Time-aware Point-of-interest Recommendation

Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, Nadia Magnenat-Thalmann
School of Computer Engineering, Nanyang Technological University, Singapore 639798
{qyuan1@e., gaocong@, zma4@e., axsun@, nadiathalmann@}ntu.edu.sg

ABSTRACT

The availability of user check-in data in large volume from the rapid growing location-based social networks (LBSNs) enables many important location-aware services to users. Point-of-interest (POI) recommendation is one of such services, which is to recommend places where users have not visited before. Several techniques have been recently proposed for the recommendation service. However, no existing work has considered the temporal information for POI recommendations in LBSNs. We believe that time plays an important role in POI recommendations because most users tend to visit different places at different time in a day, *e.g.,* visiting a restaurant at noon and visiting a bar at night. In this paper, we define a new problem, namely, the *time-aware POI recommendation*, to recommend POIs for a given user at a specified time in a day. To solve the problem, we develop a collaborative recommendation model that is able to incorporate temporal information. Moreover, based on the observation that users tend to visit nearby POIs, we further enhance the recommendation model by considering geographical information. Our experimental results on two real-world datasets show that the proposed approach outperforms the state-of-the-art POI recommendation methods substantially.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Filtering

Keywords

Recommendation; Point-of-interest; Location-based Social Networks; Spatio-Temporal

1. INTRODUCTION

Location-based social networks (LBSNs) such as Foursquare, Gowalla, Facebook Places, *etc.*, have been growing rapidly in recent years. As of January 2013, Foursquare had over 3 billion check-ins made by 30 million users[1]. In these LBSNs, users can post their physical locations in the form of "check-in" (see Figure 1 for an example), and share their experiences and tips for points of interest (POI), such as restaurants, sightseeing sites, *etc.* The availability of user check-in data in large volume makes it possible to recommend unvisited POIs to users. *POI recommendation* is of great value to both users and the business owners of POIs. Several techniques have been exploited for POI recommendations. Since *user-based collaborative filtering* method has performed well for POI recommendations according to a recent work [23], in this paper, we aim to further enhance the user-based collaborative filtering method by considering the important factors that have been neglected in earlier studies.

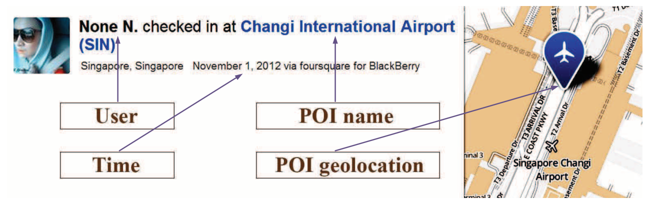


Figure 1: An example of check-in

Users' activities are often influenced by time. For example, a user is more likely to go to a restaurant rather than a bar for lunch at noon, and is more likely to go to a bar rather than a library at midnight. Therefore, the recommendation results should be time-aware or time-oriented. However, to the best of our knowledge, *no existing work has considered the time factor* for POI recommendations in LBSNs. In this paper, we define a new problem for POI recommendations, namely, the *time-aware POI recommendation*, which aims to return a set of POIs for a user to visit at a specified time in a day. We believe that this is a natural and useful extension to the conventional POI recommendation problem which has not considered the time factor.

To address the POI recommendation problem, we explore the following two human behaviors, namely, *temporal behavior* and *spatial behavior* for POI recommendations. Note that these two human behaviors are not applicable to most of the recommendation tasks over conventional purchase or rating data (*e.g.,* for movies, books or other products).

1. **Temporal behavior**: Temporal influence plays an important role in analyzing users' daily activities [16]. In LBSNs, the temporal behavior of a user is reflected by the check-ins of the user to POIs over time. We can therefore mine the temporal behaviors of users by analyzing their historical check-ins, and make use of the mined temporal behaviors for time-aware POI recommendations. For example, if many users visit a bar at midnight but very few users visit a library at the same time, then the bar should be given a higher priority

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

than the library when we recommend POIs for a user to visit at midnight.

2. **Spatial behavior**: Spatial influence is another important factor in analyzing users' daily activities [13, 23]. Intuitively, users tend to visit nearby POIs, and thus the POIs visited by users often form spatial clusters. Hence, the spatial behaviors of users can be utilized to enhance POI recommendations.

To exploit the temporal behavior in POI recommendations, we propose an approach incorporating temporal influence. Intuitively, if two users have similar temporal behavior, they are likely to visit similar POIs at the same time. Thus, we perform collaborative filtering by exploiting other user's temporal preferences to POIs. Since most people maintain a relatively fixed routine of their daily mobility (*e.g.,* go to office in the morning and have dinner at restaurants after work) [5, 13], we split time into hourly-based slots and model the temporal preference to POIs of a user in a time slot by the POIs visited by the user in the time slot. However, splitting time into slots will make the check-in data sparser. To tackle this problem, we enhance the proposed method with *smoothing* by taking advantage of user's temporal preference at other time slots. Given a user and a specific time, we first find the users sharing similar temporal preference with him or her, and then produce the time-specific recommendations based on their historical check-ins made around the time.

We also propose a new method to exploit the spatial behavior. We assume the willingness of a user moving from one POI to another POI is influenced by the distance between the two POIs. Based on the assumption, we calculate the conditional probability that a user will visit another POI given a POI. By applying the Bayes rule, we derive that the probability that a user will visit a candidate POI is determined by both the popularity of the candidate POI and the conditional probability of visiting the historical check-in POIs of the user given the candidate one. We further extend the model to accommodate the temporal factor. Note that spatial behavior has been exploited in [23] with a different assumption from ours. A detailed discussion on the differences between our method and that in [23] is given in Section 2.

Finally, we employ a unified framework to combine the aforementioned two approaches, *i.e.*, integrating the spatial and temporal influences, to make the time-aware POI recommendation. The contributions of this paper are threefold:

- We define a new time-aware POI recommendation problem, which aims at recommending time-specific POIs for a user.
- We analyze the temporal influence and spatial influence from historical check-in data, and develop POI recommendation methods that exploit the two kinds of influences. Moreover, we fuse the spatial and temporal influences with a framework to make the time-aware POI recommendation.
- We evaluate the proposed POI recommendation method by comprehensive experiments on two real world LBSN datasets collected from Foursquare and Gowalla, respectively. Experimental results show that: (1) Time has a significant influence on the accuracy of POI recommendations, improving on the recommendation accuracy by 37% to 51% over the baseline collaborative filtering method without considering time; (2) The proposed method of utilizing the spatial influence outperforms the state-of-the-art POI recommendation approach [23] that also considers the spatial influence; and (3) The proposed unified method substantially outperforms the baseline methods in terms of recommendation accuracy.

The rest of this paper is organized as follows. In Section 2, we review related work. In Sections 3 and 4, we describe the proposed recommendation methods incorporating temporal influence and spatial influence, respectively. Section 5 presents the unified framework. We present experimental results in Section 6. Finally Section 7 concludes this paper.

