

# Unified Point-of-Interest Recommendation with Temporal Interval Assessment

Yanchi Liu  
Management Science and  
Information Systems  
Rutgers University  
yanchi.liu@rutgers.edu

Chuanren Liu  
Decision Sciences and MIS  
LeBow College of Business  
Drexel University  
chuanren.liu@drexel.edu

Bin Liu  
Management Science and  
Information Systems  
Rutgers University  
binben.liu@rutgers.edu

Meng Qu  
Management Science and  
Information Systems  
Rutgers University  
meng.qu@rutgers.edu

Hui Xiong<sup>\*</sup>  
Management Science and  
Information Systems  
Rutgers University  
hxiong@rutgers.edu

## ABSTRACT

Point-of-interest (POI) recommendation, which helps mobile users explore new places, has become an important location-based service. Existing approaches for POI recommendation have been mainly focused on exploiting the information about user preferences, social influence, and geographical influence. However, these approaches cannot handle the scenario where users are expecting to have POI recommendation for a specific time period. To this end, in this paper, we propose a unified recommender system, named the ‘Where and When to go’ (WWO) recommender system, to integrate the user interests and their evolving sequential preferences with temporal interval assessment. As a result, the WWO system can make recommendations dynamically for a specific time period and the traditional POI recommender system can be treated as the special case of the WWO system by setting this time period long enough. Specifically, to quantify users’ sequential preferences, we consider the distributions of the temporal intervals between dependent POIs in the historical check-in sequences. Then, to estimate the distributions with only sparse observations, we develop the low-rank graph construction model, which identifies a set of bi-weighted graph bases so as to learn the static user preferences and the dynamic sequential preferences in a coherent way. Finally, we evaluate the proposed approach using real-world data sets from several location-based social networks (LBSNs). The experimental results show that our method outperforms the state-of-the-art approaches for POI recommendation in terms of various metrics, such as F-measure and NDCG, with a significant margin.

<sup>\*</sup>Contact Author

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

KDD '16, August 13-17, 2016, San Francisco, CA, USA

© 2016 ACM. ISBN 978-1-4503-4232-2/16/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2939672.2939773>

## CCS Concepts

•Information systems → Location based services; *Geographic information systems*; •Computing methodologies → Factor analysis; •Human-centered computing → *Mobile computing*;

## Keywords

POI Recommendation; Sequential Preference

## 1. INTRODUCTION

The successful development of location-aware services, such as location-based social networks (LBSNs), has changed people’s lives. For example, Foursquare<sup>1</sup> reported 8 billion check-ins at 65 million point-of-interests (POIs) by over 55 million users as of December, 2015. Even in one local area, there are often multiple competing POIs with similar utilities, and individual users are not capable to make fully informed choices. Based on collective intelligence, recommending the right POIs to the right users thus becomes beneficial for users exploring new places with higher satisfactory level. The POI recommendations are also essential for service providers to improve service quality and attract new customers.

As we know, POI recommendation is a time-critical task [4, 19]. First of all, some POIs need more time to plan than others due to the capacity and budget issues. Let us consider to recommend a visit to a museum, such type of POI recommendation is preferred to be provided several days in advance so as to allow the user to be more prepared (e.g., learning the background of the museum). More importantly, the users’ needs and preferences often vary from time to time, and we need to capture the users’ dynamic needs and evolving preferences to deliver the right recommendations at the right time. For instance, if one visits a museum in one day, he/she might be more interested in going for shopping instead of visiting another museum in the following day. The *right POI* (e.g., a shopping center) for the coming trip or the *right time* for the next visit to a museum could be inferred from the historical check-in records. In summary, it

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

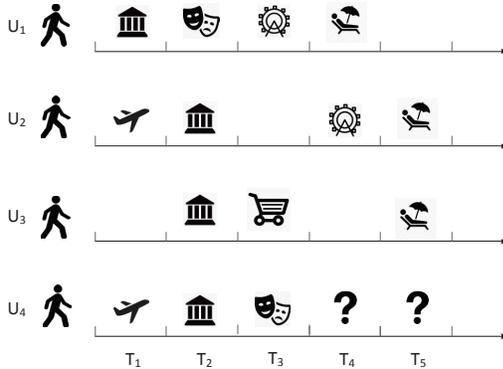


Figure 1: Examples of users check-in sequences.

is critical to investigate how to recommend the right POIs for a specific time period by learning the users’ evolving sequential preferences from their historical check-in records.

In the literature, various methods have been proposed for POI recommendation in recent years [3, 22, 12, 13]. Also, there are studies taking temporal factors into consideration for the purpose of improving the algorithm efficiency and/or effectiveness. For example, Yuan et al. [29] and Gao et al. [7] proposed to separate each day into different time slots and learn the users’ preferences for each slot for time-aware POI recommendations. However, these methods do not consider the temporal relationship between the related check-ins by considering all check-ins as ‘a bag of words’. Thus, they may recommend *museums in the morning* and *bars at night*, but the same recommendations will be provided for the same time period on different days. Indeed, these methods cannot capture the evolving changes of user preferences. The assumption of unchanging user preferences across several days may not hold due to the temporal relationship between dependent POI check-ins. More recently, the importance of the sequential relationships/patterns hidden in the historical check-in sequences has been realized for ‘next’ POI recommendations [2, 4, 6, 30]. In particular, given the current check-in POI, the ‘next’ POI recommendation predicts the next interested POI which is most likely to be visited. However, these studies are unable to recommend POI for a specific time period due to the lack of modeling temporal interval information in their methods.

To better motivate this work, let us consider the example in Figure 1. For illustration purpose, we assume the time unit is day, i.e.  $T_1$  is the first day and  $T_5$  is the fifth day in the records of each user. Given users’ check-ins to airport, museum, theater, shopping mall, amusement park and beach, what POIs should we recommend to user  $U_4$  at time  $T_4$  and  $T_5$ ? Traditional POI recommender systems may recommend beach for both time periods since beach appears most frequently with museum and theater, and the existing ‘next’ POI recommender systems will recommend amusement park for both since amusement park appears mostly after theater. However, intuitively, we can see amusement park should be more likely to be recommended for time  $T_4$ , and beach for  $T_5$  because users went to amusement park mostly two days after going to museum according to the history records, and

followed by going to beach. The possible reason could be that, after spending one day in museum, one would prefer to do something else to refresh the mind, e.g., by watching a show. Then, he/she would do something fun, e.g., going to amusement park, followed by relaxing on beach.

Indeed, in this paper, we investigate how to do POI recommendations for a specific time period in LBSNs by capturing users’ evolving sequential preferences from their historical check-in records. This task is much harder than the traditional POI recommendation due to the following challenges. First, it is necessary to model the sequential check-in patterns with temporal intervals between dependent POIs. Second, the sequential check-in data is very sparse. Only limited observations are available for estimating the distributions of the temporal intervals between dependent POIs in the historical check-in sequences.

