R2SIGTP: a Novel Real-Time Recommendation System with Integration of Geography and Temporal Preference for Next Point-of-Interest

Conference Paper · May 2019

DOI: 10.1145/3308558.3314120

Xu Jiao
Tjut Nyak Dhien University

Yingyuan Xiao
Tianjin University of Technology

Wenguang Zheng
Tianjin University of Technology

Hongya Wang
Donghua University

A Novel Next New Point-of-Interest Recommendation System based on Simulated User Travel Decision-Making Process

Anonymous Author(s)

ABSTRACT

POI(point of interest) recommendation systems have been widely investigated in recent years. Currently, most POI recommendation systems only recommend POIs that may be visited by users in the future, and rarely consider next new POI recommendation based on the current time and the current location of a particular user. In fact, next new POI recommendation problem is more challenging for the reason that multiple factors associated with both POIs and users need to be comprehensively incorporated in a unified recommendation system. In this paper, we design a novel and effective next new POI recommendation system. Our system simulates a user's travel decision-making process by weighing two important factors that affect a user's travel decision: preference factors and geographic factors. First, we use tensor to model user's check-in history and dynamically predict user preferences. Then, in order to characterize the influence of geographic factor on individual users, we designed a personalized user similarity calculation method and fitted curves for the target user to reflect the relationship between travel distance and travel probability. Finally, a recommendation list is generated by combining the effects of these two factors on a particular user. Compared with the state-of-the-art POI recommendation approach, the experimental results demonstrate that our system achieves much better performance.

CCS CONCEPTS

• **Social and professional topics** → **Geographic characteristics**;

KEYWORDS

Next New POI Recommendation, Tensor, Location Based Social Networks, Preference

1 INTRODUCTION

In recent years, with the rapid development of mobile Internet technology, positioning technology, wireless sensor technology and the popularity of smart phones, location-based social networks(LBSNs) as shown in Figure 1 and its application services have developed rapidly. The currently popular LBSNs are Foursquare, Gowalla, Facebook Place, Microsoft GeoLife, Bikely, Flickr, Panomamio, etc. The LBSNs represented by Foursquare, Gowalla, and Facebook Place mainly provide check-in services for POIs. Encourage users to share their favorite POIs with friends in the form of check-in and share their experiences and tips for POIs. The main difference between LBSNs and Online Social Networks (OSN) is that LBSNs add geographical location information. Location data bridges the gap between the physical and digital worlds and enables a deeper understanding of users' preferences and behaviors. Since users generate a large amount of check-in data in LBSNs, it is possible to recommend the unvisited POIs to users. POI recommendation can help users better understand their city and explore the surrounding environment. Therefore, POI recommendation is of high value to both users and the business owners of POIs.

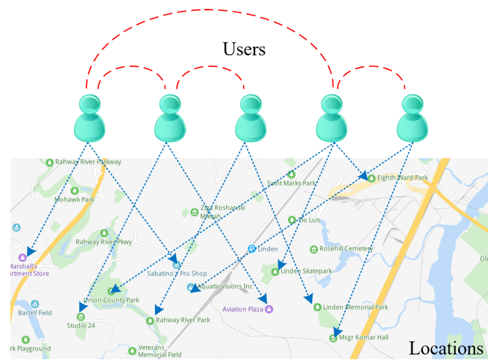


Figure 1: location based social networks

POI recommendation is one of the most important tasks in LBSNs, which is to provide recommendations of places to users, and has attracted much attention in both research and industry. However, the general POI recommendation system can only recommend POIs globally and ignores the time context that the recommendation result itself should have. Specifically, the general POI recommendation system only considers which POIs users may access in the future, and they cannot predict where users want to go in the next time interval. Furthermore, the general POI recommendation system does not consider whether the POIs recommended to a user have been previously accessed by this user. If the recommendation results are not novel enough, it will seriously affect the user's experience. In order to provide users with a POI recommendation system more practical and good experience, we have taken next new POI recommendation system as the research object. The next new POI recommendation is an extension of the general POI recommendation. It does not merely recommend the POIs that the user may visit in the future. Next new POI recommendation system takes into account the sequential influence and recommends the POIs that the user may access at the succeeding moment according to various contextual information such as current time and current location. And these POIs were never visited by this user before.