2. RELATED WORK

Collaborative Filtering. Collaborative filtering (CF) technique is widely adopted for recommender systems and many collaborative filtering recommendation methods (*e.g.*, [8, 12, 17]) have been proposed. The CF methods can be divided into two categories, namely *memory-based CF* and *model-based CF*. Memory-based CF methods can be further divided into user-based CF and item-based CF. User-based CF first finds similar users based on their ratings on items using a similarity measure, such as cosine similarity. Then the recommendation score for an item is computed by a weighted combination of historical ratings on the item from similar users. We will detail the user-based CF for POI recommendations in Section 3. In contrast, item-based CF works by finding items that are similar to other items the user has liked or rated. Model-based CF builds models using data mining techniques, such as clustering, on user ratings and the models are used to make recommendations [18]. The model building algorithms are usually computationally expensive.

POI Recommendation. The work by Ye *et al.* [23] is the most closely related to our work. Ye *et al.* tailor the CF model for POI recommendations, aiming at improving the recommendation accuracy. The work considers the social influence under the framework of user-based CF, and models the spatial influence by model-based method (a Bayesian CF algorithm). Specifically, the work employs user-based CF to compute the recommendation score of a candidate POI for a user. To exploit the social influence, the work makes use of the user's friends for recommendation rather than all the users by following the approach in [15]. To explore the geographical influence, this work assumes that the probability that a user visits two POIs is determined by their distance, and that the probability of a user visiting a set of POIs is the product of probabilities of visiting all the pairwise POIs in the set. The probability that a user checks in a new POI is estimated by the product of the probabilities of visiting all the pairwise POIs, each pair consisting of the new POI and each previously visited POI. This work reports three findings: (i) Geographical influence has a significant impact on the accuracy of POI recommendations; (ii) The social friend links themselves contribute little for the accuracy of POI recommendations; and (iii) Random walk based method and item-based CF perform much worse than user-based CF for POI recommendations.

In light of these findings, we develop our method based on user-based CF. We also investigate the geographical influence for POI recommendations and propose a new method to exploit the geographical influence. We make a different assumption from that in [23]. Specifically, we assume that the willingness that a user moves from a POI to another POI is a function of their distance. Based on the Bayes rule, we develop a method with a sound probabilistic foundation to recommend POIs. Because of the different assumptions made in [23] and our work, the techniques are significantly different. An additional advantage of our method is that the temporal information can be easily incorporated into our probabilistic model, which can further improve the recommendation accuracy.

In their earlier work [22], Ye *et al.* aim to improve recommendation efficiency for POI recommendations in LBSNs by exploiting the social and geospatial information in recommendation. Nevertheless, the work focuses on efficiency instead of effectiveness.

There exists other work that incorporates social link information into POI recommendations, such as the probabilistic generative model-based method [21], and matrix factorization-based method [4]. The focus of [4, 21] is to explore social link information for POI recommendations and their problem setting is different from ours.

POI Prediction. The recent work on POI prediction in LBSNs [5] is also related, in which any POI, irrespective of whether it has been visited by a user, can be suggested to the user as the next-to-visit POI. The authors report that the movement of human is periodic between "work" and "home" states, and the Gaussian mixture model is employed to predict the POI to be visited next. Specifically, given a user and the previously visited POIs of the user, the method [5] first calculates the probability of the user being in either of the two states in the next time point, and then uses the Gaussian models of the two states to estimate the visiting probability for each POI. Although the method is developed for POI prediction, it can be adapted for POI recommendations by disregarding the POIs that have been visited by the user in the prediction results. We implemented the method [5] and applied it for time-aware POI recommendations, but it performed much worse than user-based CF method. One reason is that the method is not developed for recommendation, and another potential reason could be: the method requires that each user have a number of check-ins (more than 10 used in [5]) on each day to estimate the parameters of Gaussian models. However, the datasets used in this paper are much sparser in which less than 15% of users have at least 10 check-ins per day (see Section 6.1 for more details about the data).

Clements *et al.* propose a kernel convolution method to predict similar locations for a given location based on users' travel behavior [6]. The problem settings of this work is different from the above studies that focus on location prediction for users.

Location Identification and Recommendation. There exists work on location identification and recommendation from the GPS trajectories of a number of users over a long period. Zheng *et al.* [24] mine interesting locations and travel sequences, which can be recommended to users as they are popular among many users. Cao *et al.* [3] develop a technique capable of extracting semantic locations from GPS data. The outcome of the extraction can serve as location recommendations that takes the significance of location and the distance from the user to the location into account. Leung *et al.* [14] perform co-clustering on a graph consisting of users, locations and activity entities to find similar users, locations and activities from the GPS trajectories, and make location recommendation based on the cluster results. Our work differs from these studies in that our work aims to provide personalized time-aware recommendation results for a specific user while the results in these studies are not personalized.

Our work is also related to the work on location recommendation in location-based services. Horozov *et al.* [9] propose a user-based CF method to recommend restaurants for a user. The method first finds candidate POIs close to the user, and then makes recommendation based on the similarities between users who have rated these POIs and the user, and their ratings to these POIs.

Recommendation with Temporal Information. Ye *et al.* [20] present a tagging approach to recommending semantic tags for POIs in LBSNs, and the tagging method considers the temporal information as features. However, both the problem and the approach in the work [20] are different from ours. Existing work on POI recommendations does not take into account the time factor. In other domains time factor is considered for recommendation in several studies, including using item-based CF method [7], matrix factorization method [11] and random walk based method [19]. However, the time factor in these studies is different from the periodic time factor considered in this paper. In these studies, the time gap between the occurring time of a previous rating and the recommendation time is used as a decaying factor to weight the rating—earlier rating will be given a smaller weight. In contrast, we divide time into periodic time slots (*e.g.*, by hour) and make use of the periodic temporal property in our method.

Table 1: Symbols

Symbol	Description
U, L, T	user set, POI set, time slot set
u, v, l, t	user $u, v \in U$, POI $l \in L$, time slot $t \in T$
$c_u, c_{u,t}$	the binary check-in vector of u over L, and the binary check-in vector u over L at t
$c_{u,l}, c_{u,t,l}$	element of c_u and $c_{u,t}$, respectively
$w_{u,v}$	the similarity between u and v
$w^{(t)}_{u,v}, w^{(ts)}_{u,v}$	time-enhanced similarity, smoothed similarity
$dis(l_i, l_j)$	distance between l_i and l_j
$wi(dis)$	the willingness a user visits a dis far away POI
$CI_l, CI_{l,t}$	the set of check-ins at l, CI_l at time t

Context-aware Recommendation. Our work is related to the existing work [1] on integrating context into recommender systems. The approach proposes a reduction-based algorithm that uses only the ratings that pertain to the recommendation context. The approach suffers from rating sparsity. The general motivation of using only context-pertinent data is similar to that of our method to be presented in Section 3.2. However, we address the sparsity problem in our proposed method in Section 3.3.

Summary. Our study differentiates itself from all these existing studies in that the study of both temporal influence and spatial influence under a unified framework for time-aware POI recommendations in LBSNs is unexplored in previous work.

