Joint Modeling of User Check-in Behaviors for Point-of-Interest Recommendation

ABSTRACT

Point-of-Interest (POI) recommendation has become an important means to help people discover attractive and interesting locations, especially when users travel out of town. However, extreme sparsity of user-POI matrix creates a severe challenge. To cope with this challenge, a growing line of research has exploited the temporal effect, geographical-social influence, content effect and word-of-mouth effect. However, current research lacks an integrated analysis of the joint effect of the above factors to deal with the issue of data-sparsity, especially in the out-of-town recommendation scenario which has been ignored by most of existing work.

In light of the above, we propose a joint probabilistic generative model JIM to model users' check-in activities in LBSNs, which strategically integrates the above factors to effectively overcome the data sparsity and improve recommendation results, especially for out-of-town users. To demonstrate the applicability and flexibility of JIM, we investigate how it supports two recommendation scenarios in a unified way, i.e., home-town recommendation and out-of-town recommendation. We conduct extensive experiments to evaluate the performance of JIM on two real large-scale datasets in terms of both recommendation effectiveness and efficiency, and the experimental results show superiority of JIM over other competitors. Besides, we study the importance of each factor in the two recommendation scenarios, respectively, and find that exploiting temporal effect is most important for the home-town recommendation scenario, while the content information plays a dominant role in improving the recommendation effectiveness for out-of-town users.

1. INTRODUCTION

Recent years have witnessed the fast development of location-based social networks (LBSNs), such as Foursquare and Facebook Places, due to the advances in location-acquisition and wireless communication technologies. In these LBSNs, users can post their physical locations or geo-tagged information in the form of "check-in", and share their visiting experiences and tips for points of interest (POI) with friends, such as restaurants, sightseeing sites. In LBSNs, it is crucial to utilize user check-in data to make personalized POI recommendation, which help users know new POIs and explore new regions (e.g., cities), facilitate advertisers to launch mobile advertisements to targeted users, and make LBSNs more attractive to users and advertisers.

Recently, POI recommendation has become a popular research topic due to easy access of large-scale check-in records. One of the most important problems 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 [16] made on LBSNs shows that most of users' check-ins are left in their living regions (e.g., home cities) due to the distance constraint. An investigation shows that the check-in records generated by users in their non-home cities are very scarce and only take up 0.47% of the check-in records left in their home cities, which 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) [11, 31].

The most popular approach in recommender systems is that of *collaborative filtering* [1]. There exists a considerable body of research [16, 28, 18, 11, 12] which deposited people's check-in history into user-POI matrix where each row corresponds to a user's POI-visiting history and each column denotes a POI. A collaborative filtering-based method is then employed by [16, 28, 11] to infer the user's preference regarding each unvisited POI. Based on the core idea of collaborative filtering, similar users of the target user (i.e., those who exhibit similar POI visiting behaviors) are chosen to provide clues for making recommendation. Due to travel locality, most of these similar users are more likely to live in the same region with the target user than other regions. As a recommendation is made by considering POIs visited by the similar users, most of the recommended POIs would be located in the target user's home town. So, these CF-based methods cannot be directly applied to the POI recommendation for out-of-town users [11, 31].

Moreover, unlike the traditional GPS trajectories [37, 36], the time-ordered check-in records of a user are low-sample-rate where the details of movement information are lost. The spatial gap between any two consecutive check-ins is typically in the scale of kilometers while the spatial gap between consecutively logged GPS points in the GPS trajectories is typically 5-10 meters. Besides, the time interval between consecutive check-ins is much larger than that in the GPS trajectories. Therefore, existing sequential pattern mining methods such as Markov-chain models on GPS trajectories cannot apply to the sparse users' check-in data in LBSNs.

To deal with the issue of data sparsity, especially for out-of-town recommendation scenario, we explore the following factors which can influence the decision-making process of a user choosing a POI in a joint manner.

- Geographical Influence. Many recent studies show that peo-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

ple tend to explore nearby POIs of a POI that they have visited before [29]. So, POIs visited by users often form spatial clusters, i.e., people tend to check in around several centers (e.g., "home" and "office") [6, 18].

- Temporal Effect. As observed in [32, 12], human mobility exhibits strong temporal cyclic patterns in terms of hour of the day or day of the week. For example, a user is more likely to go to a restaurant rather than a bar at lunch time, and is more likely to go to a bar rather than an office at midnight.
- Content Effect. Content information on LBSNs related to a user's check-in activity provides a unique opportunity to infer personal interests. For example, by observing a POI's description as "vegetarian restaurant", we infer that users who check-in at this POI might be interested in vegetarian diet.
- Word-of-Mouth Effect. The region-level popularity of POIs also affects user visiting behaviors [31, 20]. In fact, the probability of a user visiting a POI is largely affected by the local word-of-mouth about the POI, especially when users travel in unfamiliar regions.

While there are some studies that exploit one of the above factors to improve POI recommendation effectiveness, they lack an integrated analysis of their joint effect to deal with the issue of data sparsity, especially in the out-of-town recommendation scenario. Specifically, most prior work of POI recommendation focuses on exploiting users' mobility patterns by investigating geographical influence [29, 6, 18], temporal effects [32, 12], and social influence [11, 29, 10], but they ignore the potential effect of content information of checked-in POI, thus missing the opportunity to transfer user interest inferred at home town to other out-of-town regions by the medium of POI contents. Although some recent work [31, 13, 21] exploited the content information of POIs to deal with the issue of data sparsity, they do not consider the temporal effect and geographical influence. In light of this, we propose a Joint probabilistic generative model JIM to model users' check-in behaviors in LBSNs, which strategically integrates the above factors to effectively overcome the issue of data sparsity, especially in the out-of-town recommendation scenario. There are two components in JIM: Time-aware User Interest Component (TIC) and Popularity-aware User Mobility Component (PMC).

TIC aims to exploit both the contents of POIs and their temporal effect. Specifically, we infer individual user's interest according to the contents of his/her checked-in POIs. Thus, TIC alleviates the data sparsity, especially for out-of-town recommendation users, as the contents play the role of medium which can transfer users' interests inferred at their home regions to out-of-town regions where they are traveling. Unfortunately, based on recent analysis of LBSNs data [27], about 30% of all POIs are lacking any meaningful textual descriptions. To address this problem, we exploit the association between the contents of checked-in POIs and the checking-in time by mining the co-occurrence patterns of user activity contents and the activity time, since the check-in time provides important clues about the content of POIs.

PMC is developed to exploit the geographical influence and word-of-mouth effect. Different from user online behaviors in the virtual world, user activities in the physical world are limited by travel distance. In this component, we first infer each user's activity range according to the location distribution of his/her historical visited POIs or his/her current location. Specifically, we divide the geographical space into several regions and compute the probability of individual user visiting each region. Then, we infer the probability of each region generating a POI according to both the public's check-in behaviors at that region and the distance between the POI and the center of that region, i.e., considering both the region-level popularity of the POI and the geographical influence. By integrating the region-level word-of-mouth effect, i.e., the wisdom of crowds, JIM model can solve the problem of user interest drift across geographical regions which indicates that user interests inferred at one region (e.g., home town) cannot always be applied to recommendation at another region. For example, a user u never goes gambling when he lives in Beijing, China, but when he travels in Macao or Las Vegas he is most likely to visit casinos.

To demonstrate the applicability of JIM, 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 his/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 personalized, time-aware [32] and location-based [11], i.e., to recommend different ranked lists of POIs for the same target user at different time and locations.