To address these challenges, we propose a unified recommender system, named ‘Where and When to go’ (WWO), to integrate the static user interests and evolving sequential preferences with temporal interval assessment. Specifically, given POI sequences consisting of check-in POIs ordered by check-in time, we first assess the temporal intervals between POIs from check-in sequences of each user as a POI-POI transition matrix, where each item is a set of observed intervals for a POI-POI pair. Then, we develop a bi-weighted low-rank graph construction model to learn individual user’s behavioral preferences by identifying a set of common graph bases. The graph is bi-weighted so that the static user interests used by the traditional recommender systems are simultaneously learned with their evolving sequential preferences.<sup>2</sup> Finally, the main contributions of this paper can be summarized as follows:

- A new recommender system, named WWO, is developed for providing POI recommendations for a specific time period. WWO exploits user check-in sequential patterns with temporal interval assessment based on all historical user check-ins. Moreover, WWO is able to capture both static user interests and their evolving sequential preferences for POI recommendations.
- A bi-weighted low-rank graph construction model is proposed to estimate the distributions with only sparse observations. The model helps to identify a set of bi-weighted graph bases, which in turn can be leveraged for learning the user interests and their sequential preferences in a coherent way.
- The WWO recommender system has been evaluated on large-scale real-world data for POI recommendations. The experimental results show that our method outperforms state-of-the-art methods in terms of multiple metrics such as F-measure and Normalized Discounted Cumulative Gain (NDCG).

## 2. THE WWO RECOMMENDER SYSTEM

In this section, we introduce our WWO (Where and When to go) recommender system. Assume that we have  $M$  POIs denoted by the set  $\mathcal{P}$  and  $N$  users. For simplicity, we let  $\mathcal{P} = \{1, 2, \dots, M\}$ , i.e., we use integers to represent the POIs. For the  $n$ -th user,  $n = 1, 2, \dots, N$ , we have his/her

<sup>2</sup>In this way, the traditional recommendation methods can be treated as the special cases of our approach.

**Table 1: Descriptions of mathematical notations.**

Notation	Description
$M, N$	The number of POIs and users, respectively.
$\mathcal{P} = \{1, 2, \dots, M\}$	The set of POIs.
$s^n = (s_1^n, s_2^n, \dots, s_{L_n}^n)$	The check-in sequence of $n$ -th user.
$s_l^n = (p_l^n, t_l^n)$	The $l$ -th check-in event with POI $p_l^n \in \mathcal{P}$ and time $t_l^n$ .
$d_{ij}^n$	The observed temporal intervals from POI $i$ to POI $j$ in the sequence $s^n$ .
$\mathcal{N}(\mu_{ij}^n, \sigma_{\mu}^2)$	The distribution of $d_{ij}^n$ .
$c_p^n$	The observed visiting frequency to POI $p$ in the sequence $s^n$ .
$\text{Poisson}(\nu_p^n)$	The distribution of $c_p^n$ .
$\alpha \in \mathbb{R}^{N \times K}$	The approximating coefficients in low-rank graph construction.
$\beta^k \in \mathbb{R}^{M \times M}$	The edge weight matrix of the $k$ -th graph basis, for $k = 1, 2, \dots, K$ .
$\gamma \in \mathbb{R}^{K \times M}$	The node weights of all graph bases.
$\mathcal{N}(0, \sigma_{\alpha}^2), \mathcal{N}(0, \sigma_{\beta}^2), \Gamma(\eta, \theta)$	The priors of $\alpha$ , $\beta$ , and $\gamma$ , respectively.

check-in records represented as a sequence of check-in events  $s^n = (s_1^n, s_2^n, \dots, s_{L_n}^n)$  with the length  $L_n$ . Each check-in event  $s_l^n$ ,  $l = 1, 2, \dots, L_n$ , is a tuple  $s_l^n = (p_l^n, t_l^n)$  where  $p_l^n \in \mathcal{P}$  is the  $l$ -th POI in the check-in records ordered by the corresponding event time  $t_l^n$ . Therefore, we have  $t_{l'}^n \geq t_l^n$  when  $l' > l$ . Table 1 lists some notations used in this paper.

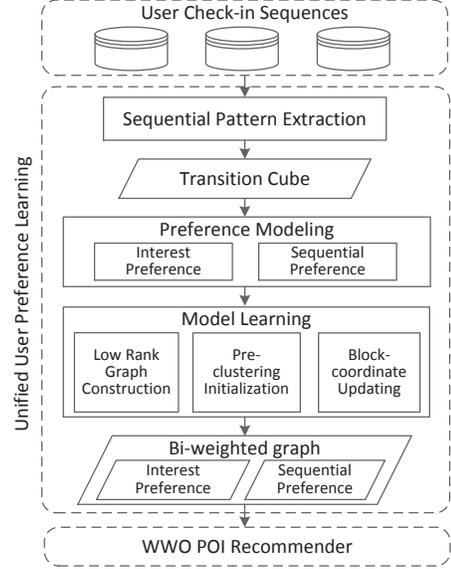
Given the historical check-in records of all the users  $\mathcal{S} = \{s^n | n = 1, 2, \dots, N\}$  and a future time  $t$ , we predict the  $n$ -th user's possible check-in, e.g.  $p \in \mathcal{P}$ , based on not only the user interest on  $p$  but also the temporal dependency between the prediction  $(p, t)$  and the historical records  $s_l^n = (p_l^n, t_l^n)$ .

Formally, our idea of WWO recommendations is to maximize the likelihood of check-in  $p \in \mathcal{P}$  at time  $t$ ,  $\Pr(p, t | s^n)$ , which is computed as:

$$\Pr(p, t | s^n) \propto f^n(p) \cdot g^n(t | p). \quad (1)$$

The term  $f^n(p)$  computes the user's *interest* on  $p$ , which is the focus of traditional POI recommender systems. The term  $g^n(t | p)$  computes the user's *evolving sequential preference*, which is the likelihood of the predicted check-in event  $(p, t)$  at time  $t$ . In other words, we aim at recommending the right POI at the right time to the right user. As such, we distribute the ongoing recommendations over time based on the historical temporal interval patterns so as to make the recommendation less disturbing and more favorable. In the following, we provide the details about computing these two terms with probability density functions.

Figure 2 shows the framework of our WWO recommender system. We construct a POI transition cube by extracting sequential patterns from user check-in sequences, then the user interests and their evolving sequential preferences are modeled and learned simultaneously with a low-rank graph construction model. Finally, by ranking the potential POIs


**Figure 2: The WWO recommendation framework.**

according to both sequential preferences and user interests for specific time period, we have our WWO POI recommender system.