3. UTILIZING TEMPORAL INFLUENCE

In this section, we first present the baseline user-based CF method in Section 3.1. We then present the method of incorporating time influence in the user-based CF in Section 3.2. The enhancement to the method by smoothing is then presented in Section 3.3. All the notations used in this paper are listed in Table 1.

3.1 User-based Collaborative Filtering

Given a user, user-based CF first calculates the similarities between the user and other users, and then produces a prediction for a POI by taking a weighted combination of the other users' check-in records on the POI. More specifically, let $v \in U$ denote a user in the user set U, and $l \in L$ denote a POI in the POI set L. We set $c_{v,l} = 1$, if v has checked in (or visited) l before; and $c_{v,l} = 0$ otherwise. Given a user u, the recommendation score that u will check-in a POI l that she has not visited before is computed by the following equation, where $w_{u,v}$ is the similarity between user u and user v.

$$\widehat{c_{u,l}} = \frac{\sum_v w_{u,v} c_{v,l}}{\sum_v w_{u,v}}$$

The similarity between two users $w_{u,v}$ can be computed with various measures. Among these measures, cosine similarity is a widely adopted measure for implicit data (check-in records are implicit data and $c_{u,l}$ takes either 1 or 0 value). The cosine similarity between u and v is defined in Equation 1, where each user is represented by a binary check-in vector over all POIs L.

$$w_{u,v} = \frac{\sum_l c_{u,l} c_{v,l}}{\sqrt{\sum_l c_{u,l}^2} \sqrt{\sum_l c_{v,l}^2}} \tag{1}$$

3.2 Incorporating Temporal Influence

3.2.1 Check-in Representation

As reported in [5], human show strong periodic behavior throughout a day. For example, people tend to check-in restaurants during daily lunch hours.

We split a day into multiple equal time slots based on hour (We can also split the time based on the day of a week). For ease of presentation, *time* and *time slot* are used interchangeably in this paper unless noted otherwise. Then, we represent the behavior of a user at a specific time by the set of check-ins that user has made at that time.

To represent the temporal check-in behavior of users, we introduce the the time dimension into the conventional user-POI matrix. Specifically, we use user-time-POI cube (UTP) to represent the temporal check-in records. In the UTP cube, each element $c_{u,t,l}$ represents the check-in activity of a user u, at a POI l at time slot t, where $c_{u,t,l} = 1$ if user u has checked in POI l at time t, and $c_{u,t,l} = 0$ otherwise.

3.2.2 Recommendation

To make use of the time influence for time-aware POI recommendations, we extend the user-based CF model in two aspects: (i) we leverage the time factor when computing the similarity between two users (to be presented in Section 3.2.3); (ii) we consider the historical check-ins at time t in the repository, rather than at all time, during recommendation.

Given a user u and time t, the recommendation score that the user will check-in an unvisited POI l at t is:

$$\widehat{c}_{u,t,l}^{(t)} = \frac{\sum_v w_{u,v}^{(t)} c_{v,t,l}}{\sum_v w_{u,v}^{(t)}},$$

where $w_{u,v}^{(t)}$ is the temporal behavior similarity between u and v. Next, we detail the computation of the temporal behavior similarity $w_{u,v}^{(t)}$.

3.2.3 Similarity Estimation

In our method, the similarity between two users is estimated based on their temporal behaviors over all time. Specifically, if two users always check-in the same POIs at the same time, the similarity value between the two will be high, and one user's check-in history has a large impact on the POI recommendation for the other user. Therefore, we extend the cosine similarity measure to calculate the similarity between u and v as follows:

$$w_{u,v}^{(t)} = \frac{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{u,t,l} \cdot c_{v,t,l}}{\sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{u,t,l}^2} \sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{v,t,l}^2}} \tag{2}$$

Note that, the similarity might be calculated in an alternative manner based on the check-in history at the *exact time*, which is similar to the general idea of context-aware recommendation [1]. We did not adopt this approach because of two reasons: (i) There is a high possibility that the user has never made any check-in at the time. In our two datasets (Foursquare check-ins in Singapore, and Gowalla check-ins in California and Nevada, see Section 6.1 for more details), on average the check-ins of each user fail into 9.97 and 13.58 hour slots, respectively. Thus, if we only consider the time, no similar users may be found for the user; (ii) Even if the user has check-ins at the time, the number of these check-ins is still too small for meaningful similarity computation (on average, only 4.17 and 5.81 check-ins in each hour for the two datasets, respectively).

3.3 Enhancement by Smoothing

The straightforward extension of incorporating temporal influence presented in Section 3.2 has its weakness in handling data sparsity. In this section, we first explain the reason of the weakness and then present two enhancements to overcome this weakness. First, we smooth the similarity estimation using the similarity values of the other time slots. Second, we consider the POIs visited by similar users at different time slots, and weight them with the estimated similarity for recommendation.

The similarity incorporating temporal influence in Equation 2 is estimated based on the UTP cube, which is much sparser than the user-POI matrix. The sparsity could easily make it fail to characterize the users' similarity when a POI is visited at different time slots by different users. For example, consider two POIs and two time slots. User u checked in l_1 and l_2 at t_1 and t_2, respectively, while user v checked in l_1 and l_2 at t_2 and t_1, respectively. If we do not consider time, the similarity between the two users will be 1 by Equation 1, since both users checked in both POIs. However, if the time is taken into account, their similarity becomes 0 according to Equation 2, since their check-ins to the two POIs are made at different time. Obviously, the time-aware similarity is not desirable in the case, particularly when t_2 is very close to t_1, *e.g.,* the next hour to t_1.

We proceed to discuss how to address this problem caused by data sparsity. A straightforward method is to use a decaying parameter to give a higher weight to the POIs checked in at close time slots, and a smaller weight to those checked in at distant time slots. However, this method faces the following challenge: A user's behavior at a time is described by the check-in records of the user at that time, and the user behavior at different time may be similar. For example, the check-in behavior of a user at 9-10AM might be similar to the check-in behavior of the user at 3-4PM, because the user is likely to stay at workplace and make check-ins around it. Such similarity cannot be easily captured by a decaying parameter.

To further illustrate the point, we analyze the check-in data in Foursquare made within Singapore for the similarity patterns between check-in behaviors of users at different time. Note that, the results on the other data used in our experiment arequalitatively similar to this dataset, and thus are omitted here.

Let $c_{u,t} = \{c_{u,t,1}, c_{u,t,2}, ..., c_{u,t,L}\}$ be the check-in vector of user u at time t, which is extracted from the UTP cube. For each user u, we calculate the cosine similarity between every pair of check-in vectors c_{u,t_i} and c_{u,t_j} at time t_i and t_j respectively. Then, we calculate the similarity value between two time slots t_i and t_j to be the average of the similarity values of all users between these two time slots t_i and t_j.