We conduct extensive experiments to evaluate the performance of our JIM model on two real large-scale datasets in terms of recommendation effectiveness and efficiency, and the results show superiority of JIM model over other competitors. Besides, we study the contribution of each factor to improve the recommendation results in the two respective scenarios under a unified recommendation framework, and find that the *content effect* plays a dominant role to alleviate the data sparsity in the out-of-town recommendation scenario, while the *temporal effect* is most important to improve home-town recommendation. To the best of our knowledge, this is the first work to compare the effect of each factor in the two different recommendation scenarios.

The remainder of the paper is organized as follows. Section 2 details JIM model. We present how to effectively and efficiently deploy JIM model to the POI recommendation in Section 3. We report the experimental results in Section 4. Section 5 reviews the related work and we conclude the paper in Section 6.

2. JOINT MODELING OF USER CHECK-IN BEHAVIORS

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

2.1 Preliminary

For the ease of presentation, we define the key data structures and notations used in this paper.

Name: Darling Harbour
Location: Longitude: 151.200, Latitude: -33.877
Categories: Harbor, Marina, Park, and Plaza
Tags: scenic views, harbourside, fireworks, cafés sunsets, harbors, museums, playground
Total check-ins: 27,692

Table 1: A POI and its associated information

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 contents. We use v to represent a POI identifier and l_v to denote its corresponding geographical attribute in terms of longitude and latitude coordinates. It should be noted that many different POIs can share the same geographical coordinate. Besides, there may be textual semantic information associated with a POI, such as the category and tag words. We use the notation W_v to denote the set of words describing POI v. Table 1 shows an example.

Variable	Interpretation
$\boldsymbol{\vartheta_u}$	the activity range of user u, expressed by a multinomial distribution over a set of regions
$\boldsymbol{\theta_u}$	the interests of user u, expressed by a multinomial distribution over a set of topics
$\boldsymbol{\phi_z}$	a multinomial distribution over words specific to topic z
$\boldsymbol{\psi_z}$	a beta distribution over time specific to topic z
$\boldsymbol{\varphi_r}$	the region-level popularity distribution of POIs specific to region r
$\boldsymbol{\mu_r}$	the mean location of region r
$\boldsymbol{\Sigma_r}$	the location covariance of region r
$\gamma, \alpha, \beta, \tau$	Dirichlet priors to multinomial distributions $\boldsymbol{\vartheta_u}$, $\boldsymbol{\theta_u}$, $\boldsymbol{\phi_z}$ and $\boldsymbol{\varphi_r}$, respectively

Table 2: Notations of parameters

Definition 2. **(User Home Location)** Following the recent work of [17], given a user u, we define the user's home location as the place where the user lives, denoted as l_u.

Note that, we assume a user's home location is "permanent" in our problem. In other words, a home location is a static location instead of a real-time location that is "temporally" related to him (e.g., the places where he/she is traveling). Due to privacy, user home locations are not always available. For a user whose home location is not explicitly given, we adopt the method developed by [23] which discretizes the world into $25km$ by $25km$ cells and defines the home location as the average position of check-ins in the cell with most of his/her check-ins.

Definition 3. **(Check-in Activity)** A user check-in activity is represented by a five tuple $(u,\ v,\ l_v,\ W_v,\ t)$ that means user u visits POI v at time t.

Definition 4. **(User Profile)** For each user u, we create a user profile D_u, which is a set of check-in activities associated with u. The dataset D used in our model consists of user profiles, i.e., $D = \{D_u : u \in U\}$.

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

Given a dataset D as the union of 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 check-in activity dataset D and a querying user u_q with his/her current location l_q and time t_q (that is, the query is $q = (u_q,\ t_q,\ l_q)$), our goal is to recommend a list of 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 the target user's current location and his/her home location (that is, $|l_q - l_u|$) is greater than d. Otherwise, the problem is a* ***home-town recommendation****.*

Following related studies [11, 10], we set $d = 100km$ in our work, since a distance around $100km$ is the typical radius of human "reach" − it takes about 1 to 2 hours to drive such a distance.

2.2 Model Description

To model user check-in behaviors in LBSNs, we propose a joint probabilistic generative model JIM to take into account various factors including geographical influences, content effect, temporal effect and word-of-mouth effect. Figure 1 shows the graphical representation of JIM. We first introduce the notations of our model and list them in Table 2. Figure 1 shows the graphical representation of JIM where N, K and R denote the number of users, topics and regions, respectively. Our input data, i.e., users' check-in records, are modeled as observed random variables, shown as shaded circles in the figure. The topic and region indexes of check-in records are considered as latent random variables, which are denoted as z and r. Specifically, JIM consists of two components: Time-aware User Interest Component and Popularity-aware User Mobility Component, which will be described in the following.

Time-Aware User Interest Modeling. Intuitively, a user chooses a POI at a given time by matching his/her personal interests with the contents of that POI. Inspired by the early work about user interest modeling [21, 31, 14], JIM adopts latent semantic topics to characterize users' interests to overcome the data sparsity. Specifically, we infer individual user's interest distribution over a set of topics according to the contents (e.g., tags and categories) of his/her checked-in POIs, denoted as $\boldsymbol{\theta_u}$ which is a user-dependent multinomial distribution.

Unfortunately, based on the recent analysis of LBSNs data in [27], about 30% of all POIs are lacking any meaningful textual information. To address this problem, we exploit the association between the contents of checked-in POIs and the check-in time by mining the co-occurrence patterns of user activity contents and the activity time, since the check-in time provides important clues about the content of POIs. Intuitively, the POIs which are checked-in at the same/similar time by most users are more likely to have same/similar functions and categories. So the introduction of check-in time is helpful to infer the topics of POIs, especially whose contents are not available. Technically, each topic z in our JIM model is not only associated with a multinomial distribution over words $\boldsymbol{\phi_z}$, but also with a continuous distribution over time $\boldsymbol{\psi_z}$. This design enables $\boldsymbol{\phi_z}$ and $\boldsymbol{\psi_z}$ to be mutually influenced and enhanced during the topic discovery process by associating them. Another benefit of this design is that the inferred personal interests are not only semantic-aware, but also time-aware.

To integrate the check-in time information to the topic discovery process, the method of time division is essential. Instead of adopting existing discretization methods to split the time into hourly-based slots [12, 32], we treat time t as continuous real value and normalize the time during a day to a range from 0 to 1, considering that 1) time is intrinsically continuous; 2) discretization of time always begs the question of selecting the proper time interval, and the interval is invariable too small for some periods and too large for others. Following the literature [26], we employ the Beta distribution to describe the time distribution of topic z which can behave versatile shape. Double-bounded distributions are appropriate because the training data are bounded in time (i.e., a day).

In the standard topic models [4, 25], a document (i.e., a bag of words) contains a mixture of topics, represented by a topic distribution, and each word has a hidden topic label. While this is a reasonable assumption for long documents, for short document W_v, it is most likely to be about a single topic since most of POIs belong to a single category. We therefore assign a single topic with the document W_v. Similar idea of assigning a single topic to a twitter post has been used before [35].