## 2.1 Sequential Pattern Extraction

To extract the sequential patterns from historical check-in records, we first define:

**DEFINITION 1 (Observed Temporal Intervals).** *The observed temporal intervals from  $i$  to  $j$  in  $s^n$  are represented by a set:*

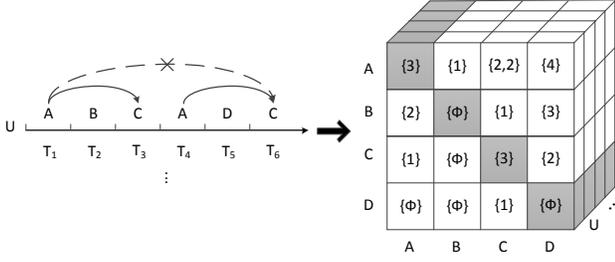
$$d_{ij}^n = \{ \min_{\substack{l' > l \\ s_{l'}^n = j}} t_{l'}^n - t_l^n \mid l = 1, 2, \dots, L_n, s_l^n = i \}. \quad (2)$$

Note that we use the min operator with several constraints in Equation 2 because both  $i$  and  $j$  can be visited by the same user multiple times. As a result, we consider only the soonest visit of  $j$  after each recent visit of  $i$ . With the temporal intervals between POI pairs extracted for user  $n$ , a POI-POI transition interval matrix is constructed, with each item  $d_{ij}^n$  as a set of intervals from POI  $i$  to POI  $j$  of user  $n$ . By combining the transition matrices for all the users, a transition interval cube can be obtained, which unifies both user interests and their sequential preferences. To provide an intuitive understanding, an example is shown in Figure 3 about how to generate the transition matrix for one user from a sequence of user check-ins.

## 2.2 Preference Modeling

*Sequential Preference:*  $g^n(t | p)$ . We compute the sequential preference term as:  $g^n(t | p) = \max_{1 \leq l \leq L_n} g_l^n(t | p)$ , where  $g_l^n(p, t)$  is the probability density of temporal interval  $(t - t_l^n)$  from the check-in of  $p_l^n$  at time  $t_l^n$  to the check-in of  $p$  at time  $t$ . We use the max operator to identify the historical check-in event  $s_l^n$  having the strongest temporal dependency with the recommendation  $(p, t)$ .

Generally, by letting  $s_l^n = i$  and  $p = j$ , we want to estimate a distribution of the temporal intervals when the user

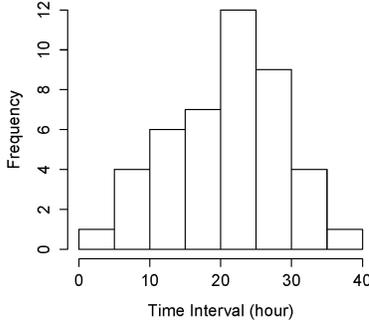


**Figure 3: Example of observed temporal intervals.** For the POI pair A to C, the temporal intervals with solid lines are counted, and we get  $\{2, 2\}$  as  $d_{AC}^U$ .

would visit  $j$  after visiting  $i$ . In this paper, we approximate this distribution using  $\mathcal{N}(\mu_{ij}^n, \sigma_\mu^2)$ . It follows that:

$$g_i^n(t|p) = \mathcal{N}(\delta | \mu_{ij}^n, \sigma_\mu^2) = \frac{1}{\sqrt{2\pi}\sigma_\mu} \exp\left(-\frac{(\delta - \mu_{ij}^n)^2}{2\sigma_\mu^2}\right),$$

where  $s_i^n = i, p = j$  and  $t - t_i^n = \delta$ . We have a histogram of one random POI pair in Figure 4 which shapes like a Gaussian distribution and indicates our model specifications are appropriate.



**Figure 4: Histogram of the temporal interval of one POI pair**

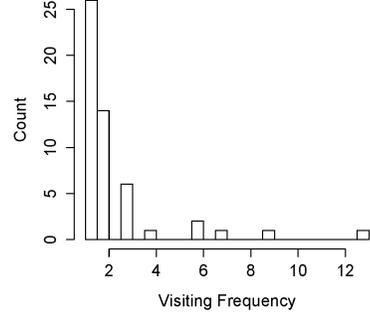
*Interest Preference:*  $f^n(p)$ . To compute the interest preference term, we also fit the observed visiting frequency with a distribution, e.g., Gaussian. However, the visiting frequencies in our data is polarized as shown in Figure 5, which does not shape like a Gaussian. As noticed by [1, 13], the nature of Poisson distribution is more suitable and effective for modeling the skewed interest preference in terms of visiting counts, which provide implicit feedback for better POI recommendations. Therefore, we adopt the Poisson distribution  $\text{Poisson}(\nu_p^n)$  to estimate  $f^n(p)$ :

$$f^n(p) = \text{Poisson}(c_p^n | \nu_p^n) = \frac{(\nu_p^n)^{c_p^n}}{c_p^n!} \exp(-\nu_p^n),$$

where  $c_p^n$  is the observed visiting frequency:

**DEFINITION 2 (Observed Visiting Frequency).** *The observed visiting frequency to  $p \in \mathcal{P}$  of the user in  $s^n$  is the cardinality of visiting records:*

$$c_p^n = |\{l \mid l = 1, 2, \dots, L_n, s_l^n = p\}|, \quad (3)$$



**Figure 5: Histogram of the visiting frequency of one user**

### 2.3 Model Learning

Both the observed temporal intervals and visiting frequency can be very sparse. Therefore, 1) we have empty  $d_{ij}^n = \emptyset$  or  $|d_{ij}^n|$  is too small, for many pairs of POI  $(i, j)$ ; 2) we have  $c_p^n = 0$  for many unobserved check-ins  $p \in \mathcal{P}$ . As a result, using these observations directly does not suffice to robustly estimate the temporal interval distributions and the user preferences. To address these issues, we develop the low-rank graph construction model in Section 3.

### 2.4 Recommendation

In practice, we aim to recommend the historically unvisited POIs within a time duration  $T$  instead of a single timestamp  $t$ . This can be achieved by ranking the options  $p$  according to the expected visiting frequency within the time duration  $T = [t, t + \Delta]$ :

$$\begin{aligned} & \mathbb{E}[c_p^n | \nu_p^n] \cdot \max_{1 \leq l \leq L_n} \int_t^{t+\Delta} g_l^n(t|p) dt \\ &= \nu_p^n \cdot \max_{1 \leq l \leq L_n} \left( \Phi\left(\frac{t + \Delta - t_l^n - \mu_{s_l^n p}^n}{\sigma_\mu}\right) - \Phi\left(\frac{t - t_l^n - \mu_{s_l^n p}^n}{\sigma_\mu}\right) \right) \end{aligned} \quad (4)$$

where  $c_p^n \sim \text{Poisson}(\nu_p^n)$  and the integral term can be easily computed with the cumulative distribution function (CDF)  $\Phi(\cdot)$  of standard  $\mathcal{N}(0, 1)$ . The top ranked options can then be recommended to the user. It's worthy to note that, when  $t \rightarrow t_{L_n}^n$  and  $\Delta \rightarrow \infty$ , the computation of Equation 4 will be dominated by the interest preference  $\nu_p^n$ . In other words, our unified recommender system is a proper generalization of the existing approaches.