Figure 2 shows the similarity curves for three time slots (6:00, 8:00, and 16:00) over the Singapore data. The similarity curve for 6:00 shows the check-in similarity between 6:00 and every other hour in a day, similarly for the other two curves. Observe that the check-in behavior at a time is similar to the check-in behavior of its close time slots. For example, the check-in similarity between

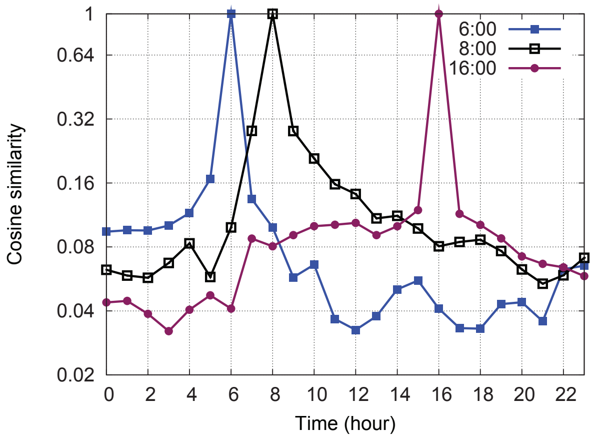


Figure 2: User behavior similarities between a given hour (6:00, 8:00, and 16:00) and other hours

6:00 and 5:00 is much higher than the similarity between 6:00 and 4:00. Nevertheless, although 6:00 and 8:00 are close to each other in terms of time, their *similarity curves* are in quite different shapes. For instance, the check-in behavior of 6:00 is similar to that at its previous hours (0:00 – 5:00), but quite different from the behavior at the later hours (8:00 – 23:00). The curve of time 8:00 displays an opposite shape. In contrast, the curve of 16:00 drops with equal speed on both sides. Observe that the curve of 16:00 almost does not decrease from 8:00 to 14:00, and it even increases around 12:00 to 13:00. This is, users might stay at workplace during that period, and hence have similar check-ins (while people tend to have lunch about 12:00, making the curve drops to some extent). Instead of using cosine similarity to compute the user check-in behavior, we have also tried other metrics, such as Pearson correlation and Total Variation Distance, but observed similar results. In summary, the check-in behavior at one time may be more similar to some time slots than others. This motivates us to develop a method to utilize the check-in behavior similarity for POI recommendations.

We propose to smooth the UTP cube based on the check-in similarity between different time slots. For each check-in vector in the cube, we smooth it using the vectors of those similar time slots. Specifically, we compute a new value for element $c_{u,t,l}$ using Equation 3, where $\rho_{t,t'}$ is the similarity between time slots t and t'. Accordingly, the smoothing enhanced similarity between two users u and v is calculated using Equation 4.

$$\widetilde{c}_{u,t,l} = \sum_{t'=1}^{T} \frac{\rho_{t,t'}}{\sum_{t''=1}^{T} \rho_{t,t''}} c_{u,t',l} \tag{3}$$

$$w_{u,v}^{(te)} = \frac{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{u,t,l} \cdot \widetilde{c}_{v,t,l}}{\sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{u,t,l}^{2}} \sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{v,t,l}^{2}}} \tag{4}$$

With the similarity formulation in Equation 4, if two users u and v have visited the same POIs at *the same* or *similar* time slots, their similarity value will be high. Otherwise, if they have visited the same POIs at dissimilar time slots, their similarity value will be low.

The check-in behavior similarity between two time slots also enables us to introduce another enhancement in recommending POIs for user u at a given time t. Recall that in the method described in Section 3.2, only the POIs visited by a similar user v at the time t are considered in recommendation. The check-in behavior similarity between time slots makes it possible to consider any candidate POI l visited by a similar user v, irrespective if the check-in time t' by v is the same as the time t. More specifically, in the enhancement method, a candidate POI l is weighted by the similarity value between two time slots, t' and t, if the historical check-ins are in different from the time t. The recommendation score, denoted by $\widehat{c}_{u,t,l}^{(te)}$, that user u will check-in l at t is then updated as follows:

$$\widehat{c}_{u,t,l}^{(te)} = \frac{\sum_{v} w_{u,v}^{(te)} \sum_{t'} \widetilde{c}_{v,t',l} \cdot \rho_{t,t'}}{\sum_{v} w_{u,v}^{(te)}} \tag{5}$$

4. UTILIZING SPATIAL INFLUENCE

The geolocation of POI is an important factor affecting human's check-in behavior and has been exploited in earlier studies [23]. In this work, we however make a different assumption from the work in [23] and show that POI recommendation can be improved by considering spatial influence in an alterative manner. In our experiments, we shall compare the performance of different approaches of incorporating spatial influence.

We first analyze the effect of distance on user's check-in behavior, and then propose a new POI recommendation method by exploiting the spatial influence. In our work, we assume that *human tend to visit nearby POIs to their previous locations, and their willingness to visit a POI decreases as the distance increases*. To verify this assumption, we perform a data analysis on two datasets (see Section 6.1 for more details on the datasets) to study the impact of distance on users' check-in behaviors.

For each user, we sort the user's check-ins by time. For the check-ins made within a day, we calculate the distance between two POIs of every two adjacent check-ins. We aggregate the results of all users, and plot the number of check-ins as a function of the distance in Figure 3. Note that a larger probability value implies that users are more willing to check in POIs at that distance. Observe from Figure 3, the distribution of the probability values follows a power law, suggesting that users are more willing to check in nearby POIs to their current places. Moreover, observe that the slop of the curve of Singapore (Figure 3(a)) drops quickly after 10KM. Similar observation holds for the curve of California and Nevada (Figure 3(b)). That is, the willingness of people visiting a faraway POI (*e.g.,* 10KM) drops significantly.

4.1 Incorporating Spatial Influence

To incorporate spatial influence in POI recommendations, we use a power law distribution to model the willingness of a user moving from one place to another as a function of their distance. Defined in Equation 6, $wi(dis)$ is the willingness of a user to visit a dis km far away POI, and a, k are parameters of the power law function.

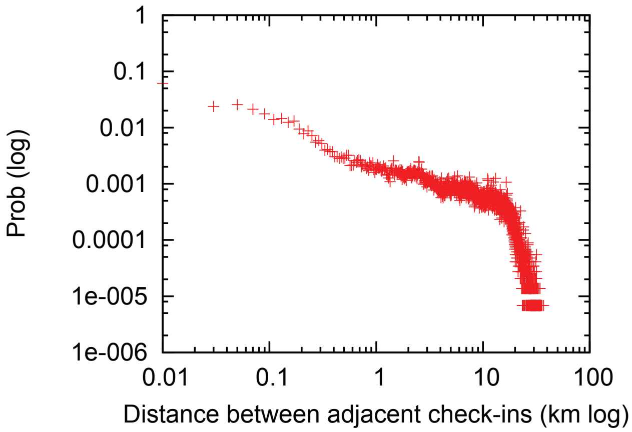
$$wi(dis) = a \cdot dis^{k} \tag{6}$$

Maximum likelihood estimation [2] is used to estimate the two parameters a and k. More specifically, we take logarithmic on both side of Equation 6, and get the following equation.