Popularity-Aware User Mobility Modeling. Different from user online behaviors in the virtual world, users' activities in the physical world are limited by travel distance. So, it is also important to capture users' spatial mobility patterns according to the location distributions of their checked-in POIs. In this component, we cluster the geographical locations into R regions. The spatial clustering phenomenon indicates that users are most likely to check-in a number of POIs and these POIs are usually limited to some geo-

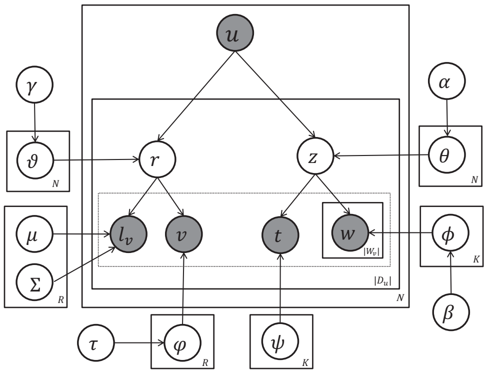


Figure 1: The Graphical Representation of JIM

graphical regions [29]. To model a user's propensity for a POI, we first need to choose the region at which the checked-in POI is located, instead of directly generating the geographical coordinate of the POI. We apply a multinomial distribution $\boldsymbol{\vartheta_u}$ to model user u's mobility over the R latent regions. Following the literatures [20, 19], we assume a Gaussian distribution for each region r, and the location for POI v is characterized by $l_v \sim \mathcal{N}(\boldsymbol{\mu_r}, \boldsymbol{\Sigma_r})$, as follows:

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

where $\boldsymbol{\mu_r}$ and $\boldsymbol{\Sigma_r}$ denote the mean vector and covariance matrix. Since the user trajectories derived from user check-in data are low-sampling-rate [9], and most detailed moving information are lost, the explicit location of a user before visiting v is unknown. Thus, we use the region r to represent the user's activity area at that time.

Popularity can also affect the users' check-in behaviors to a great extent, especially the out-of-town users. When people travel out of town, especially in a new region, their decision-makings are significantly affected by the word-of-mouth opinions, which can be represented as the popularity of the POIs. We use the multinomial distribution $\boldsymbol{\varphi_r}$ to model the normalized popularity of POIs in a region level, i.e., $\varphi_{r,v}$ denotes the normalized popularity of POI v in region r. On the one hand, the local popular POIs represent the local attractions, to some extent. On the other hand, the POIs with high popularity score potentially provide better user experiences. That is why two POIs with the same semantic words can be rated differently in the same region.

Note that, to avoid overfitting, we place a Dirichlet prior [4, 25] over the multinomial distribution $\boldsymbol{\theta_u}$, as parameterized by α:

$$P(\boldsymbol{\theta_u}|\alpha) = \frac{\Gamma(\sum_z \alpha)}{\prod_z \Gamma(\alpha)} \prod_z \theta_{u,z}^{\alpha-1}, \quad (2)$$

where $\Gamma(\cdot)$ is the gamma function. Similarly, priors over $\boldsymbol{\vartheta_u}$, $\boldsymbol{\phi_z}$ and $\boldsymbol{\varphi_r}$ are imposed with parameters γ, β, and τ, respectively. We formally describe the probabilistic generative process of the JIM model in Algorithm 1. Finally, we obtain the joint distribution of the observed and hidden variables as in Equation 3.

2.3 Model Inference

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables $\boldsymbol{v}, \boldsymbol{l_v}, \boldsymbol{W_v}$ and $\boldsymbol{t}$. However, the exact marginalization is intractable due to the coupling

Algorithm 1: Probabilistic generative process in JIM

```
for each topic z do
    Draw φ_z ~ Dirichlet(·|β);
end
for each region r do
    Draw φ_r ~ Dirichlet(·|τ);
end
for each user u do
    Draw θ_u ~ Dirichlet(·|α);
    Draw ϑ_u ~ Dirichlet(·|γ);
end
for each D_u in D do
    for each check-in (u, v, l_v, W_v, t) ∈ D_u do
        Draw a topic index z ~ Multi(θ_u);
        Draw a time t ~ Beta(ψ_{z,1}, ψ_{z,2});
        for each token w ∈ W_v do
            Draw w ~ Multi(φ_z);
        end
        Draw a region index r ~ Multi(ϑ_u);
        Draw a POI index v ~ Multi(φ_r);
        Draw a location l_v ~ N(μ_r, Σ_r);
    end
end
```

between hidden variables. Therefore, we follow the studies [31, 24] to use Markov Chain Monte Carlo method (MCMC) to maximize the complete data likelihood in Equation 3. Note that we adopt conjugate prior (Dirichlet) for multinomial distributions, and thus we can easily integrate out $\boldsymbol{\vartheta}$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$. Due to the space limitation, we omit the derivation details. In this way we facilitate the sampling – that is we need not sample $\boldsymbol{\vartheta}$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ at all. For simplicity and speed, we estimate these Beta distribution parameters $\boldsymbol{\psi_z}$ and Gaussian distribution parameters $(\boldsymbol{\mu_r}, \boldsymbol{\Sigma_r})$ by the method of moments after per iteration of Gibbs sampling. As for the hyperparameters α, β, γ and τ, for simplicity, we take a fix value, i.e., $\alpha = 50/K$, $\gamma = 50/R$ and $\beta = \tau = 0.01$, following the studies [31, 24]. Our algorithm is easily extended to allow these hyperparameters to be sampled and inferred, but this extension can slow down the convergence of the Markov chain.

In the Gibbs sampling procedure, we need to obtain the posterior probability of sampling latent topic z and latent region r for each user check-in record $(u,\ v,\ l_v,\ W_v,\ t)$. First, we need to compute the conditional probability $P(z|\boldsymbol{z}_{\neg u,v}, \boldsymbol{r}, \boldsymbol{v}, \boldsymbol{l_v}, \boldsymbol{W_v}, \boldsymbol{t}, u, \cdot)$, where $\boldsymbol{z}_{\neg u,v}$ represents topic assignments for all check-in records except the current one. We begin with the joint probability distribution of the latent and observed variables shown in Equation 3, and using the Bayes chain rule, we can obtain the conditional probability conveniently as:

$$\begin{aligned} P(z|\boldsymbol{z}_{\neg u,v}, \boldsymbol{r}, \boldsymbol{v}, \boldsymbol{l_v}, \boldsymbol{W_v}, \boldsymbol{t}, u, \cdot) &\propto \frac{n_{u,z}^{\neg u,v} + \alpha}{\sum_{z'}(n_{u,z'}^{\neg u,v} + \alpha)} \\ \times \frac{(1-t)^{\psi_{z,1}-1} t^{\psi_{z,2}-1}}{B(\psi_{z,1}, \psi_{z,2})} &\prod_{w \in W_v} \frac{n_{z,w}^{\neg u,v} + \beta}{\sum_{w'}(n_{z,w'}^{\neg u,v} + \beta)} \end{aligned} \quad (4)$$

where $n_{u,z}$ is the number of times that latent topic z has been sampled from the interest distribution of user u, and $n_{z,w}$ is the number of times that word w is generated from topic z; the number $n^{\neg u,v}$ with superscript $\neg u,v$ denotes a quantity excluding the current instance. It is worth mentioning that our JIM can handle POIs without any content information. In that case, we will remove the last part related with W_v in the above sampling formula.