## 3. LOW-RANK GRAPH CONSTRUCTION

Since the observations for specific users are sparse, we estimate the distribution parameters  $\mu^n$  and  $\nu^n$  for all users  $n = 1, 2, \dots, N$  collaboratively. To this end, we define the bi-weighted graph  $G^n = \langle \mathcal{P}, E, \mu^n, \nu^n \rangle$  with  $\mathcal{P}$  as graph nodes, where each graph node  $p \in \mathcal{P}$  is weighted by  $\nu_p^n$  and each graph edge  $(i, j) \in E$  is weighted by  $\mu_{ij}^n$ . In other words, the bi-weighted graph  $G^n$  is a unified representation of two aspects in the POI check-in behaviors of the  $n$ -th user: 1) The node weight vector  $\nu^n \in \mathbb{R}^M$  characterizes the distribution of user interests. 2) The edge weight matrix  $\mu^n \in \mathbb{R}^{M \times M}$  characterizes the distribution of the temporal intervals between the check-in events.

With this definition, we estimate the distribution parameters by collaboratively constructing the bi-weighted graphs for all the users. Specifically, we assume there are  $K$  graph bases  $B^k$  for  $k = 1, 2, \dots, K$  and each user-specific graph  $G^n$  can be approximated by:

$$G^n \leftarrow \sum_{k=1}^K \alpha_{nk} B^k. \quad (5)$$

Equivalently, by letting the edge weight matrix and the node weight vector of the graph basis  $B^k$  be  $\beta^k \in \mathbb{R}^{M \times M}$  and  $\gamma^k \in \mathbb{R}^M$ , respectively, it follows that:

$$\mu^n = \sum_{k=1}^K \alpha_{nk} \beta^k, \quad (6)$$

$$\nu^n = \sum_{k=1}^K \alpha_{nk} \gamma^k. \quad (7)$$

Here,  $\alpha_{nk}$  are the approximating coefficients. Note that, the number of graph bases,  $K$ , can be deemed the rank of the graph construction, which is set to be much smaller than the number of user-specific graphs, i.e.,  $K \ll N$ .

Now we can compute the probability density of the observations in  $d_{ij}^n$  and  $c_p^n$ :

$$\begin{aligned} & \Pr(d, c | \alpha, \beta, \gamma, \sigma_\mu) \\ &= \prod_{n=1}^N \prod_{i=1}^M \prod_{j=1}^M \prod_{\delta \in d_{ij}^n} \frac{1}{\sqrt{2\pi}\sigma_\mu} \exp\left(-\frac{(\delta - \mu_{ij}^n)^2}{2\sigma_\mu^2}\right) \\ & \quad \prod_{n=1}^N \prod_{p=1}^M \frac{(\nu_p^n)^{c_p^n}}{c_p^n!} \exp(-\nu_p^n) \end{aligned}$$

Therefore the log-likelihood is:

$$\begin{aligned} & \ln \Pr(d, c | \alpha, \beta, \gamma, \sigma_\mu) \\ &= -\frac{1}{2\sigma_\mu^2} \sum_{n=1}^N \sum_{i=1}^M \sum_{j=1}^M \sum_{\delta \in d_{ij}^n} (\delta - \mu_{ij}^n)^2 \\ & \quad - \frac{1}{2} \sum_{n=1}^N \sum_{i=1}^M \sum_{j=1}^M \sum_{t \in d_{ij}^n} \ln(2\pi\sigma_\mu^2) \\ & \quad + \sum_{n=1}^N \sum_{p=1}^M (c_p^n \ln(\nu_p^n) - \nu_p^n) - \sum_{n=1}^N \sum_{p=1}^M \ln(c_p^n!) \end{aligned}$$

Maximizing this log-likelihood is equivalent to minimize the following objective function:

$$\begin{aligned} \mathcal{L}(\alpha, \beta, \gamma) &= \frac{1}{2\sigma_\mu^2} \sum_{n=1}^N \sum_{i=1}^M \sum_{j=1}^M \sum_{\delta \in d_{ij}^n} (\delta - \mu_{ij}^n)^2 \\ & \quad - \sum_{n=1}^N \sum_{p=1}^M (c_p^n \ln(\nu_p^n) - \nu_p^n), \end{aligned}$$

where the hyperparameter  $\sigma_\mu$  is fixed. With the low-rank graph construction Equation 6 and Equation 7, it follows

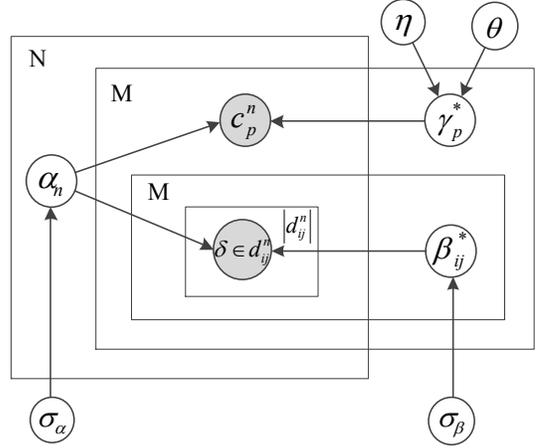


Figure 6: The probabilistic graphical model.

that

$$\begin{aligned} \mathcal{L}(\alpha, \beta, \gamma) &= \frac{1}{2\sigma_\mu^2} \sum_{n=1}^N \sum_{i=1}^M \sum_{j=1}^M \sum_{\delta \in d_{ij}^n} (\delta - \sum_{k=1}^K \alpha_{nk} \beta_{ij}^k)^2 \\ & \quad - \sum_{n=1}^N \sum_{p=1}^M (c_p^n \ln(\sum_{k=1}^K \alpha_{nk} \gamma_p^k) - \sum_{k=1}^K \alpha_{nk} \gamma_p^k). \end{aligned}$$

Indeed, the above factorization can be demonstrated as a probabilistic graphical model shown in Figure 6. Moreover, in order to reduce the generalization error (accuracy on unseen data), priors of the latent variables ( $\alpha, \beta, \gamma$ ) can be used as regularizations:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, \sigma_\alpha^2), \\ \beta &\sim \mathcal{N}(0, \sigma_\beta^2), \\ \gamma &\sim \Gamma(\eta, \theta). \end{aligned}$$

In particular, we use the Gamma distribution  $\gamma \sim \Gamma(\eta, \theta)$  since it is the conjugate one with Poisson. Then the joint log-likelihood to be minimized is as follows:

$$\begin{aligned} \mathcal{L}(\alpha, \beta, \gamma) &= \frac{1}{2\sigma_\mu^2} \sum_{n=1}^N \sum_{i=1}^M \sum_{j=1}^M \sum_{\delta \in d_{ij}^n} (\delta - \sum_{k=1}^K \alpha_{nk} \beta_{ij}^k)^2 \\ & \quad - \sum_{n=1}^N \sum_{p=1}^M (c_p^n \ln(\sum_{k=1}^K \alpha_{nk} \gamma_p^k) - \sum_{k=1}^K \alpha_{nk} \gamma_p^k) \\ & \quad + \frac{1}{2\sigma_\alpha^2} \|\alpha\|_F^2 + \frac{1}{2\sigma_\beta^2} \sum_{k=1}^K \|\beta^k\|_F^2 \\ & \quad - \sum_{k=1}^K \sum_{p=1}^M ((\eta - 1) \ln \gamma_p^k - \theta \gamma_p^k), \quad (8) \end{aligned}$$

where  $\alpha \in \mathbb{R}^{N \times K}$  is the matrix of approximating coefficients  $\alpha_{nk}$ ,  $\beta^k \in \mathbb{R}^{M \times M}$  is the edge weight matrix of graph basis  $B^k$ , and  $\gamma \in \mathbb{R}^{K \times M}$  is the matrix of node weights  $\gamma_p^k$  in all graph bases.