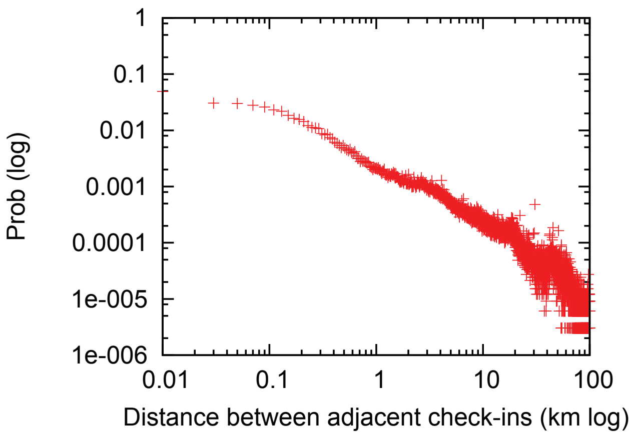
$$\ln(wi(dis)) = \ln(a) + k\ln(dis). \tag{7}$$

The above linear function over $\ln(dis)$ can be easily learned by the least-square regression. As the result, we learn the parameters in Equation 6. Note that, in learning the two parameters, irregular portion in Figures 3(a) and 3(b) (*i.e.,* data points having distance larger than 10km) is not considered. These data points represent fewer than 15% of the total number of check-ins.

Consider a user is currently at POI l_i and POI l_j is a candidate POI to check in next, at distance $dis(l_i, l_j)$ from l_i. We model the



(a) Singapore



(b) California and Nevada

Figure 3: Distribution of distance between successive check-ins

probability that the user will check in l_j to be proportional to the user's willingness to check in a POI at distance $dis(l_i, l_j)$. The conditional probability is computed using the following equation.

$$p(l_j|l_i) = \frac{wi(dis(l_i, l_j))}{\sum_{l_k \in L, l_k \neq l_i} wi(dis(l_i, l_k))} \tag{8}$$

Observe that, as the distance increases, the conditional probability decreases, which reflects that the user is less likely to visit a faraway candidate POI.

Given a user u, and his/her historical POIs L_u, we calculate $P(l|L_u)$ as the ranking score for each candidate POI l, and then recommend the top ranked POIs to the user. Based on the Bayes rule, this score is calculated as follows.

$$\widehat{c}^{(s)}_{u,l} = P(l|L_u) \propto P(l)P(L_u|l) \tag{9}$$

$$= P(l) \prod_{l' \in L_u} P(l'|l) \tag{10}$$

Note that the derivation from Equation 9 to Equation 10 is based on the assumption that for a given l, the check-in probabilities of POIs in L_u are independent from each other.

4.2 Enhancement by Temporal Popularity

In Equations 9 and 10, $P(l)$ is the prior probability that POI l is checked in by all users in the dataset. However, the popularity of a POI varies over time, *e.g.*, a restaurant is more popular around noon and evening, and a workplace is more popular during working hours. To illustrate this point, we plot the check-in probabilities of the top-5 most popular POIs in Figure 4, based on the check-in data of Singapore, collected from Foursquare. The check-in probability at a given time is computed by the ratio of the number of check-ins at that time to the total number of check-ins at the POI. Observe that the popularity of each POI varies greatly over time, and different POIs become popular at different time.

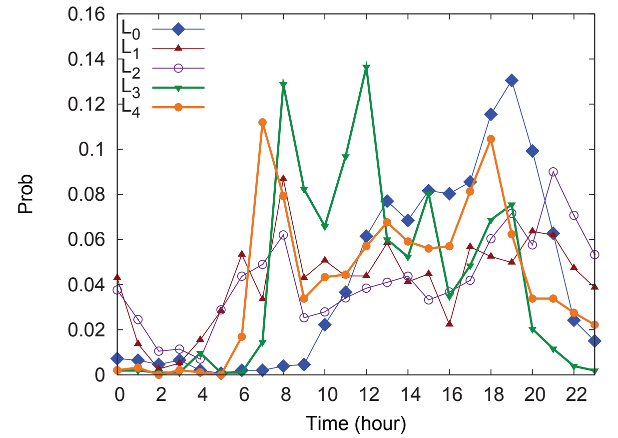


Figure 4: The distribution of 5 popular POIs over time

Reconsider Equations 9 and 10. The probability of checking in a POI should reflect both its popularity at the specific time and the distance to the user's current location. In other words, the probability user u will check in POI l at time t depends on the probability (or popularity) of l at t, along with the distance between l and u's visited POIs. It provides us a way to enhance this model by adjusting $P(l)$ with the temporal popularity of POI l, denoted by $P_t(l)$.

$$P_t(l) = \beta \frac{|CI_{l,t}|}{\sum_{l' \in L} |CI_{l',t}|} + (1-\beta)\frac{|CI_l|}{\sum_{l' \in L} |CI_{l'}|} \tag{11}$$

In the above equation, $|CI_l|$ is the number of check-ins at l; $|CI_{l,t}|$ is the number of check-ins at l at time t, and β tunes the weight between l's temporal popularity and long-term popularity.

By replacing $P(l)$ in Equation 10 with $P_t(l)$, the resultant method incorporates the temporal popularity. The recommendation score, denoted by $\widehat{c}^{(se)}_{u,t,l}$, is then computed by the following equation.

$$\widehat{c}^{(se)}_{u,t,l} \propto P_t(l) \prod_{l' \in L_u} P(l'|l)$$

We emphasize that the temporal information here is used in a different way from what we do in Section 3. In Section 3, the temporal information is employed to discover personalized temporal preference on POIs. In contrast, here we use the temporal preference on POIs in a collective manner (*i.e.,* by all users).

5. A UNIFIED FRAMEWORK

Given a user u, time t and a candidate POI l, we can get a recommendation score $\widehat{c}^{(t)}_{u,t,l}$ that user u will check in l at t using either of the two methods incorporating temporal influence in Section 3 (*i.e.,* with or without smoothing). Similarly, we can also compute a score $\widehat{c}^{(s)}_{u,t,l}$ using either of the two methods incorporating spatial influence in Section 4 (*i.e.,* with or without considering temporal popularity).

Table 2: Statistics on the datasets (after pre-processing)

Dataset	No. of Check-ins	No. of Users	No. of POIs
Foursquare	194,108	2,321	5,596
Gowalla	456,988	10,162	24,250

We use linear interpolation to weight the two scores to compute the final recommendation score for POI l. However, the two scores are measured by different methods, and have different value ranges. Thus, we first normalize the two scores using min-max normalization before we combine them.

$$\bar{c}^{(t)}_{u,t,l} = \frac{\hat{c}^{(t)}_{u,t,l} - min_{l'}(\hat{c}^{(t)}_{u,t,l'})}{max_{l'}(\hat{c}^{(t)}_{u,t,l'}) - min_{l'}(\hat{c}^{(t)}_{u,t,l'})} \quad (12)$$

$$\bar{c}^{(s)}_{u,t,l} = \frac{\hat{c}^{(s)}_{u,t,l} - min_{l'}(\hat{c}^{(s)}_{u,t,l'})}{max_{l'}(\hat{c}^{(s)}_{u,t,l'}) - min_{l'}(\hat{c}^{(s)}_{u,t,l'})} \quad (13)$$

In the above equations, $max_{l'}(\cdot)$ and $min_{l'}(\cdot)$ denote the maximum and minimum check-in scores of u at t across all POIs.