Then, we sample region r according to the following posterior

$$\begin{aligned} &P(\boldsymbol{v}, \boldsymbol{l_v}, \boldsymbol{W_v}, \boldsymbol{t}, \boldsymbol{z}, \boldsymbol{r}|\alpha, \beta, \gamma, \tau, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ =&P(\boldsymbol{z}|\alpha)P(\boldsymbol{r}|\gamma)P(\boldsymbol{W_v}|\boldsymbol{z}, \beta)P(\boldsymbol{v}|\boldsymbol{r}, \tau)P(\boldsymbol{t}|\boldsymbol{z}, \boldsymbol{\psi})P(\boldsymbol{l_v}|\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ =&\int P(\boldsymbol{z}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\alpha)d\boldsymbol{\theta} \int P(\boldsymbol{r}|\boldsymbol{\vartheta})P(\boldsymbol{\vartheta}|\gamma)d\boldsymbol{\vartheta} \int P(\boldsymbol{W_v}|\boldsymbol{z}, \boldsymbol{\phi})P(\boldsymbol{\phi}|\beta)d\boldsymbol{\phi} \int P(\boldsymbol{v}|\boldsymbol{r}, \boldsymbol{\varphi})P(\boldsymbol{\varphi}|\tau)d\boldsymbol{\varphi} P(\boldsymbol{t}|\boldsymbol{z}, \boldsymbol{\psi})P(\boldsymbol{l_v}|\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned} \quad (3)$$

probability:

$$\begin{aligned}&P(r|\boldsymbol{r}_{\neg u,v}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{l_v}, \boldsymbol{W_v}, \boldsymbol{t}, u, \cdot)\\ \propto &\frac{n_{u,r}^{\neg u,v}+\gamma}{\sum_{r'}(n_{u,r'}^{\neg u,v}+\gamma)}\frac{n_{r,v}^{\neg u,v}+\tau}{\sum_{v'}(n_{r,v'}^{\neg u,v}+\tau)}P(l_v|\boldsymbol{\mu_r}, \boldsymbol{\Sigma_r})\end{aligned} \tag{5}$$

where $n_{u,r}$ is the number of times that region r has been sampled from the spatial activity distribution of user u, and $n_{r,v}$ is the number of times that POI v is generated by region r.

After each iteration, we employ the method of moments to update the Beta and Gaussian distribution parameters (i.e., $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\psi}$) according to the assigned latent variables $\boldsymbol{r}$ and $\boldsymbol{z}$ for simplicity and speed. Specifically, parameters $\boldsymbol{\mu_r}$ and $\boldsymbol{\Sigma_r}$ are updated as in Equations (6) and (7).

$$\boldsymbol{\mu_r} = E(r) = \frac{1}{|S_r|}\sum_{v\in S_r} l_v \tag{6}$$

$$\boldsymbol{\Sigma_r} = D(r) = \frac{1}{|S_r|-1}\sum_{v\in S_r}(l_v-\boldsymbol{\mu_r})(l_v-\boldsymbol{\mu_r})^T \tag{7}$$

where S_r denotes the set of POIs assigned with latent region r. The Beta distribution parameters $\boldsymbol{\psi}$ are updated as follows:

$$\psi_{z,1} = \bar{t}_z(\frac{\bar{t}_z(1-\bar{t}_z)}{s_z^2}-1) \tag{8}$$

$$\psi_{z,2} = (1-\bar{t}_z)(\frac{\bar{t}_z(1-\bar{t}_z)}{s_z^2}-1) \tag{9}$$

where $\bar{t}_z$ and s_z^2 indicate the sample mean and the sample variance of the timestamps assigned with topic z, respectively.

Inference Framework. After a sufficient number of sampling iterations, the approximated posteriors can be used to estimate parameters by examining the counts of $\boldsymbol{z}$ and $\boldsymbol{r}$ assignments to check-in records. The detailed inference framework is shown in Algorithm 2. We first initialize the latent geographical regions by a K-means algorithm (Lines 3-4), and then randomly initialize the topic assignments for the check-in records (Lines 5-9). Afterwards, in each iteration, Equations (4, 5) are utilized to update the region and topic assignments for each check-in record $(u,\ v,\ l_v,\ W_v,\ t)$ (Lines 12-17). After each iteration, we update the Gaussian distribution and Beta distribution parameters (Lines 18-19). The iteration is repeated until convergence (Lines 11-24). In addition, a burn-in process is introduced in the first several hundreds of iterations to remove unreliable sampling results (Lines 20-23). We also introduce the sample lag (i.e., the interval between samples after burn-in) to sample only periodically thereafter to avoid correlations between samples.

Time Complexity. We analyze the time complexity of the above inference framework as follows. 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 $\mathcal{O}(K)$ operations to compute the posterior distribution for sampling latent topic, and then needs $\mathcal{O}(R)$ operations to compute the posterior distribution for sampling latent region. Thus, the whole time complexity is $\mathcal{O}(I(K+R)\sum_u |D_u|)$. To speed up the model training, we parallelize the Gibbs sampling procedure based on the GraphLab framework [1], which is scalable to large-scale check-in datasets.

Algorithm 2: Inference Framework of JIM Model

Input: user check-in collection D, number of iteration I, number of burnin I_b, sample lag I_s, Priors $\alpha, \gamma, \beta, \tau$
Output: estimated parameters $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{\vartheta}}$, $\hat{\boldsymbol{\phi}}$, $\hat{\boldsymbol{\varphi}}$, $\hat{\boldsymbol{\psi}}$, $\hat{\boldsymbol{\mu}}$, and $\hat{\boldsymbol{\Sigma}}$

1 Create temporary variables $\boldsymbol{\theta^{sum}}$, $\boldsymbol{\vartheta^{sum}}$, $\boldsymbol{\phi^{sum}}$, $\boldsymbol{\varphi^{sum}}$, $\boldsymbol{\psi^{sum}}$, $\boldsymbol{\mu^{sum}}$ and $\boldsymbol{\Sigma^{sum}}$, and initialize them with zero;
2 Create temporary variables $\boldsymbol{\psi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$;
3 Initialize the clustering of geographical locations using K-Means method.
4 Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ according to Equations (6,7), respectively;
5 **for** *each* $D_u \in D$ **do**
6   **for** *each check-in record* $(u,\ v,\ l_v,\ W_v,\ t) \in D_u$ **do**
7   Assign topic randomly;
8   **end**
9 **end**
10 Initialize variable *count* with zero;
11 **for** $iteration = 1$ *to* I **do**
12   **for** *each* $D_u \in D$ **do**
13   **for** *each check-in record* $(u,\ v,\ l_v,\ W_v,\ t) \in D_u$ **do**
14   Update topic assignment using Equation (4);
15   Update region assignment using Equation (5);
16   **end**
17   **end**
18   Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ according to Equations (6, 7), respectively;
19   Update $\boldsymbol{\psi}$ according to Equations (8, 9);
20   **if** *(iteration* $> I_b$*) and (iteration mod* $I_s == 0$*)* **then**
21     $count = count + 1$;
22   Update $\boldsymbol{\vartheta^{sum}}$, $\boldsymbol{\theta^{sum}}$, $\boldsymbol{\phi^{sum}}$, $\boldsymbol{\varphi^{sum}}$, $\boldsymbol{\psi^{sum}}$, $\boldsymbol{\mu^{sum}}$ and $\boldsymbol{\Sigma^{sum}}$ as follows:

$$\vartheta_{u,r}^{sum} += \frac{n_{u,r}+\gamma}{\sum_{r'}(n_{u,r'}+\gamma)} \tag{10}$$

$$\theta_{u,z}^{sum} += \frac{n_{u,z}+\alpha}{\sum_{z'}(n_{u,z'}+\alpha)} \tag{11}$$

$$\phi_{z,w}^{sum} += \frac{n_{z,w}+\beta}{\sum_{w'}(n_{z,w'}+\beta)} \tag{12}$$

$$\varphi_{r,v}^{sum} += \frac{n_{r,v}+\tau}{\sum_{v'}(n_{r,v'}+\tau)} \tag{13}$$

$$\boldsymbol{\psi_z^{sum}} += \boldsymbol{\psi_z} \tag{14}$$

$$\boldsymbol{\mu_r^{sum}} += \boldsymbol{\mu_r} \tag{15}$$

$$\boldsymbol{\Sigma_r^{sum}} += \boldsymbol{\Sigma_r} \tag{16}$$

23   **end**
24 **end**
25 **Return** model parameters $\hat{\boldsymbol{\theta}} = \frac{\boldsymbol{\theta^{sum}}}{count}$, $\hat{\boldsymbol{\vartheta}} = \frac{\boldsymbol{\vartheta^{sum}}}{count}$, $\hat{\boldsymbol{\phi}} = \frac{\boldsymbol{\phi^{sum}}}{count}$, $\hat{\boldsymbol{\varphi}} = \frac{\boldsymbol{\varphi^{sum}}}{count}$, $\hat{\boldsymbol{\psi}} = \frac{\boldsymbol{\psi^{sum}}}{count}$, $\hat{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu^{sum}}}{count}$, and $\hat{\boldsymbol{\Sigma}} = \frac{\boldsymbol{\Sigma^{sum}}}{count}$;