## 4. LEARNING ALGORITHM

We use an alternative algorithm to solve  $\alpha$ ,  $\beta$ , and  $\gamma$  iteratively. Here, we let

$$D_{ij}^n = \sum_{\delta \in d_{ij}^n} \delta, C_{ij}^n = |d_{ij}^n|.$$

Also, at the beginning of each iteration, we compute  $\mu$  and  $\nu$  with Equation 6 and Equation 7 with current solution  $(\alpha, \beta, \gamma)$ . Then we have

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \alpha_{nk}} &= \frac{\sum_{ij} (D_{ij}^n - C_{ij}^n \mu_{ij}^n) \beta_{ij}^k}{-\sigma_\mu^2} - \sum_p \left( \frac{c_p^n}{\nu_p^n} - 1 \right) \gamma_p^k + \frac{\alpha_{nk}}{\sigma_\alpha^2}, \\ \frac{\partial \mathcal{J}}{\partial \beta_{ij}^k} &= \frac{\sum_n (D_{ij}^n - C_{ij}^n \mu_{ij}^n) \alpha_{nk}}{-\sigma_\mu^2} + \frac{\beta_{ij}^k}{\sigma_\beta^2}, \\ \frac{\partial \mathcal{J}}{\partial \gamma_p^k} &= - \sum_n \left( \frac{c_p^n}{\nu_p^n} - 1 \right) \alpha_{nk} - \left( \frac{\eta - 1}{\gamma_p^k} - \theta \right). \end{aligned}$$

Although it is straightforward to update one entry at a time with arbitrary initialization for all entries in  $\alpha$ ,  $\beta$ , and  $\gamma$ , we improve the robustness of the alternative algorithm using: 1) the pre-clustering initialization based on spherical KMeans; and 2) the block-coordinate updating [21].

### 4.1 Pre-clustering Initialization

Specifically, we first propose a novel clustering method to initialize the graph bases and the approximating coefficients. Note that, since the graph bases are used to approximate the distribution parameters  $\mu$  and  $\nu$ , the bases can be initialized with the representative observed distribution samples. Therefore, we aggregate and cluster the observed temporal intervals and visiting frequencies. To aggregate the observations, we construct the feature matrix  $X \in \mathbb{R}^{N \times (M^2 + M)}$ , where each row  $X_n$  corresponds to one sequence  $s^n$ . The first  $M^2$  entries of  $X_n$  encodes the average temporal interval:

$$X_{n, (i-1)M+j} = D_{ij}^n / C_{ij}^n.$$

The last  $M$  entries of  $X_n$  encodes the observed visiting frequency:

$$X_{n, M^2+p} = c_p^n.$$

Then, we apply the spherical KMeans algorithm to group the rows of  $X$  into  $K$  clusters, where  $K$  is the number of graph bases for the low-rank graph construction. Here, we use spherical KMeans with cosine similarity instead of Euclidean distance since the dimension of  $X$  is high.

Although the results of this pre-clustering may not be the true graph bases, we believe the clustering results can provide positive guidance for the graph basis learning algorithm. Thus, given the centroid  $\hat{X}_k$  of the  $k$ -th cluster in the pre-clustering solution, we initialize the graph basis  $B^k$  by reformatting the first  $M^2$  entries of  $\hat{X}_k$  as the matrix  $\beta^k$  and the last  $M$  entries as the vector  $\gamma^k$ . Moreover, we initialize the approximating coefficients  $\alpha$  with the cosine similarity between  $X_n$  and  $\hat{X}_k$  used by the spherical clustering:

$$\alpha_{nk} = \frac{\cos(X_n, \hat{X}_k)}{\sum_{k'} \cos(X_n, \hat{X}_{k'})}.$$

### 4.2 Block-coordinate Updating

With the above pre-clustering initialization, we then iteratively update rows in  $\alpha$  where each row corresponds to one user. Then we update each graph basis  $B^k$  with the edge weight matrix  $\beta^k$  and the node weight vector  $\gamma^k$ . For each updating, the learning rate  $\epsilon$  is determined to make sure the objective function  $\mathcal{J}(\alpha, \beta, \gamma)$  is decreasing. The overall algorithm repeats until the value of  $\mathcal{J}(\alpha, \beta, \gamma)$  keeps stable or the limitation on the number of iterations is reached. More details are shown in Algorithm 1.

---

#### Algorithm 1 Block-coordinate Optimization

---

- 1: Initialize  $\alpha$ ,  $\beta$ , and  $\gamma$  with pre-clustering.
  - 2: **repeat**
  - 3:   **for**  $n = 1, \dots, N$  **do**
  - 4:     Update the row  $\alpha_n \leftarrow \alpha_n - \epsilon \frac{\partial \mathcal{J}}{\partial \alpha_n}$ .
  - 5:   **end for**
  - 6:   **for**  $k = 1, \dots, K$  **do**
  - 7:     Update the base  $\beta^k \leftarrow \beta^k - \epsilon \frac{\partial \mathcal{J}}{\partial \beta^k}$ .
  - 8:   **end for**
  - 9:   **for**  $k = 1, \dots, K$  **do**
  - 10:     Update the row  $\gamma_p^k \leftarrow \gamma_p^k - \epsilon \frac{\partial \mathcal{J}}{\partial \gamma_p^k}$ .
  - 11:   **end for**
  - 12: **until** Convergence
- 

## 5. EXPERIMENTAL RESULTS

In this section, we evaluate the performances of the proposed WWO recommender system.

### 5.1 The Experimental Data

All the experiments were performed on real-world datasets including two LBSN datasets collected from Foursquare and Gowalla<sup>3</sup>, which are shown in Figure 7.

**The Foursquare dataset** includes 2,932 users for 4,194 POIs with 121,678 check-in observations. Each user checked into 41.5 POIs on average. Each check-in contains the user ID, check-in time, venue ID and the venue’s geo-coordinates. **The Gowalla dataset** includes 1,313 users for 2,196 POIs with 45,410 check-in observations. Each user checked into 34.58 POIs on average. Note that, for Foursquare and Gowalla data, we remove those POIs with less than 10 users and remove users with less than 10 check-ins.