After normalization, we compute the combined score that user u will check-in POI l at time t using the following equation, where α is a tuning parameter.

$$c_{u,t,l} = \alpha \times \bar{c}^{(t)}_{u,t,l} + (1-\alpha) \times \bar{c}^{(s)}_{u,t,l}, \quad (14)$$

By this framework, we calculate the check-in score for each candidate POI, and return the top ranked POIs as the recommendation results.

6. EXPERIMENTS

We evaluate the accuracy of the proposed methods, including the recommendation methods utilizing temporal influence (presented in Section 3), the methods utilizing spatial influence (presented in Section 4), and the unified method (presented in Section 5). We also study the effect of the length of time slot on the recommendation accuracy.

6.1 Experimental Setup

We first introduce the datasets used in our experiments and the metrics. We then list the methods to be evaluated and explain the design of the experiments.

Dataset. Two datasets are used in our experiments. We collect 342,850 check-ins from Foursquare which were made within Singapore between Aug. 2010 and Jul. 2011. We also set a bounding box and extracted 736,148 Gowalla check-ins from the dataset provided by [5], which were made within California and Nevada between Feb. 2009 and Oct. 2010. Each check-in contains user, time and POI ID information. For both datasets, we removed users who have checked in fewer than 5 POIs, and then removed POIs which fewer than 5 users checked in. After pre-processing, the Foursquare dataset (Foursquare) contains 194,108 check-ins made by 2,321 users at 5,596 POIs, and the Gowalla dataset (Gowalla) contains 456,988 check-ins made by 10,162 users at 24,250 POIs (see Table 2). The two datasets have different scales in terms of the size of entities (*i.e.,* users, POIs, and check-ins) and geographical range. For each user, we randomly mark off 12.5% of his or her visited POIs as development data to tune parameters, and mark off another 25% of POIs as testing data to evaluate the effectiveness of the recommendation methods. The two datasets are available online [2].

[2] `http://www.ntu.edu.sg/home/gaocong/datacode.htm`

The densities of the training data of Foursquare and Gowalla are 6.35×10^{-3} and 9.85×10^{-4}, respectively. As expected, after splitting a day into 24 slots by hours, the data becomes much sparser. The densities of the two datasets after splitting become 4.82×10^{-4} and 6.65×10^{-5}, respectively.

Metrics. To study the effectiveness of the proposed methods, *i.e.*, how well the methods can recover the hold-off POIs in the testing data for a given user at a given time, we use two metrics, namely, precision@N and recall@N (denoted by pre@N and rec@N, respectively), where N is the number of recommendation results, following the work [23].

The pre@N measures how many POIs in the top-N recommended POIs correspond to the hold-off POIs in the testing data, and the rec@N measures how many POIs in the hold-off POIs in the testing set are returned as top-N recommended POIs. Given an user u and a time t (a time slot), we denote by $T_{u,t}$ the set of corresponding groundtruth POIs in the testing data, and by $R_{u,t}$ the set of recommended POIs by a method. We divide the POIs in the two sets, and get the following three values: $tp_{u,t}$, $tn_{u,t}$, and $fp_{u,t}$.

$tp_{u,t}$	: the number of POIs contained in both $T_{u,t}$ and $R_{u,t}$
$tn_{u,t}$	: the number of POIs contained in $T_{u,t}$ but not in $R_{u,t}$
$fp_{u,t}$	: the number of POIs contained in $R_{u,t}$ but not in $T_{u,t}$

Then, the precision and recall for time slot t are calculated as follows:

$$precision(t) = \frac{\sum_{u'\in U} tp_{u',t}}{\sum_{u'\in U}(tp_{u',t} + np_{u',t})} \quad (15)$$

$$recall(t) = \frac{\sum_{u'\in U} tp_{u',t}}{\sum_{u'\in U}(tp_{u',t} + tn_{u',t})} \quad (16)$$

The overall precision and recall are calculated by averaging the precision and recall values over all time slots, respectively:

$$precision = \frac{1}{T}\sum_{t'\in T} precision(t') \quad (17)$$

$$recall = \frac{1}{T}\sum_{t'\in T} recall(t') \quad (18)$$

Equations 17 and 18 actually compute the macro-precision and recall over all time slots. We also considered other ways to calculate the overall precision and recall, such as micro-precision and recall. We found that the results from these variant measures are very similar and we choose to report the results calculated by Equations 17 and 18. For each metric, we consider three values of N (*i.e.,* 5, 10, 20) in our experiments, where 5 is the default value.

Recall that both the Foursquare and Gowalla datasets have a low density, which usually results in relatively low precision and recall values [10, 23]. In addition, the POIs in the test data of each user may represent only a small portion of POIs that the user may be interested in. Thus, although the relatively low precision and recall values are common and reasonable, in this paper, *we focus on the relative improvements we achieved, instead of the absolute values*.

Recommendation Methods. We evaluated 9 recommendation methods as listed in Table 3. Among the 9 methods, User-based CF with time function (UTF) is a baseline method that has not been discussed before. We detail it in the following.

UTF is the user-based version of the algorithm proposed in [7]. It estimates the similarity between users as does the conventional user-based CF, but weights the check-ins of a similar user according to the gaps between their time slots and the recommendation time slot. The rationale is that with a larger time gap the check-in would

Table 3: Methods for comparison

Method	Description
U	User-based CF (Section 3.1)
UTF	U with Time Function [7]
UT	U with Temporal preference (Section 3.2.2)
UTE	UT with smoothing Enhancement (Section 3.3)
SB	Spatial influence based Baseline [23] (Section 2)
S	Spatial influence based recommendation (Sec. 4.1)
SE	S with popularity Enhancement (Section 4.2)
U+SB	Combination of U and SB [23]
UTE+SE	Combination of UTE and SE (Section 5)

be less useful for the POI recommendation. Formally, given a user u and time t, the recommendation score that u will check in POI l at time t is:

$$\vec{c}^{\,t}_{u,l} = \frac{\sum_v w_{u,v} f(t_{v,l}, t)}{\sum_v w_{u,v}}$$

where the time function $f(t_{v,l}, t) = e^{-\frac{1}{H}|t-t_{v,l}|}$; $|t - t_{v,l}|$ is the time gap between the time slot v checked in l and the time of recommendation, and H is a parameter controlling the decaying factor of the time gap. We tune H on the development data by varying H from 1 to 5, and found the best recommendation accuracy is always reached when $H = 1$ on both datasets. We also tried the original item-based method [7], but its results were much worse.

We conduct three groups of experiments. The first group is to evaluate the accuracy of the proposed recommendation methods utilizing temporal influence, and we compare methods U, UTF, UT and UTE.