Conference'17, July 2017, Washington, DC, USA
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Compared to general POI recommendation, next new POI recommendation is a more difficult task. In order to achieve more accurate and personalized recommendations, next new POI recommendation system will face more difficult challenges as follows:

- Next new POI recommendation system needs to consider a user's current location, because the current location is the starting point for the user to travel at the succeeding moment. The POIs that the user may access during the next time interval are closely related to the current location.
- Next new POI recommendation emphasizes the real time nature of the recommendation system. This requires the recommendation system to be able to focus on the dynamic changes of the user's preferences in real-time, and give the user satisfactory recommendation results based on the user's preferences at the current moment and the most recent check-in.
- Next new POI recommendation system needs to take into account the geographic factors of the recommendation results. Specifically, the recommendation results are POIs that the user may access from the current location in the next time interval. If these POIs are too far away from the user's current location, there is no feasibility of travel.

Our system is focused on next new POI recommendation. The main contributions of this paper can be summarized as follows:

- Our system simulates a user's travel decision-making process and considers two important factors that influence people's travel destination choices: preference factors and geographic factors, and integrates them into a unified recommendation process.
- We leverage tensor to model users' check-in history based on POI's classification properties and check-in data's time information, which overcomes the sparseness of check-in data. Further, we realize the dynamic prediction of users' preferences.
- We propose a novel approach to modeling the personalized impact of geographic factors on individual user travel. Specifically, we fit a curve to personally reflect the relationship between the user's travel distance and travel probability. Due to the sparseness of the check-in data of individual users, we established a virtual common access sequence for two users and designed a novel user similarity algorithm to find users with similar historical behavior to the target user, using similar users' check-in data as supplement of the target user's data.
- We conduct comprehensive experiments by comparing our approach with the state-of-the-art techniques over two real-world datasets.

2 RELATED WORK

POI recommendations using user's check-in history are often influenced by multiple factors such as geography, time, sequence and society. Geographical influences are the most essential feature that distinguishes POI recommendations from traditional recommendations. Since the user's check-in behavior presents a spatial clustering phenomenon, geographical influence can be modeled by power law distribution [24], Gauss distribution [4], Poisson distribution [25], and kernel density estimation [17, 28, 29]. Temporal influence [6, 10, 12, 16, 26, 27, 32] in a POI recommendation system performs in two aspects: periodicity and non-uniformness. And the sequential influence [19, 31, 33] is the result of the interaction of temporal periodicity, adjacency of POIs in geographical space, property of POIs and human habits. Inspired by the assumption that friends in LBSNs share more common interests than non-friends, social influence [9, 23, 30] is explored to enhance POI recommendation. Most existing POI recommendation systems use the methods based on content, link analysis [1, 3, 34], collaborative filtering [8, 15, 20, 23, 26] or matrix factorization [2, 4, 18, 25, 35] in the recommendation process.

At present, there is relatively little research on the next POI recommendation. Anastasios Noulas et al. [14] propose a set of features that aim to capture the factors that may drive users movements. They further extend their study combining all individual features in two supervised learning models, based on linear regression and M5 model trees, resulting in a higher overall prediction accuracy. Chen Cheng et al. [5] propose a novel matrix factorization method, namely FPMC-LR, to embed the personalized Markov chains and the localized regions. Jing He et al. [13] propose to adopt a third-rank tensor to model the successive check-in behaviors. By incorporating softmax function to fuse the personalized Markov chain with latent pattern, and they furnish a Bayesian Personalized Ranking (BPR) approach and derive the optimization criterion accordingly. Shanshan Feng et al. [11] propose a personalized ranking metric embedding method (PRME) to model personalized check-in sequences. They further develop a PRME-G model, which integrates sequential information, individual preference, and geographical influence, to enhance the recommendation performance.

3 PROBLEM DEFINITION

In order to facilitate our system, we have the following notations:

(1) U**:** the set of the entire users.
(2) F**:** the set of all the preferences. Each POI has its own category, such as western restaurant, shopping center, park, and so on. The category of a POI that a user has visited usually implies the user's travel preferences. In our system, we refer to the categories of POIs as user preferences.
(3) L**:** the set of the entire POIs. Each POI $l \in L$ is represented as $l =< x, y, f >$, where x and y denote the latitude and longitude respectively and $f \in F$ is the preference which l belongs to.
(4) P**:** the set of all the check-ins. Specifically, each check-in $p \in P$ is represented as $p =< l, t, u >$, where l denotes

the POI of check-in, t is the check-in time, and u is the check-in user.