### 5.2 Evaluation Metrics

In our experiments, the first 80% check-ins are used as the training data, and the other 20% are used as the testing data. We learn the models and obtain the recommendation POI list for a future time period  $T_n = [t_n + \delta, t_n + \Delta]$  for each user  $n$ , where  $t_n$  is the latest check-in time in the training data. Specifically, we are interested in investigating the following questions: (1) How WWO performs with different lengths of time period from now, which means  $\delta = 0$  and  $\Delta$  varies; (2) How WWO performs with specific time period from different future time, which means  $\Delta - \delta$  is a constant and  $\delta$  varies.

Note that if there is no check-in existing in the  $T_n$  period, this user will not be included for evaluation. Also, the visited POIs in the training data have been removed in the testing data since recommending new POIs is our target. Finally,

<sup>3</sup><https://snap.stanford.edu/data/loc-gowalla.html>

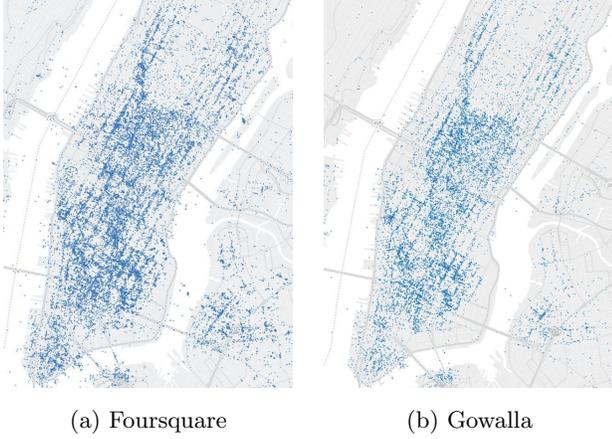


Figure 7: Data visualization

the evaluation metrics include F-measure and the NDCG of the newly visited POIs.

**F-measure.** F-measure combines precision and recall together with a harmonic mean. Here we use the  $F_\beta$  measure with  $\beta = 0.5$  which puts more emphasis on precision than recall [13],

$$F_\beta@P = (1 + \beta^2) \frac{\text{Precision}@P \times \text{Recall}@P}{\beta^2 \cdot \text{Precision}@P + \text{Recall}@P}. \quad (9)$$

Given a top- $P$  recommendation list  $S_{rec}$  sorted in a descending order based on the prediction values, precision and recall can be obtained as follows:  $\text{Precision}@P = \frac{S_{rec} \cap S_{new}}{P}$ ,  $\text{Recall}@P = \frac{S_{rec} \cap S_{new}}{S_{new}}$ , where  $S_{new}$  are the POIs a user newly visited in the test data. The precision and recall for the entire recommender system are computed by averaging all the precision and recall values of all the users.

**Normalized Discounted Cumulative Gain (NDCG).** Given a top- $P$  recommendation list sorted in a descending order of the prediction values, NDCG [10] is defined as

$$\text{NDCG}@P = \frac{1}{\text{IDCG}} \times \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log(i + 1)}, \quad (10)$$

where IDCG is the maximum possible DCG for a given set of recommendations, and  $rel_i$  is 1 if the recommended POI at position  $i$  is visited by the user and 0 otherwise. NDCG measures the ranking quality of the recommender system based on a graded relevance scale of recommendations, which assumes more interested POIs should appear earlier in the recommendation list. The NDCG for the entire recommender system are computed by averaging all the NDCG values of all the users.

### 5.3 Baselines

Our proposed system focuses on assessing user evolving sequential preferences together with user interests for POI recommendation. We thus introduce the following baseline methods for comparison.

- Probabilistic Matrix Factorization (PMF) [18] is a generalized matrix factorization model, which has been widely used for recommendation tasks.
- Factorized Personalized Markov Chains (FPMC) [20]

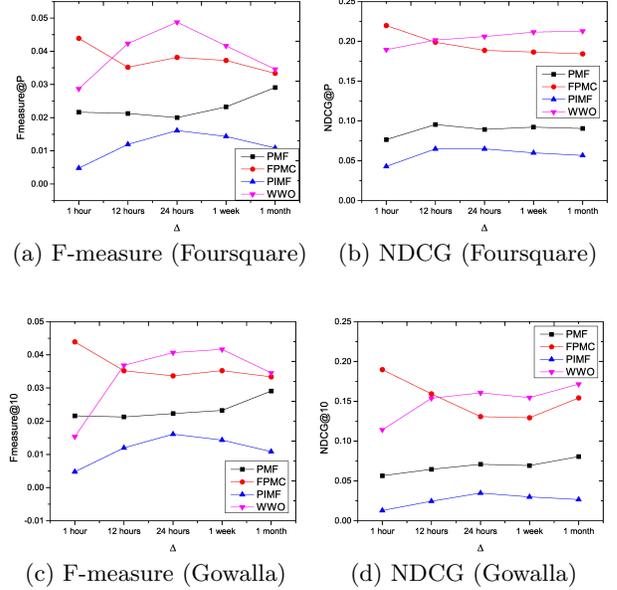


Figure 8: The performances for different  $\Delta$ , with  $\delta = 0$ .

embeds users' preferences and their personalized Markov chain to provide next basket item recommendation. An extended factorized personalized Markov chain with localized region constraint (FPMC-LR) [4] was proposed for POI recommendation while considering only the neighborhood locations. Since we only focus on POI recommendation and try to explore the time interval influence, so we do not consider the location constraint here.

- Purchase Interval based Matrix Factorization (PIMF) [32] shows that the intervals between user purchases have an influence on a user's purchase decision, and thus PIMF incorporates this purchase interval factor into matrix factorization for recommendation.

### 5.4 POI Recommendation Performances

In this subsection, we present the performance comparison on the recommendation accuracy between WWO and baselines. The results are based on Foursquare and Gowalla data by setting the latent dimensions to  $K = 10$ .