The second group is to evaluate the effectiveness of the proposed methods utilizing spatial influence, and we compare methods SB, S, SE, among which SB is the state-of-the-art method utilizing spatial influence.

The third group is used to evaluate the effectiveness of the proposed unified method, and we compare methods U, U+SB, and UTE+SE. Among them, U is the baseline method, and U+SB is the unified method proposed in [23].

6.2 Performance of Methods

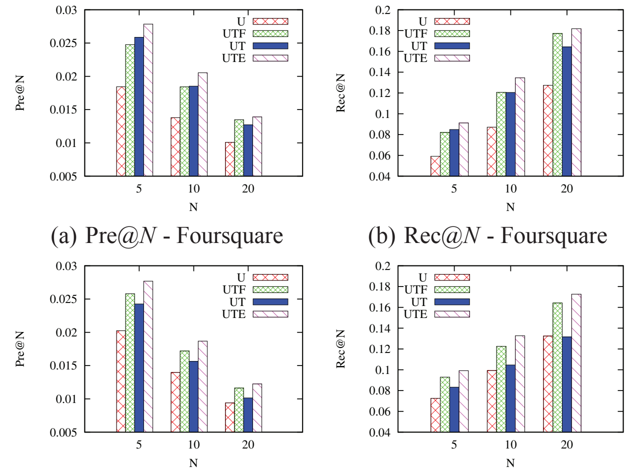
Methods Utilizing Temporal Influence. We compare the effectiveness of the methods utilizing the temporal influence on the two datasets. The precision and recall of the 4 methods (U, UTF, UT, and UTE) are reported in Figure 5.

Observe that the methods that make use of the time information always outperform the baseline method U, which does not take time into account. In terms of pre@5, UT outperforms U by 40% and 20% on Foursquare data and Gowalla data, respectively. UTE outperforms U by 51% on Foursquare data and 37% on Gowalla data, respectively. This result suggests that, taking time into consideration is essential for time-aware POI recommendations.

Compared to U, the UTF baseline method always exhibits better results in terms of all measures. This is because UTF exploits the time information, and recommends POIs which have been visited by users who have similar visiting history with the user.

Compared to UTF, UT performs better on the Foursquare data, but poorer on the Gowalla data. This is because the Gowalla data is much sparser than the Foursquare data. When data is sparse, the similarity between users in UT becomes less accurate because the similarity only considers exact matching on the UTP cube.

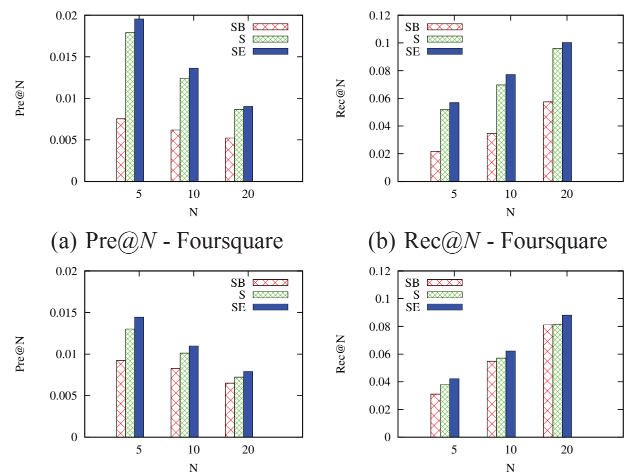
Among all 4 methods, UTE always achieves the best results in terms of both precision and recall at different N values on both datasets. The superior performance is due to the smoothing enhancement, which addresses the sparsity problem. The detailed results of each time slot on the Foursquare data are provided in Section 6.4 as a case study.



(a) Pre@N - Foursquare (b) Rec@N - Foursquare

(c) Pre@N - Gowalla (d) Rec@N - Gowalla

Figure 5: Performance of Methods Utilizing Temporal Influence



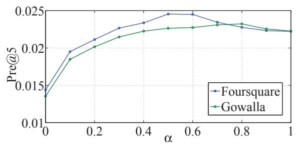
(a) Pre@N - Foursquare (b) Rec@N - Foursquare

(c) Pre@N - Gowalla (d) Rec@N - Gowalla

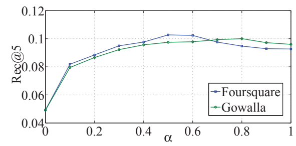
Figure 6: Performance of Methods Utilizing Spatial Influence

Methods Utilizing Spatial Influence. We compare the effectiveness of the proposed methods utilizing spatial influence with the state-of-the-art method utilizing spatial influence [23]. The precision and recall are plotted in Figure 6.

Observe from Figure 6, S achieves much better recommendation accuracy than SB in terms of both precision and recall, on both datasets. This clearly demonstrates the effectiveness of the proposed method. Moreover, the integration of temporal popularity (SE) further improves the precision and recall of the recommendation. Recall that in method SE, parameter β (Equation 11) tunes the weight between a POI's temporal popularity and long-term popularity. We tune β on the development data by varying it from 0.0 to 1.0, and observe that the best recommendation accuracy is achieved when $\beta = 0.5$ and 0.9 on Foursquare and Gowalla, respectively.

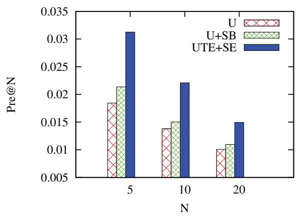


(a) Pre@5

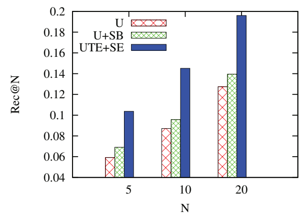


(b) Rec@5

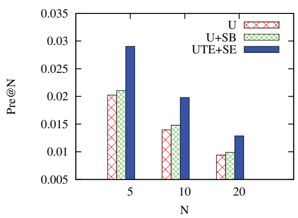
Figure 7: Tuning parameter α



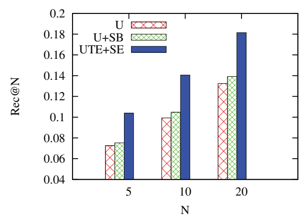
(a) Pre@N - Foursquare



(b) Rec@N - Foursquare



(c) Pre@N - Gowalla



(d) Rec@N - Gowalla

Figure 8: Performance of Unified Methods

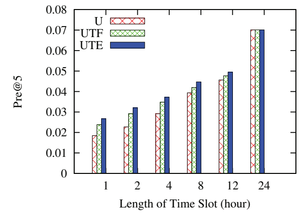
Unified Methods. The unified method (UTE+SE) combines the method utilizing temporal influence and the method utilizing spatial influence for time-aware POI recommendations. We select the state-of-the-art method U+SB as the baseline.