We then formalize the next new POI recommendation problem as follows. Given the current time t_c and the current location l_c of user u_i, a set R can be obtained. $R = \left\{ l \in L - L_{u_i} \right\}$, where L_{u_i} is the set of POIs visited by u_i before time t_c. The next new POI recommendation is to calculate the ranking score of each POI belonging to the set R, and then rank the set R in descending order according to the ranking score. Finally, the top k POIs are recommended to user u_i.

Compared to general POI recommendation, it's challenging to estimate the implicit transition probability of potential new POIs based on the sparse historical data. Thus, the next new POI recommendation problem is much more difficult than general POI recommendation problem. Correspondingly, Precision and Recall of the next new POI recommendation system will be relatively low.

4 OUR SYSTEM MODEL

Our system focuses on the next new POI recommendation, which can help users make travel decision. In this paper, the travel decision refers to how a person chooses a travel destination at the succeeding moment. In real life, people's travel decisions are influenced by many factors, the most important of which are preference factors and geographic factors. Consider only these two most critical factors we divide a person's travel decision-making process into three steps as follows:

(1) **Preference Decision:** A person's travel decision-making process is first to determine the categories of travel destinations. In this paper, such categories are represented by preferences. Without losing generality, a person usually has multiple preferences when making travel decisions. This requires sorting the preferences in descending order according to intensity.

(2) **Understanding Distance:** According to Tobler's First Law of Geography: everything is related to everything else, but near things are more related than distant things, we can know that people are more inclined to visit places closer to themselves. Therefore, the influence of travel distance in geographic factors is very important for travel decision. In this paper, understanding distance refers to a person's perception of whether a certain distance is far or near. It is well known that different people have different feelings about a specific distance. And the same person's feelings about the same distance may also be different in different contexts. For example, a person in the morning of the weekend would think that 10km is a relatively close distance, while on a working day of the night 10km becomes a relatively distant travel distance. So the second step in a person's travel decision-making process is for the person to determine his or her own understanding of distance based on his or her current context.

(3) **Destination Decision:** After completing the first two steps, this person can determine the final destination based on preferences and travel distance. For example, a person's first preference for a certain time interval is going to a park and the second preference is shopping. The nearest park to the person's current location is 30 km and the nearest shopping mall is 1 km. Then choosing the park or the shopping mall requires the person to weigh these two factors: preference factors and geographic factors. Specifically, If the preference factors dominate, although the park is far away, and he or she would still choose the park. And if the geographic factors dominate, he or she will choose the shopping mall.

In order to simulate the user travel decision-making process described above, our system designed three modules: preference dynamic prediction module, geographical influence curve fitting module and next new POI recommendation module to simulate the above three steps respectively. The preference dynamic prediction module can predict and rank users' preferences according to the change of time. The geographical influence curve fitting module can personally reflect the relationship between travel probability and travel distance of a single user in different time intervals. The next new POI recommendation module can calculate the ranking scores of the POIs to be recommended based on the predicted preference probabilities and the travel probabilities of the user at different distances and generate a recommendation list. Our system framework is shown in Figure 2, and the detailed description is as follows:

(1) Using the preference dynamic prediction module, the preferences of the target user are predicted according to the current time and the preference probability of the corresponding preference is obtained.

(2) Use the geographical influence curve fitting module to fit the travel probability and travel distance curves of the target user in different time intervals. And select the corresponding fitting curve according to the current time.

(3) Generated a recommendation candidate set based on the current time and the preferences predicted in (1). And the distance between each POI in the recommendation candidate set and the current location is calculated.

(4) Use the curve selected in (2) and the distance calculated in (3) to get the target user's travel probability with respect to the distance for each POI in the recommendation candidate set.

(5) Calculate the ranking score of each POI in the recommendation candidate set according to the preference probability predicted in (1) and the travel probability calculated in (4) using the next new POI recommendation module.

(6) The recommendation candidate set is ranked in descending order according to the ranking score calculated in (5) to obtain a recommendation list.

The specific details of the three modules are as follows.

