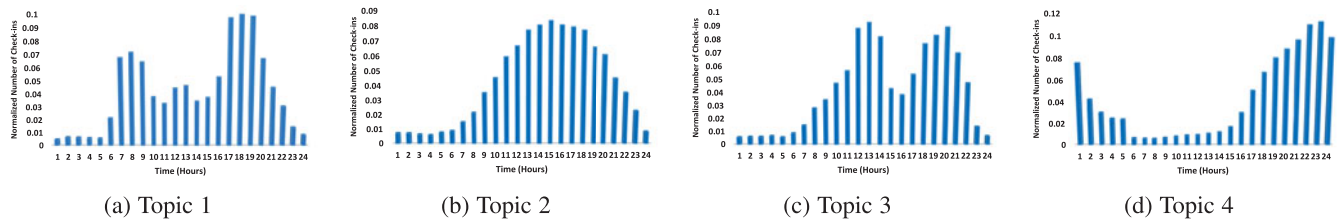


Fig. 8. Topic distribution over time.

there are clearly two peak times for Topic 3, corresponding to lunch and dinner periods. On the other hand, shopping time (i.e., the time distribution of Topic 2) looks like a normal distribution with most activities between 9:00 am and 10:00 pm, and there is no obvious peak shopping time. Fig. 8d indicates that many users visit nightlife spots (e.g., bars and pubs) at midnight. These observations regarding the temporal patterns of the detected topics provide strong support that check-in time distribution in 24-hours time scale is a good feature to improve the topic discovery. Thus, the topics discovered by our proposed ST-LDA not only capture the semantic information of POIs, but also enable the temporal analysis of human's daily activity patterns.

7 CONCLUSION

To adapt to user interest drift across geographical regions and improve mobile POI recommendation, we proposed a probabilistic generative model ST-LDA to learn region-dependent personal interests. ST-LDA incorporated the crowd's preferences to alleviate the extreme sparsity of the users' check-in records left in the out-of-town regions by exploiting the public's visiting behaviors at the target region. Besides, a social-spatial smoothing framework was developed to enhance the inference of region-dependent personal interests by exploiting the social and spatial correlation information. Based on ST-LDA, we designed an efficient attribute pruning algorithm to overcome the curse of dimensionality and support fast online recommendation for large-scale geo-social data. We conducted extensive experiments to evaluate the performance of our ST-LDA model and AP algorithm in terms of both recommendation effectiveness and efficiency. The results showed the superiority of our proposals over other competitor methods. Besides, we carried out an ablation study to show benefits of each factor in ST-LDA, and found that *distinguishing user interests w.r.t. regions* plays a dominant role in improving mobile POI recommendation.

ACKNOWLEDGMENTS

This work was supported by ARC Discovery Early Career Researcher Award (Grant No. DE160100308) and ARC Discovery Project (Grant No. DP140103171). It was also partially supported by the National Natural Science Foundation of China (Grant No. 61472263, 61572335, 61303164, 61572039), Beijing Natural Science Foundation (Grant No. 4152023), Jiangsu Natural Science Foundation of China (Grant No. BK20151223), 973 program under No. 2014CB340405, Shenzhen Gov Research Project (JCYJ20151014093505032), and Special Funds for Outstanding Youth Development (Grant No. ISCAS2014-JQ02).

REFERENCES

- [1] G. Ferenc, M. Ye, and W.-C. Lee, "Location recommendation for out-of-town users in location-based social networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2013, pp. 721–726.
- [2] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "Lars: A location-aware recommender system," in *Proc. Int. Conf. Data Eng.*, 2012, pp. 450–461.
- [3] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks," in *Proc. Int. Conf. Weblogs Soc. Media*, 2011, pp. 329–336.
- [4] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: A location-content-aware recommender system," in *Proc. Knowl. Discovery Data Min.*, 2013, pp. 221–229.
- [5] B. Liu and H. Xiong, "Point-of-interest recommendation in location based social networks with topic and location awareness," in *Proc. SDM*, 2013, pp. 396–404.
- [6] H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq, "Joint modeling of user check-in behaviors for point-of-interest recommendation," in *Proc. Conf. Inf. Knowl. Manage.*, 2015, pp. 1631–1640.
- [7] 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 *Proc. Knowl. Discovery Data Min.*, 2015, pp. 1255–1264.
- [8] B. Hu and M. Ester, "Spatial topic modeling in online social media for location recommendation," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 25–32.
- [9] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, "Lcars: A spatial item recommender system," *ACM Trans. Inf. Syst.*, vol. 32, no. 3, pp. 11:1–11:37, Jul. 2014.
- [10] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proc. Knowl. Discovery Data Min.*, 2011, pp. 1082–1090.
- [11] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proc. Spec. Int. Group Inf. Retr.*, 2013, pp. 363–372.
- [12] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services," in *Proc. Int. Conf. Weblogs Soc. Media*, 2011, pp. 81–88.
- [13] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. Spec. Int. Group Inf. Retr.*, 2011, pp. 325–334.
- [14] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei, "Evaluating geo-social influence in location-based social networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2012, pp. 1442–1451.
- [15] W.-Y. Zhu, W.-C. Peng, L.-J. Chen, K. Zheng, and X. Zhou, "Modeling user mobility for location promotion in location-based social networks," in *Proc. Knowl. Discovery Data Min.*, 2015, pp. 1573–1582.
- [16] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. Am. Assoc. Artif. Intell.*, 2012, pp. 17–23.
- [17] 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 *Proc. Knowl. Discovery Data Min.*, 2014, pp. 831–840.
- [18] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 93–100.
- [19] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2605–2611.
- [20] J.-D. Zhang, C.-Y. Chow, and Y. Li, "Lore: Exploiting sequential influence for location recommendations," in *Proc. SIGSPATIAL*, 2014, pp. 103–112.