We first investigate how WWO performs with different lengths of time period from now. When  $\delta = 0$  (recommending for the next  $\Delta$  time period), the performances in terms of F-measure and NDCG with respect to  $\Delta$  (from 1 hour to 1 month) on Foursquare and Gowalla data are shown in Figure 8. In the figures, we can see that FPMC and WWO obtain better overall performances than PMF and PIMF. For a short next time period, such as 1 hour, FPMC outperforms others significantly. When  $\Delta$  is increased to 24 hours and 1 week, WWO achieves the best performance. This implies that the temporal interval information between check-ins can be helpful for identifying the temporal relationship between POIs, especially better captured for a granularity of several days. This result is consistent with the observation for the time interval histogram shown in Figure 9, where we can see that the intervals between POIs are mostly dis-

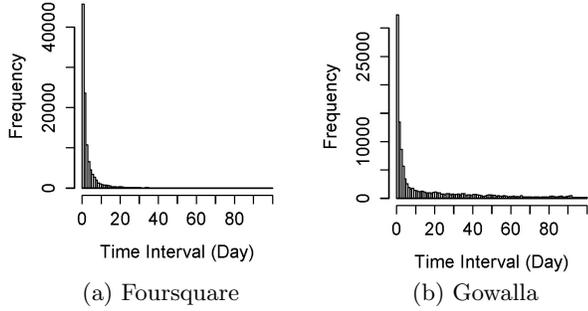


Figure 9: The histogram of all user check-in intervals

tributed in the first 7 days. Moreover, as the increase of  $\Delta$ , the performance of PMF is getting closer to FPMC and WWO. This is because PMF does not consider sequential preferences. POI recommendation for a long term is mainly determined by user preferences and the temporal factor is no longer significant. Finally, PIMF does not work well in POI recommendation, mainly because check-in data include higher level of noise and are very sparse compared to purchase data. In fact, PIMF cannot capture the intervals between POIs without considering the interval distribution.

We then investigate how WWO performs with specific time period from different future time. The recommendation results for one specified day (i.e.,  $\Delta - \delta = 24$  hours) with respect to various  $\delta$  are shown in Figure 10. We illustrate the results of  $\delta$  from 0 to 6 because these are the most effective time periods to recommend. In the figure, we can see that WWO obtains better performances than others. The reason is that all the other methods cannot make recommendations for a specific time period.

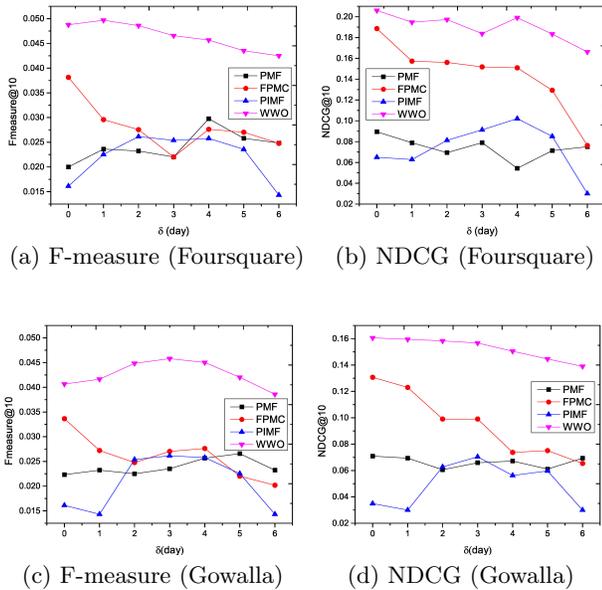


Figure 10: The performances for different  $\delta$ , with  $\Delta - \delta = 24$  hours.

## 5.5 Parameter Selection

For simplicity, we only present the parameter selection results on Foursquare data. First, we use the empirical standard deviation in  $\cup_{n,i,j} d_{ij}^n$  as  $\sigma_\mu$  in  $\mathcal{N}(\mu_{ij}^n, \sigma_\mu^2)$ . Then, we determine the number of graph bases  $K$  based on the recommendation performances. As shown in Figure 11 (a), we plot the results with increasing number of graph bases. It is worthwhile to note that the performance might not increase with more bases. The reason is that more bases imply higher modeling complexity and may lead to overfitting in the training data and decreasing generality of the identified graph bases. From Figure 11, we see that  $K = 10$  is a feasible trade off between the modeling complexity and the empirical accuracy. Thus, we choose  $K = 10$ .

Due to space limitations, we omit the similar procedures for tuning the prior parameters such as  $\sigma_\alpha$ ,  $\sigma_\beta$ ,  $\eta$ , and  $\theta$ .

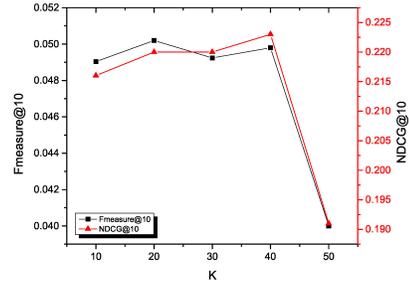


Figure 11: The performances at different number of graph bases  $K$  when  $\delta = 0$ ,  $\Delta = 24$  hours

## 5.6 Time Complexity

The overall asymptotic computational complexity of Algorithm 1 is  $O(NM^2K^2T)$  where  $T$  is the number of iterations. Although the complexity is higher than the conventional static recommender systems, it is comparable with other state-of-the-art approaches utilizing the temporal patterns. Moreover, due to the pre-clustering initialization proposed in Section 4.1, our method converges quickly. Table 2 reports the running time of all methods.

Table 2: The running time (seconds).

Data	PMF	FPMC	PIMF	WWO
Foursquare	79.34	1431.67	27416.5	1378.03
Gowalla	17.96	273.97	4301.62	287.56

## 6. RELATED WORK

In this section, we will first introduce relevant studies on POI recommendation, followed by the general recommendation tasks with sequential information.

### 6.1 POI Recommendation

POI recommendation, targeting at recommending the right POIs to the target users, has been an important task in understanding user mobility [33, 11, 28, 17]. Unlike other recommender systems based on explicit user feedback, such as user ratings, POI recommendation is developed based on implicit user feedback, such as the check-in frequency. Recently, other implicit information, such as check-in locations,

check-in time, and transition between POIs, have been exploited for POI recommendations.

Among methods proposed in previous studies, latent factor model is the most used algorithm to solve this task. Specifically, each user’s check-in times for every POI are counted, which serve as the preferences of POIs. With this user-POI check-in matrix constructed, Collaborative Filtering (CF) methods such as probabilistic matrix factorization (PMF) [18] are used to detect and recommend potential POIs which may be interested by users. Collaborative filtering algorithm is also used to fuse the check-in information, e.g., user interest preferences, social influence, temporal influence and geographical influence [29, 7, 34]. In [27], Ye et al. considered the social influence under the framework of a user-based CF model, and modeled the geographical influence by a model-based method (a Bayesian CF algorithm). To exploit the social influence, the authors made use of the users’ friends for recommendation rather than all the users. On the other hand, to explore the geographical influence, they assumed that the probability that a user visited two POIs was determined by their distance, the larger the distance the smaller the probability. Moreover, Yuan et al. [29] and Gao et al. [7] introduced temporal preference to enhance the algorithm efficiency and effectiveness. The authors separated a day into different time slots and user preferences were learned for each slot, thus POIs can be recommended according to different times of a day. Cheng et al. [3] considered more comprehensive information, such as the multi-center of user check-in patterns, and the skewed user check-in frequency. However, this work lacked an integrated consideration of factors that can influence POI recommendation. To improve the ad hoc integration between them, Liu et al. [13] proposed a geographical probabilistic factor analysis framework to analyze the joint effects of multiple factors by considering user preference for locations as a multiplication of interest in the locations, location popularity and distance between user and POIs.