3. POI RECOMMENDATION USING JIM

Once we have learnt the model parameter set $\hat{\boldsymbol{\Psi}} = \{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\varphi}}, \hat{\boldsymbol{\psi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\}$, given a querying user u_q with the querying time t_q and location l_q, i.e., $q = (u_q, t_q, l_q)$, we compute a probability of user u_q checking-in each POI v, and then select top-k POIs with highest probabilities for the target user. Specifically, the probability of user u_q checking-in POI v, given his/her associated time t_q and location l_q as well as the learnt model parameters $\hat{\boldsymbol{\Psi}}$, is computed according to Equation 17.

$$\begin{aligned}P(v, l_v, W_v|u_q, t_q, l_q, \hat{\boldsymbol{\Psi}}) &= \frac{P(v, l_v, W_v, t_q|u_q, l_q, \hat{\boldsymbol{\Psi}})}{\sum_{v'} P(v', l_{v'}, W_{v'}, t_q|u_q, l_q, \hat{\boldsymbol{\Psi}})}\\ &\propto P(v, l_v, W_v, t_q|u_q, l_q, \hat{\boldsymbol{\Psi}})\end{aligned} \tag{17}$$

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

$$\begin{aligned}&P(v, l_v, W_v, t_q|u_q, l_q, \hat{\boldsymbol{\Psi}})\\ =&\sum_r P(r|l_q, \hat{\boldsymbol{\Psi}})P(v, l_v, W_v, t_q|u_q, r, \hat{\boldsymbol{\Psi}})\end{aligned} \tag{18}$$

where $P(r|l_q, \hat{\boldsymbol{\Psi}})$ denotes the probability of user u lying in region r given his/her current location l_q, and it is computed as in Equation 19 according to Bayes rule, in which the prior probability of latent region r can be estimated using Equation 20, as follows.

$$P(r|l_q, \hat{\boldsymbol{\Psi}}) = \frac{P(r)P(l_q|r, \hat{\boldsymbol{\Psi}})}{\sum_{r'} P(r')P(l_q|r', \hat{\boldsymbol{\Psi}})} \propto P(r)P(l_q|r, \hat{\boldsymbol{\Psi}}) \tag{19}$$

$$P(r) = \sum_u P(r|u)P(u) = \sum_u \frac{N_u+\kappa}{\sum_{u'}(N_{u'}+\kappa)}\hat{\vartheta}_{u',r} \tag{20}$$

[1] http://www.select.cs.cmu.edu/code/graphlab/

where N_u denotes the number of check-ins generated by user u. In order to avoid overfilling, we introduce the Dirichlet prior parameter κ to play the role of pseudocount. $P(v, l_v, W_v, t_q|u_q, r, \hat{\mathbf{\Psi}})$ is defined as in Equation 21 where we adopt geometric mean for the probability of topic z generating word set W_v, i.e., $P(W_v|z, \hat{\mathbf{\Psi}}) = \prod_{w \in W_v} P(w|z, \hat{\mathbf{\Psi}})$, considering that the number of words associated with different POIs may be different.

Based on Equations (18-21), the original Equation 17 can be reformulated as in Equation 22.

3.1 Efficient Online Recommendation

Our proposed recommendation strategy is to first train JIM offline and obtain a knowledge model containing necessary insights about users' interests, mobility patterns and POIs' properties, and then use the knowledge model to retrieve top-k POIs for the real-time query q. To speed up the process of online recommendation, we propose a ranking framework based on Equation 22, as follows:

$$S(q, v) = \sum_{a=(r,z)} W(q, a) F(v, a) \tag{23}$$

$$W(q, a) = \hat{\theta}_{u_q,z} \hat{\psi}_{z,t_q} P(l_q|\hat{\boldsymbol{\mu}}_{\boldsymbol{r}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{r}}) \tag{24}$$

$$F(v, a) = P(r) P(l_v|\hat{\boldsymbol{\mu}}_{\boldsymbol{r}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{r}}) \hat{\varphi}_{r,v} \left(\prod_{w \in W_v} \hat{\phi}_{z,w} \right)^{\frac{1}{|W_v|}} \tag{25}$$

where $S(q, v)$ represents the ranking score of POI v for query q. Each region-topic pair (r, z) can be seen as an attribute (i.e., $a = (r, z)$), 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. This ranking framework separates the offline scoring computation from the online scoring 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., $\hat{\psi}_{z,t_q}$, $\hat{\theta}_{u_q,z}$ and $(\hat{\boldsymbol{\mu}}_{\boldsymbol{r}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{r}})$) are also computed offline, the online computation is just a simple combination process. This design enables maximum precomputation 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 $K \times R$ attributes by a simple weighted sum function.

The straightforward method of generating the top-k POIs needs to compute the ranking scores for all POIs according to Equation (23) and select top-k ones with highest ranking scores, which is, however, computationally inefficient, especially when the number of POIs or the number of POI attributes becomes large. To improve the online recommendation efficiency, the proposed JIM model can be seamlessly integrated with the TA-based query processing technique developed in [31], because $W(q, a)$ is non-negative, and thus the proposed ranking function in Equation 23 is a monotonic linear weighting function given a query q. The technology has the nice property of correctly finding top-k results by examining the minimum number of POIs without scanning all ones, which enables the JIM model scalable to large-scale datasets.

4. EXPERIMENTS

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

4.1 Experimental Settings

4.1.1 Datasets

Our experiments are conducted on two real datasets: Foursquare and Twitter. Their basic statistics are shown in Table 3.

	Foursquare	Twitter
# of users	4,163	114,508
# of POIs	21,142	62,462
# of check-ins	483,813	1,434,668
time span	Dec 2009-Jul 2013	Sep 2010-Jan 2011

Table 3: Basic statistics of Foursquare and Twitter datasets

Foursquare. This dataset contains the check-in history of 4,163 users who live in the California, USA. For each user, it contains his/her social networks, check-in POI IDs, location of each check-in POI in terms of latitude and longitude, check-in time and the contents of each check-in POI. Each check-in is stored as *user-ID, POI-ID, POI-location, POI-content, check-in time*. Each record in social networks is stored as *user-ID, friend-ID* and the total number of social relationship is 32,512. This dataset is publicly available [2].

Twitter. This dataset is based on the publicly available twitter dataset [8]. Twitter supports third-party location sharing services like Foursquare and Gowalla (where users of these services opt-in to share their check-ins on Twitter). But the original dataset does not contain the category and tag information about each POI. So, we crawled the content information associated with each POI from Foursquare with the help of its publicly available API [3]. Each check-in record has the same format with the above Foursquare dataset. But, this dataset does not contain user social network.

4.1.2 Comparative Approaches

We first compare our JIM model with the following four competitor methods which represent state-of-the-art POI recommendation techniques.

SVDFeature. SVDFeature [5] is a machine learning toolkit designed to solve the feature-based matrix factorization. This toolkit is very powerful and Chen et. al [5] adopted it to win KDD Cups for two consecutive years (2011 and 2012). Based on this toolkit, we build a factorization model incorporating more side information beyond the user-POI matrix, including POI content, POI geographical location, temporal dynamics (i.e., check-in time) and popularity of POI, to fairly compare with our model JIM. The limitation of SVDFeature is that it cannot deal with continuous time and location, thus we adopt the discretization methods developed in [32, 30] to segment them into bins and grid squares, respectively.