For UTE+SE, a parameter α is used to control the weights of the temporal part UTE and the spatial part SE (Equation 14). We tune α on the development data, and plot the average pre@5 and rec@5 on both datasets with different α values in Figure 7. From the figures, it is observed that best precision and recall are reached when $\alpha = 0.5$ and 0.8 on Foursquare and Gowalla, respectively. For method U+SB, the best results are achieved when $\alpha = 0.6$. Due to the space limitation, we omit the tuning results.

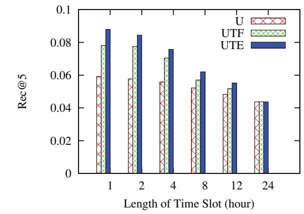
Setting α to the optimal values for both methods UTE+SE and U+SB, the precision and recall of the three methods (U, UTE+SE, and U+SB) on both datasets are reported in Figure 8. As shown in the figure, both U+SB and UTE+SE outperform U, indicating that exploiting spatial information can help improve the accuracy of time-aware POI recommendations. More importantly, observe that in terms of both precision and recall, UTE+SE outperforms U+SB by around 45% on Foursquare data, and around 38% on Gowalla data. This result demonstrates that our proposed method is superior to the state-of-the art method in terms of effectiveness.

6.3 Effect of the Length of Time Slot

This experiment is to study the effect of the length of time slot. The length of time slot controls the time granularity of time-aware POI recommendations. A larger length of time slot implies that the recommendation results will be less time-specific. Here we only consider the methods utilizing temporal influence to focus on the effect of the length of time slot. We report the prec@5 and rec@5 on the Foursquare data in Figure 9, since the results on the Gowalla data are similar. From Figure 9, we make two observations.



(a) Pre@5 - Foursquare



(b) Rec@5 - Foursquare

Figure 9: Performance of varying length of time slot

- The first observation is that as the time slot length increases, the precision of all methods increases as well, but the recall drops. The reason for better precision is, increasing the length makes the data denser, leading to better precision. Nevertheless, increasing the length of time slots will bring in more groundtruth POIs for each user at each time slot. With the number of recommendations (*i.e., N*) unchanged, poorer recall values are observed with increasing the length of time slot.
- A more important observation is that, for all lengths of time slots, the proposed method UTE outperforms the baseline method U. However, the amount of improvement decreases as the length of time slot getting larger, because increasing the length of time slot reduces the temporal influence. When the length is increased to 24-hour, the proposed method UTE reduces to the user-based CF method U.

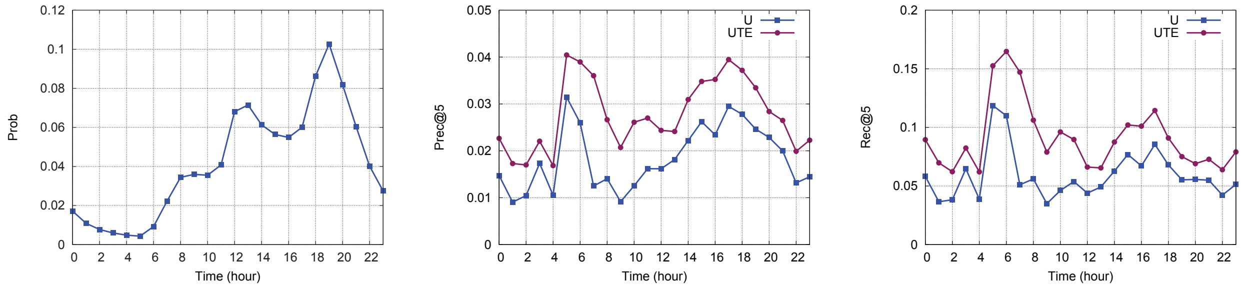
6.4 Case Study

As a case study, using the Foursquare data, we study the recommendation accuracy at different time slots in a day, (*i.e.,* at what time of a day, the POI recommendation is more accurate). We first show the check-in count distribution over time in Figure 10(a). From the figure, we find that users have more check-ins during 13:00–20:00 than in the morning or midnight. The curve reaches its peaks at 13:00 and 19:00, respectively, indicating that users are more active during or slightly after lunch and dinner hour.

The pre@5 and rec@5 of methods U and UTE are plotted in Figures 10(b) and 10(c) respectively. Observe from the figures, the proposed method UTE outperforms the baseline U over all the time slots and the percentage of relative improvement is quite stable. For both methods, the best recommendation accuracies (in terms of pre@5 and rec@5) are obtained between 5:00 and 6:00 as well as between 17:00 and 18:00. However, in these two periods, users are likely involved in very different activities and check-in different kinds of POIs. This is an interesting phenomenon that needs further study.

7. CONCLUSION AND FUTURE WORK

The availability of historical check-in data in LBSNs enables the POI recommendation service. In this paper, we define a new problem of time-aware POI recommendation as an extension of the conventional POI recommendation problem by considering the temporal influence in user activities. To the best of our knowledge, this is the first work on time-aware POI recommendations. The proposed solutions exploit both the temporal influence and spatial influence, which are specific for POI recommendations. We start with a new method that utilizes the temporal influence for POI recommendations. We then propose a new approach exploring the spatial influence. Lastly, we combine the two approaches through a unified framework. We conduct extensive experiments over two real-world



(a) Distribution of check-in in a day (b) Pre@5 - Foursquare (c) Rec@5 - Foursquare

Figure 10: Performance of different time of a day

LBSN datasets. The experimental results show that the proposed methods beat all baselines, and improve the accuracy of POI recommendations by more than 37% over the state-of-art method.

Several interesting directions exist for further exploration. First, in addition to hour, humans' check-in behavior is also influenced by the day of a week and even the month of a year. Hence we plan to exploit other time dimensions in POI recommendations. Second, POIs are often associated with category information in LBSNs. It would be promising to exploit category information in recommendation. In addition, we are also interested in incorporating the temporal influence and spatial influence in the POI prediction task.

8. ACKNOWLEDGEMENTS

This work is supported in part by a grant awarded by a Singapore MOE AcRF Tier 2 Grant (ARC30/12), a Singapore MOE AcRF Tier 1 Grant (RG66/12), and a grant awarded by Microsoft Research Asia. Quan Yuan would like to acknowledge the Ph.D. grant from the Institute for Media Innovation, Nanyang Technological University, Singapore.

9. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] B. Arnold. *Pareto Distribution*. International Cooperative Publishing House, Fairland, Maryland, 1983.

[3] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *PVLDB*, 3(1):1009–1020, 2010.

[4] 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.

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

[6] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. T. Reinders. Finding wormholes with flickr geotags. In *ECIR*, pages 658–661, 2010.

[7] Y. Ding and X. Li. Time weight collaborative filtering. In *CIKM*, pages 485–492, 2005.

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

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

[10] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, pages 195–202, 2009.

[11] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

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

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

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

[15] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *RecSys*, pages 189–196, 2009.

[16] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *ICWSM*, 2011.

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

[18] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. Artificial Intellegence*, 2009, 2009.

[19] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *KDD*, pages 723–732, 2010.

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

[21] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation: a generative model approach. In *SIGIR*, pages 671–680, 2012.

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

[23] 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.

[24] 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.