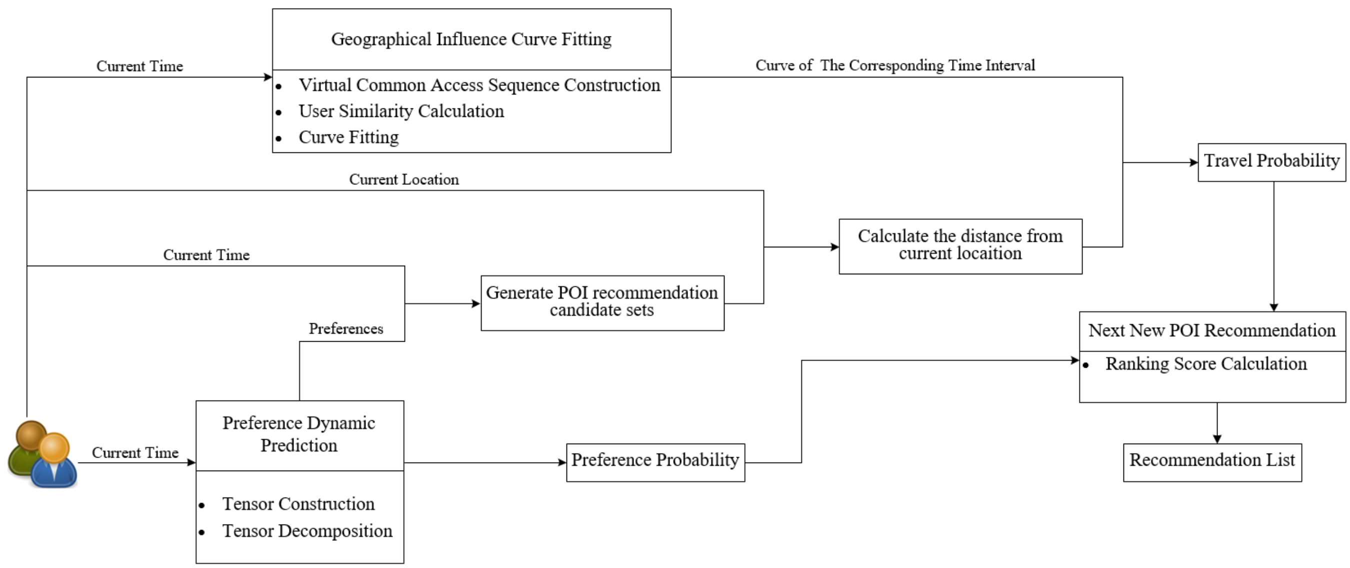


Figure 2: framework of our system

4.1 Preference Dynamic Prediction

Dynamic prediction of user preferences is an important part of the system. The accuracy of users preferences predicted in different time slots determines the accuracy of our system. Our system uses tensor to model user temporal preference for the following reasons:1) tensor is typically used to recover the missing/sparse data through tensor decomposition; 2) another usage of tensor is to isolate and analyze the patterns hidden in a dataset. Next, we will show the details of tensor construction and decomposition.

Tensor Construction: We define a 3 order-tensor $\chi \in \mathbb{R}^{I\times J\times K}$ to model user temporal preference, where I, J and K denote the size of user, time and preference dimension, respectively. Each entry of the tensor $\chi(i,j,k)$ is a sum of check-in frequencies of all the POIs which belong to preference k and are visited by user i in the time slot j. Note that we divide everyday into 6 time slots (i.e. 0:00-7:00, 7:00-9:00, 9:00-12:00, 12:00-14:00, 14:00-18:00, 18:00-0:00) in accordance with the law of most people's work and life. Accordingly, a week is divided into 42 time slots. Each time slot is represented by a Time ID. In particular, we select preference instead of the venues used in most tensor models as a dimension, which can greatly overcome the sparsity of check-in data. Because the amount of POIs in the real world is enormous, but only a few hundreds of preferences can describe our real life in detail.