Time-Aware Collaborative Filtering (TACF). TACF is the state-of-the-art time-aware POI recommendation method [32], which is

[2]http://www.public.asu.edu/ hgao16/dataset.html
[3]https://developer.foursquare.com/

$$P(v, l_v, W_v, t_q|u_q, r, \hat{\mathbf{\Psi}}) = P(l_v|r, \hat{\mathbf{\Psi}}) P(v|r, \hat{\mathbf{\Psi}}) \sum_z P(z|u_q, \hat{\mathbf{\Psi}}) P(t_q|z, \hat{\mathbf{\Psi}}) \left(\prod_{w \in W_v} P(w|z, \hat{\mathbf{\Psi}}) \right)^{\frac{1}{|W_v|}} \tag{21}$$

$$\begin{aligned} P(v, l_v, W_v|u_q, t_q, l_q, \hat{\mathbf{\Psi}}) &\propto \sum_r \left[P(r) P(l_q|r, \hat{\mathbf{\Psi}}) P(l_v|r, \hat{\mathbf{\Psi}}) P(v|r, \hat{\mathbf{\Psi}}) \sum_z P(z|u_q, \hat{\mathbf{\Psi}}) P(t_q|z, \hat{\mathbf{\Psi}}) \left(\prod_{w \in W_v} P(w|z, \hat{\mathbf{\Psi}}) \right)^{\frac{1}{|W_v|}} \right] \\ &= \sum_r \sum_z \left[\hat{\theta}_{u_q,z} P(l_q|\hat{\boldsymbol{\mu}}_{\boldsymbol{r}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{r}}) \hat{\psi}_{z,t_q} P(r) P(l_v|\hat{\boldsymbol{\mu}}_{\boldsymbol{r}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{r}}) \hat{\varphi}_{r,v} \left(\prod_{w \in W_v} \hat{\phi}_{z,w} \right)^{\frac{1}{|W_v|}} \right] \end{aligned} \tag{22}$$

a collaborative filtering model integrating *temporal effect*. Specifically, TACF splits time into hour-based slots and models the temporal preference of a given user in a time slot according to his/her visited POIs in that time slot. Given a querying user u at a specific time t, TACF first finds a group of users sharing similar temporal preferences with u.

UPS-CF. UPS-CF, proposed in [11], is a collaborative recommendation framework which is especially designed for out-of-town users. This framework integrates user-based collaborative filtering and social-based collaborative filtering, i.e., to recommend POIs to a target user according to the check-in records of both his/her friends and similar users with him/her.

LCA-LDA. LCA-LDA is a location-content-aware recommender model which is developed to support POI recommendation for users traveling in new cities [31]. This model takes into account both personal interests and local preferences of each city by exploiting both POI co-visiting patterns and contents of POIs. Compared with JIM, it ignores the temporal effect and geographical influence.

To further validate the benefits brought by exploiting the temporal effect, content effect, word-of-mouth effect and geographical influence, respectively, we design four baselines. **JIM-S1** is the first simplified version of the JIM model where we remove the check-in time information, and the latent variable z only generates the word set W_v for each check-in record. **JIM-S2** is the second simplified version of the JIM model where we remove the content information of POIs, and the latent variable z only generates the check-in time. As the third simplified version of JIM, **JIM-S3** means our model without considering the word-of-mouth effect. So, the latent variable r only generates the location of POI v. As the last simplified version of JIM, **JIM-S4** means our model without considering the geographical influence. So, the latent variable r only generates the ID of POI v, and the popularity-aware user mobility component degenerates into a model-based collaborative filtering method.

4.1.3 Evaluation methods

Since our JIM is designed for 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 in terms of a collection of user activities, we divide the user's activities into a training set and a test set. For the scenario of home-town recommendation, we randomly select 20% of the activity records occurring at the user's home town as test set, and use the remaining activity records as the training set. Similarly, for the scenario of out-of-town recommendation, we randomly select 20% of the activity records generated by the user when he/she travels out of town as the test set, and use the remaining activity records as training set. To decide whether an activity record occurs at home town or out of town, we measure the distance between the user's home location and the POI (i.e., $|l_u - l_v|$). If the distance is greater than the threshold d, then we assume the activity occurs when the user is out-of-town. Besides, to simulate a more real POI recommendation scenario, we have to choose a location coordinate as the target user's current standing position before visiting v. Specifically, for each test case $(u,\ v,\ l_v,\ W_v, t)$, we use a Gaussian function to generate a coordinate l within the circle of radius d centered at l_v, to represent the current standing point of user u. Thus, a query $q = (u,\ l,\ t)$ is formed for the test case.

According to the above dividing strategies, we split the user activity dataset D into the training set D_{train} and the test set D_{test}. To evaluate the recommendation methods, we adopt the evaluation methodology and measurement Accuracy@k proposed in [11, 31, 22]. Specifically, for each activity record $(u,\ v,\ l_v,\ W_v, t)$ in D_{test} as well as its associated query q: First, we compute the ranking score for POI v and all other POIs which are within the circle of radius d centered at l_v and unvisited by u previously. Second, 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, $p = 1$). Third, 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.

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:

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

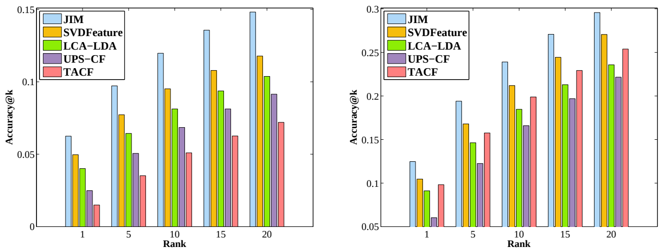
where $\#hit@k$ denotes the number of hits in the test set, and $|D_{test}|$ is the number of all test cases. The results have been validated by means of a standard 5-fold cross validation.

4.2 Recommendation Effectiveness

In this part, we present the overall performance of the recommendation methods with well-tuned parameters. Figure 2 reports the performance of the recommendation methods on the Foursquare dataset. From the figure, we observe that the accuracy values gradually become higher with the increase of k. We show only 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.

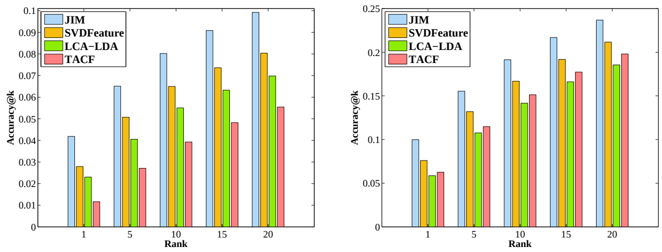
It is apparent that the models have significant performance disparity in terms of top-k accuracy, especially in the out-of-town recommendation scenario. Figure 2(a) presents the recommendation accuracy in the scenario of out-of-town recommendation where the accuracy of JIM is about 0.12 when $k = 10$, and 0.15 when $k = 20$ (i.e., the model has a probability of 12% of placing an appealing POI in the top-10 and 15% of placing it in the top-20). Clearly, our proposed JIM model outperforms other competitor models significantly, and the advantages of JIM over other competitor methods are very obvious in this scenario. Several observations are made from the results: 1) TACF and UPS-CF drop behind JIM, SVDFeature and LCA-LDA, showing the advantages of latent class models incorporating the contents of checked-in POIs. This is because users have few check-in activity records in out-of-town regions (e.g., cities), and TACF and UPS-CF suffer from the severe data sparsity in this scenario while JIM, SVDFeature and LCA-LDA are latent class models and integrate content information of POIs, which can alleviate the data sparsity problem to a great extent. 2) UPS-CF performs better than TACF, showing the benefits brought by exploiting social influence. The check-in records left by social friends can help alleviate the issue of data sparsity in the out-of-town scenario to a small extent, since we sometimes travel out of town to visit our friends. 3) JIM and SVDFeature perform better than LCA-LDA. This is because LCA-LDA ignores both temporal effect and geographical influence compared with JIM and SVDFeature. 4) JIM achieves much higher recommendation accuracy than SVDFeature although they use the same types of features and information, showing the advantage of probabilistic generative models over feature-based matrix factorization. This may be because some important information is lost when discretizing time and geographical location information in SVDFeature.