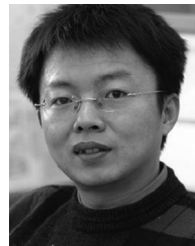
- [21] W. Zhang and J. Wang, "Location and time aware social collaborative retrieval for new successive point-of-interest recommendation," in *Proc. Conf. Inf. Knowl. Manage.*, 2015, pp. 1221–1230.
- [22] H. Cheng, J. Ye, and Z. Zhu, "What's your next move: User activity prediction in location-based social networks," in *Proc. SDM*, 2013, pp. 171–179.
- [23] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou, "Spore: A sequential personalized spatial item recommender system," in *Proc. Int. Conf. Data Eng.*, 2016, pp. 121–133.
- [24] K. Zhao, G. Cong, Q. Yuan, and K. Zhu, "Sar: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews," in *Proc. Int. Conf. Data Eng.*, 2015, pp. 675–686.
- [25] H. Yin, B. Cui, Z. Huang, W. Wang, X. Wu, and X. Zhou, "Joint modeling of users' interests and mobility patterns for point-of-interest recommendation," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 819–822.
- [26] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, "Joint modeling of user check-in behaviors for real-time point-of-interest recommendation," *ACM Trans. Inf. Syst.*, vol. 34, no. 5, pp. 3:1–3:41, Jun. 2016.
- [27] 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 *Knowl. Discovery Data Min.*, 2012, pp. 1023–1031.
- [28] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Knowl. Discovery Data Min.*, 2013, pp. 1043–1051.
- [29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [30] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsoutsoulis, "Discovering geographical topics in the twitter stream," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 769–778.
- [31] Z. Hu, J. Yao, B. Cui, and E. Xing, "Community level diffusion extraction," in *Proc. Spec. Int. Group Manage. Data*, 2015, pp. 1555–1569.
- [32] S. A. Macskassy, "Improving learning in networked data by combining explicit and mined links," in *Proc. Am. Assoc. Artif. Intell.*, 2007, pp. 590–595.
- [33] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz, "On the semantic annotation of places in location-based social networks," in *Proc. Knowl. Discovery Data Min.*, 2011, pp. 520–528.
- [34] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. Web Search Data Min.*, 2010, pp. 241–250.
- [35] C. Teflioudi, R. Gemulla, and O. Mykytiuk, "Lemp: Fast retrieval of large entries in a matrix product," in *Proc. Spec. Int. Group Manage. Data*, 2015, pp. 107–122.
- [36] N. Koenigstein, P. Ram, and Y. Shavitt, "Efficient retrieval of recommendations in a matrix factorization framework," in *Proc. Conf. Inf. Knowl. Manage.*, 2012, pp. 535–544.
- [37] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proc. Int. Conf. Web Soc. Media*, 2012, pp. 114–121.
- [38] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geofm: A ranking based geographical factorization method for point of interest recommendation," in *Proc. Spec. Int. Group Inf. Retr.*, 2015, pp. 433–442.
- [39] B. Hu and M. Ester, "Social topic modeling for point-of-interest recommendation in location-based social networks," in *Proc. Int. Conf. Data Min.*, 2014, pp. 845–850.
- [40] 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 *Proc. Euro. Conf. Inf. Retr.*, 2011, pp. 338–349.



Hongzhi Yin received the PhD degree in computer science from Peking University in 2014. His research interests include recommender system, user profiling, topic models, deep learning, social media mining, and location-based services. He has published over 30 papers in the most prestigious journals and conferences such as SIGMOD, KDD, VLDB, ICDE, the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Information Systems*, the *ACM Transactions on Intelligent Systems and Technology*, and the *ACM Transactions on Knowledge Discovery from Data*. Besides, he has one monograph published by Springer. He is an ARC DECRA (Australia Research Council Discovery Early-Career Researcher Award) Fellow with the University of Queensland.



information systems, data mining, bioinformatics, and e-research.



Bin Cui is a professor in the School of EECS and director of Institute of Network Computing and Information Systems, Peking University. His research interests include database system architectures, query and index techniques, big data management, and mining. He is currently on the editorial boards of the *VLDB Journal*, *Distributed and Parallel Databases Journal*, *Information Systems*, and *Frontier of Computer Science*. He is the winner of the Microsoft Young Professorship award and the CCF Young Scientist award.



Hao Wang received the PhD degree from the University of Tokyo. He is an associate professor in the Institute of Software, Chinese Academy of Sciences. His research interests include social media mining, social network analysis, and sentiment analysis.



Kai Zheng received the PhD degree in computer science from the University of Queensland in 2012. He is a professor in the School of Computer Science, Soochow University. He has been working in the area of social-media analysis, spatial-temporal databases, uncertain databases, data mining, and bioinformatics. He has published more than 50 papers in the most prestigious journals and conferences such as SIGMOD, ICDE, EDBT, the *VLDB Journal*, *ACM Transactions*, and *IEEE Transactions*.



Quoc Viet Hung Nguyen received the PhD degree from the Swiss Federal Institute of Technology, Lausanne, in 2014. He is a postdoctoral research fellow with the University of Queensland. His research interests include data integration, data extraction, data curation, and knowledge discovery.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.