The above works focus on evaluating the relationships between POIs and check-in features, such as location or time, while the relationship between POIs has rarely been considered. Recent works have shown the fact that human movement exhibits sequential patterns [9, 23, 5], suggesting that users usually follow some sequential behaviors when visiting POIs. In light of this, Cheng et al. [4] considered the task of next POI recommendation, in an attempt to recommend POIs to users for their next visits. This work took user check-in sequential information into account, assuming the next check-in is dependent on last check-in. Zheng et al. [31] further extended the next POI recommendation with sequential information by considering not only the latest visited POI but also the earlier visited POIs with a  $n$ -order Markov chain, and integrating geographical and social influence into the proposed method. Feng et al. [6] also proposed a metric embedding model to learn the personalized sequential information for next POI recommendation. However, these methods did not consider the temporal interval information between dependent POIs, thus cannot capture the sequential interval patterns to make recommendations for a specified future time period.

Different from the above works, we consider the sequential preferences with temporal interval assessment between check-ins. And, the temporal interval distribution for different users and POI pairs are estimated with a novel factor-

ization method, which enables our method to make recommendation for a specific time period.

## 6.2 Sequential Information and Recommendation

Most recommender systems rely on statistical models that use the event history of users on items to produce recommendations. Also, as an important information source for understanding user preferences, the information about user sequential behaviors have often been utilized in recommender systems. Indeed, sequential information is critical for many time-sensitive data mining tasks [15, 14], such as travel package recommendations [16, 8] and marketing campaign recommendations [25], since users’ preferences change over time in those application scenarios.

Recent works have shown that sequential patterns can be utilized to improve personalized recommendations at the right time. For example, Rendle et al. [20] proposed a factorized personalized Markov chain (FPMC) model to recommend the products which user will probably buy in the next visit. Specifically, a transition matrix from product to product was constructed for each user, thus a transition cube is formed for all the users. Then, the transition tensor was estimated by a factorization model to propagate information among similar users, similar items and similar transitions. Additionally, Wang et al. [24] proposed an opportunity model to estimate the follow-up purchase probability of a user at a specific time. To recommend the best next-items to each target user, Yap et al. [26] learned user-specific sequential knowledge through personalized sequential pattern mining. Moreover, the time interval information between purchase transactions was used to improve the performance of next-product recommendations [32].

Similarly, POI recommendation is also a time-sensitive task, since people’s interests for POIs are drifting over time. Therefore, the POI recommendations will be more effective if the recommender system can capture the users’ evolving sequential preferences. In order to make recommendations at the right time, the interval information between check-ins should be considered. Along this line, this paper focuses on POI recommendations for a specific time period by considering users’ evolving sequential preferences with temporal interval assessment.

## 7. CONCLUDING REMARKS

In this paper, we investigated how to exploit both user interests and their evolving sequential preferences for recommending POIs for a specific time period. Along this line, we first proposed a unified framework to integrate user interests and their sequential preferences. Specifically, the distributions of the temporal intervals between dependent POIs were studied for measuring users evolving sequential preferences based on their historical check-in sequences. Here, to address the challenge of estimating the distributions with only sparse observations, we developed a bi-weighted low-rank graph construction model to identify a set of bi-weighted graph bases, which in turn can be leveraged for learning user interests and their sequential preferences in a coherent way. As shown in the experimental results on real-world data, unlike existing POI recommender systems, the proposed WWO recommender system can provide effective POI recommendations for a future specified time period by capturing user evolving sequential preferences.

## 8. ACKNOWLEDGEMENT

This research was partially supported by Futurewei Technologies, Inc. Also, it was supported in part by Natural Science Foundation of China (71329201) and the Rutgers 2015 Chancellor's Seed Grant Program.

## References

- [1] J. Canny. Gap: a factor model for discrete data. In *SIGIR*, pages 122–129. ACM, 2004.
- [2] Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *ICDE*, pages 900–911. IEEE, 2011.
- [3] 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*, pages 17–23, 2012.
- [4] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2013.
- [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] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, pages 2069–2075. AAAI Press, 2015.
- [7] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, 2013.
- [8] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware travel tour recommendation. In *KDD*, pages 983–991. ACM, 2011.
- [9] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.
- [10] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *TOIS*, 20(4):422–446, 2002.
- [11] 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.
- [12] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, 2013.
- [13] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao. A general geographical probabilistic factor model for point of interest recommendation. *TKDE*, 27(5):1167–1179, 2015.
- [14] C. Liu, F. Wang, J. Hu, and H. Xiong. Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *KDD*, pages 705–714. ACM, 2015.
- [15] C. Liu, K. Zhang, H. Xiong, G. Jiang, and Q. Yang. Temporal skeletonization on sequential data: patterns, categorization, and visualization. *TKDE*, 28(1):211–223, 2016.
- [16] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu. A cocktail approach for travel package recommendation. *TKDE*, 26(2):278–293, 2014.
- [17] Y. Liu, C. Liu, N. J. Yuan, L. Duan, Y. Fu, H. Xiong, S. Xu, and J. Wu. Intelligent bus routing with heterogeneous human mobility patterns. *KAIS*, pages 1–33, 2016.
- [18] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.
- [19] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *ICDM*, 2012.
- [20] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.
- [21] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Prog.*, 144(1-2):1–38, 2014.
- [22] J. Sang, T. Mei, J.-T. Sun, C. Xu, and S. Li. Probabilistic sequential pois recommendation via check-in data. In *GIS*, 2012.
- [23] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
- [24] J. Wang and Y. Zhang. Opportunity model for e-commerce recommendation: right product; right time. In *SIGIR*, pages 303–312. ACM, 2013.
- [25] J. Yang, C. Liu, M. Teng, H. Xiong, M. Liao, and V. Zhu. Exploiting temporal and social factors for b2b marketing campaign recommendations. In *ICDM*, pages 499–508. IEEE, 2015.
- [26] G.-E. Yap, X.-L. Li, and S. Y. Philip. Effective next-items recommendation via personalized sequential pattern mining. In *Database Systems for Advanced Applications*, pages 48–64. Springer, 2012.
- [27] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 2011.
- [28] Z. Yu, H. Xu, Z. Yang, and B. Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE T HUM-MACH SYST*, 46(1), 2015.
- [29] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, 2013.
- [30] J.-D. Zhang and C.-Y. Chow. Spatiotemporal sequential influence modeling for location recommendations: A gravity-based approach. *TIST*, 7(1):11, 2015.
- [31] J.-D. Zhang, C.-Y. Chow, and Y. Li. Lore: Exploiting sequential influence for location recommendations. In *GIS*, 2014.
- [32] G. Zhao, M. L. Lee, W. Hsu, and W. Chen. Increasing temporal diversity with purchase intervals. In *SIGIR*, pages 165–174, 2012.
- [33] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*, 2010.
- [34] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian. Mining mobile user preferences for personalized context-aware recommendation. *TIST*, 5(4):58, 2015.