In Figure 2(b), we report the performance of all recommendation models for the home-town scenario. From the results, we observe that the recommendation accuracies of all methods are higher in Figure 2(b) than that in Figure 2(a). Besides, LCA-LDA outperforms TACF in Figure 2(a) while TACF slightly exceeds LCA-LDA in Figure 2(b) due to its incorporation of temporal effect, showing



(a) Out-of-town Recommendation (b) Home-town Recommendation

Figure 2: Top-k Performance on Foursquare Dataset



(a) Out-of-town Recommendation (b) Home-town Recommendation

Figure 3: Top-k Performance on Twitter Dataset

that time-aware collaborative filtering method better suits the setting where the user-POI matrix is not sparse, and the model-based method which integrates content information of POIs is more capable of overcoming the difficulty of data sparsity in the out-of-town scenario. Another observation is that the performance gap between our JIM model and other competitor methods is smaller than that in Figure 2(a), showing that the performance differences among recommendation methods become less obvious when the issue of data sparsity is not serious. The comparison between Figure 2(a) and Figure 2(b) also reveals that the two recommendation scenarios are intrinsically different, and should be separately evaluated.

Figure 3 reports the performance of the recommendation models on the Twitter dataset. We do not compare our model with UPS-CF since this dataset does not contain user social network information. From the figure, we can see that the trend of comparison result is similar to that presented in Figure 2, and the main difference is that all recommendation methods achieve lower accuracy. This may be because users in the Foursquare dataset have more check-in records on average than users in the Twitter dataset, which enables the models to capture users' interests more accurately.

4.3 Impact of Different Factors

Methods	Out-of-Town Scenario			Home-Town Scenario		
	Ac@1	Ac@10	Ac@20	Ac@1	Ac@10	Ac@20
JIM-S1	0.052	0.108	0.134	0.101	0.204	0.253
JIM-S2	0.045	0.096	0.119	0.117	0.226	0.280
JIM-S3	0.049	0.101	0.125	0.112	0.219	0.271
JIM-S4	0.056	0.113	0.140	0.106	0.212	0.262
JIM	**0.062**	**0.121**	**0.149**	**0.124**	**0.241**	**0.298**

Table 3: Recommendation Accuracy on Foursquare Dataset.

Methods	Out-of-Town Scenario			Home-Town Scenario		
	Ac@1	Ac@10	Ac@20	Ac@1	Ac@10	Ac@20
JIM-S1	0.033	0.072	0.089	0.084	0.162	0.199
JIM-S2	0.028	0.064	0.079	0.095	0.179	0.220
JIM-S3	0.031	0.067	0.083	0.091	0.173	0.213
JIM-S4	0.037	0.075	0.093	0.088	0.168	0.206
JIM	**0.041**	**0.080**	**0.099**	**0.100**	**0.191**	**0.235**

Table 4: Recommendation Accuracy on Twitter Dataset.

To explore the benefits of integrating the temporal effect, content effect, word-of-mouth effect and geographical influence into JIM model, respectively, we compare our JIM model with four baselines, JIM-S1, JIM-S2, JIM-S3 and JIM-S4. The comparison results are shown in Tables 3 and 4. From the results, we first observe that JIM consistently outperforms the four baselines in both out-of-town recommendation scenario and home-town recommendation scenario, indicating that JIM benefits from simultaneously considering the four factors influencing users' decision-making in a joint way. Second, we observe that the contribution of each factor to improve recommendation accuracy is different. Besides, another observation is that the contributions of the same factor are different in the two different recommendation scenarios. Specifically, according to the importance of the four factors in the out-of-town recommendation scenario, they can be ranked as follows: Content Effect $>$ Word-of-Mouth Effect $>$ Temporal Effect $>$ Geographical Influence, while in the home-town recommendation scenario they can be ranked as: Temporal Effect $>$ Geographical Influence $>$ Word-of-Mouth Effect $>$ Content Effect. Obviously, the content information plays a dominant role in overcoming the issue of data sparsity of out-of-town recommendation scenario, while the temporal effect is most important to improve home-town recommendation. 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 in the out-of-town scenario does not matter as much as that in home town; and 3) users' daily routines may change when they travel out of town.

4.4 Impact of Model Parameters

Tuning model parameters, such as the number of topics (i.e., K), the number of regions (i.e., R), is critical to the performance of JIM model. We therefore study the impact of model parameters on Foursquare dataset in this section.

As for the hyperparameters α, γ, β and τ, following recent works [24, 31], we empirically set fixed values (i.e., $\alpha = 50/K$, $\gamma = 50/R$, $\beta = \tau = 0.01$). We tried different setups and found that the performance of JIM model is not sensitive to these hyperparameters, but the performance of JIM is sensitive to the number of topics and regions. Thus, we tested the performance of JIM model by varying the number of topics and regions, and present the results in Tables 5 and 6. From the results, we observe that the recommendation accuracy of JIM first increases with the increasing number of topics, and then it does not change significantly when the number of topics is larger than 80. Similar observation is made for increasing the number of regions (i.e., R): the recommendation accuracy of JIM increases with the increasing number of regions, and then it does not change much when the number of regions is larger than 120. 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 complex enough to handle the data. At this point, it is less helpful to improve the model performance by increasing K and R. It should be noted that the performance reported in Figure 2 is achieved with 100 latent topics (i.e., $K = 100$) and 150 latent regions (i.e., $R = 150$). Similar observations are also made on the Twitter dataset, and the experimental results presented in Figure 3 are obtained with the optimal parameter settings $K = 100$ and $R = 240$ since the check-in records in the Twitter dataset are more widespread in the geographical space than the check-ins contained in the Foursquare dataset.

R \ K	K=20	K=40	K=60	K=80	K=100	K=120
R=30	0.089	0.095	0.099	0.101	0.101	0.101
R=60	0.095	0.102	0.106	0.108	0.108	0.109
R=90	0.101	0.108	0.113	0.115	0.116	0.116
R=120	0.106	0.113	0.118	0.120	0.120	0.121
R=150	0.106	0.113	0.118	0.120	0.121	0.121
R=180	0.106	0.113	0.118	0.121	0.121	0.121

Table 5: Out-of-Town Recommendation Accuracy@10.

R \ K	K=20	K=40	K=60	K=80	K=100	K=120
R=30	0.178	0.190	0.198	0.202	0.203	0.203
R=60	0.191	0.204	0.213	0.217	0.217	0.217
R=90	0.204	0.217	0.227	0.231	0.232	0.232
R=120	0.212	0.227	0.236	0.241	0.241	0.241
R=150	0.212	0.227	0.236	0.241	0.241	0.242
R=180	0.212	0.227	0.237	0.241	0.241	0.242

Table 6: Home-Town Recommendation Accuracy@10.

4.5 Recommendation Efficiency

This experiment is to evaluate the online recommendation efficiency. For the online recommendation of JIM, we adopt two methods. The first one is called JIM-TA which adopts the TA-based query processing technique [31] to produce online recommendation. The second is called JIM-BF which uses a naive brute-force algorithm to produce top-k recommendations, i.e., to compute a ranking score for all POIs, and then choose k ones with largest ranking scores. As the feature-based matrix factorization model SVDFeature can be seamlessly integrated with the Metric Tree-based retrieval algorithm proposed in [15], we compare JIM-TA with it (SVDFeature-MT). Note that the ranking function in SVDFeature is not monotonic and, hence, the TA-based query processing technique cannot apply to it. All the online recommendation algorithms were implemented in Java (JDK 1.6) and run on a Linux Server with 32G RAM.

Table 7 presents the average online efficiency of three different methods over all queries created for D_{test}. We show the performance where k is set to 1, 5, 10, 15 and 20. A greater value of k is not necessary for the top-k recommendation task. For example, on average JIM-TA finds the top-10 recommendations from about 62,000 POIs in 10.4 ms. From the results, we observe that 1) both JIM-TA and SVDFeature-MT outperform JIM-BF significantly, justifying the benefits brought by pruning POI search space to avoid scan all ones; 2) JIM-TA is more efficient than SVDFeature-MT, this is because JIM-TA makes full use of the monotonic of the ranking function in Equation 23 and achieves a tighter upper bound for effective pruning, which results in the decrease in the number of POIs to be examined, while the ranking function in SVDFeature does not have the nice property of monotonicity; and 3) the time costs of both JIM-TA and SVDFeature-MT increase with the increasing number of recommendations (i.e., k), but they are still much lower than that of JIM-BF in the recommendation task.

Methods	Online Recommendation Time Cost (ms)				
	k=1	k=5	k=10	k=15	k=20
JIM-TA	3.21	5.45	9.67	17.24	33.36
JIM-BF	145.34	145.34	145.34	145.34	145.34
SVDFeature-MT	12.02	28.46	36.78	54.06	82.17

Table 7: Recommendation Efficiency on Twitter Dataset.

5. RELATED WORK

POI recommendation, also called location or place recommendation, has been considered as an essential task in the domain of recommender systems. Bao et al. [2] provided a good survey on POI recommendation. It was firstly investigated and studied on trajectory data. Due to the lack of mapping relationship between geographical coordinates and specific real-world POIs, a POI is usually defined as the stay points extracted from users' trajectory logs [37, 36]. Recently, with the development of location-based social networks, it is easy for users to check-in at POIs in the physical world, resulting in easy access of large-scale check-in records. Based on the LBSNs data, many recent work has tried to improve POI recommendation by exploiting and integrating geographical and social influence, temporal effect and content information of POIs.

Geo-Social Influence. Many recent studies [8, 11, 29, 10, 33] 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. [29] delved into POI recommendation by investigating the geographical influences among locations and proposed a system that combines user preferences, social influence and geographical influence. Cheng et al. [6] investigated the geographical influence through combining a multi-center Gaussian model, matrix factorization and social influence together for location recommendation. Lian et al. [18] incorporated spatial clustering phenomenon resulted by geographical influence into a weighted matrix factorization framework to deal with the challenge from matrix sparsity. However, all of them do not consider the current location of the user. Thus, no matter whether the user is located in the home town or traveling out of town, they will recommend the same POIs to the user. In light of this, Ference et al. [11] designed a collaborative recommendation framework which not only investigates the roles of friends and similar users in POI recommendation, but also considers the current location of the user.

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 chronological patterns on LBSNs. Gao et al. [12] investigated the temporal cyclic patterns of user check-ins in terms of temporal non-uniformness and temporal consecutiveness. Yuan et al. [32] also incorporated the temporal cyclic information into a user-based collaborative filtering framework for time-aware POI recommendation. Cheng et al. [7] introduced the task of successive personalized POI recommendation in LBSNs by embedding the temporal chronological patterns.

Content Information. Most recently, researchers explored the content information of POIs to alleviate the problem of data sparsity. Hu et al. [14] proposed a spatial topic model for POI recommendation considering both spatial aspect and textual aspect of user posts from Twitter. Liu et al. [21] studied the effect of POI-associated tags for POI recommendation with an aggregated LDA and matrix factorization method. Yin et al. [31] exploited both personal interests and local preferences based on the contents associated with spatial items. Gao et al. [13] and Zhao et al. [34] studied both POI-associated contents and user sentiment information (e.g., user comments) into POI recommendation and reported their good performance. However, all of them do not consider the time information associated with the contents of POIs.

As described above, while there are many studies to improve the POI recommendation by exploiting geographical-social influence, temporal effect and content information, they lack an integrated analysis of their joint effect to alleviate the issue of data sparsity, especially in out-of-town recommendation scenario. Most of the above work assumed that users are in home town, thus they ignored that users' interests can vary drastically at different regions (e.g., cities), failing to deal with user interest drift. Our proposed method strategically takes all these factors into consideration and presents a flexible probabilistic generative model for both home-town recommendation and out-of-town recommendation. To deal with the drift of user interests, our method considers the current location of the user and exploits the local word-of-mouth effect. Moreover, we are the first to study the importance of each mentioned factor to overcome the data sparsity in both home-town and out-of-town recommendation scenarios under a unified framework.

Although some recent literatures [3, 22] used classification-based method to predict the next place a user will move by extracting multiple features from users' movement history, their problem definition is different from ours. They assumed that the querying user is currently located at a POI, and exploited sequential pattern information to predict the next POI.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a joint probabilistic generative model JIM to model users' check-in behaviors in LBSNs, which strategically integrates the factors of temporal effect, content effect, geographical influence and word-of-mouth effect in a unified probabilistic framework to effectively overcomes the issues of data sparsity and user interest drift, especially when users travel out of town. To demonstrate the applicability and flexibility of JIM, we investigated how it supports two recommendation scenarios in a unified way, *home-town recommendation* and *out-of-town recommendation*. We conducted extensive experiments to evaluate the performance of our JIM model in terms of both effectiveness and efficiency. The results showed superiority of JIM model over other competitor methods. Besides, we studied the importance of each factor in improving both home-town and out-of-town recommendation under the same framework, and found that the *content information* plays a dominant role in overcoming the data sparsity in out-of-town recommendation scenario, while the *temporal effect* is most important to improve home-town recommendation.

As a promising research direction, we would like to explore enhancements to our model by integrating the social influence.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734 – 749, june 2005.

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

[3] R. Baraglia, C. I. Muntean, F. M. Nardini, and F. Silvestri. Learnext: Learning to predict tourists movements. In *CIKM*, pages 751–756, 2013.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[5] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. Svdfeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.*, 13(1):3619–3622, Dec. 2012.

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

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

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

[9] M.-F. Chiang, Y.-H. Lin, W.-C. Peng, and P. S. Yu. Inferring distant-time location in low-sampling-rate trajectories. In *KDD*, pages 1454–1457, 2013.

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

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

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

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

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

[15] N. Koenigstein, P. Ram, and Y. Shavitt. Efficient retrieval of recommendations in a matrix factorization framework. In *CIKM*, pages 535–544, 2012.

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

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

[18] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*, pages 831–840, 2014.

[19] M. Lichman and P. Smyth. Modeling human location data with mixtures of kernel densities. In *KDD*, pages 35–44, 2014.

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

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

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

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

[24] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *KDD*, pages 1285–1293, 2012.

[25] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In *NIPS*, pages 1973–1981, 2009.

[26] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *KDD*, 2006.

[27] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz. On the semantic annotation of places in location-based social networks. In *KDD*, pages 520–528, 2011.

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

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

[30] H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang. Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data*, 9(3):19:1–19:41, Apr. 2015.

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

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

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

[34] K. Zhao, G. Cong, Q. Yuan, and K. Zhu. Sar: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *ICDE*, 2015.

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

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

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