Tensor Decomposition: After the tensor is modeled, our task is to infer missing entries in the tensor. We can extract the latent features of each user, time slot and preference by decomposing the tensor χ into three matrixes A, B, C and a core tensor G. As is shown in Equation1.

$$\chi \approx G \times_1 A \times_2 B \times_3 C = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr} a_p \circ b_q \circ c_r \quad (1)$$

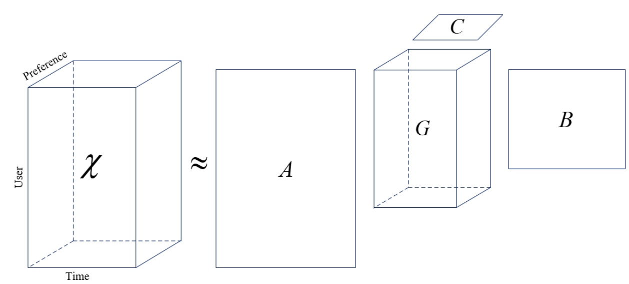


Figure 3: HOSVD decomposition of a 3 order-tensor

Here, $A \in \mathbb{R}^{I\times P}$, $B \in \mathbb{R}^{J\times Q}$, and $C \in \mathbb{R}^{K\times R}$ are the factor matrices. $G \in \mathbb{R}^{P\times Q\times R}$ is the core tensor. There are many methods for tensor decomposition. Our system use HOSVD approach to obtain approximate tensor $\hat{\chi}$. As shown in Figure 3.

4.2 Geographical Influence Curve Fitting

In this subsection, we will describe the details of the geographical influence curve fitting module. We tend to build a curve for each user to quantitatively measure the probability of POI visiting w.r.t. the travel distance. Besides, different time slots need to be considered. Thus, we actually build six curves for each user according to six time slots determined in section 4.1. Due to the sparsity of check-in data, for a target user, we exploit other similar users' check-ins to collaboratively build a curve. Therefore, our first task is to calculate the similarity between users to find similar users of the target user.

4.2.1 User Similarity Calculation. In this subsection, we illustrate our similarity calculation approach aiming at finding the most similar users for each particular user. Before going into detail, it is necessary to first define some terms that will be used in the future.

Definition 4.1 (Check-in History). For each user $u_i \in U$, a check-in history H_i of user u_i is a sequence of check-ins formatted as $H_i = p_1 \xrightarrow{\triangle t_1} p_2 \xrightarrow{\triangle t_2} \ldots \xrightarrow{\triangle t_{n-1}} p_n$, where $p_j \in P$, $\triangle t_j = p_{j+1}.t - p_j.t$ and $\forall 0 < j < n$, $p_{j+1}.t > p_j.t$. Besides, there is no $p_{j'}$ between p_j and p_{j+1} to make $p_j.t < p_{j'}.t < p_{j+1}.t$.

Our approach focuses on users' multiple check-in behavior within a short time interval. There is no need to investigate two check-in records with large time spans. Thus, the definition of continual check-ins is given.

Definition 4.2 (Continual check-ins). Given a time threshold $\triangle t'$. Based on check-in history H_i of user u_i, if the time interval $\triangle t_j$ of two check-ins p_j and p_{j+1} is less than $\triangle t'$. The two check-ins p_j and p_{j+1} are regarded as continual check-ins.

At present, there are more methods for calculating user similarity, such as Euclidean distance, Cosine similarity, Jaccard similarity coefficient, and so on. Although these traditional similarity calculation methods have achieved good results in many fields, there are two disadvantages in using them to calculate user similarity in LBSNs as follows:

(1) The similarity of users calculated using traditional methods has no directionality. That is to say, similarity(u_i, u_q) is equivalent to similarity(u_q, u_i). In reality, this may not accurately reflect the similar relationship between two users. For example, in the check-in history of LBSNs, user u_i has visited a total of 50 POIs, and a total of 300 POIs are accessed by user u_q. User u_i's 50 POIs are completely contained in the 300 POIs of u_q. That is, u_q has visited all POIs that u_i has visited. Obviously, for user u_i, user u_q is very similar. On the contrary, it is not the case for u_q. In other words, the similarity between a pair of users has a directional attribute.

(2) Most of the traditional user similarity calculation methods only consider the single factor of user preference, but ignore the equally important factors such as time, geography and sequence.

Our user similarity calculation method is more advanced than the traditional method, and overcomes the above two weaknesses, mainly in: First, our approach is based on the target user directly building a virtual common access sequence with other users to calculate user similarity. Therefore, the user similarity has directionality. Secondly, our method takes into account multiple factors such as preference, time, geography and sequence, and utilizes the historical behavior of users accessing POIs belonging to the same preference in the same geographical area in the same time interval. Next we will first introduce how to build a virtual common access sequence between target user u_i and other user u_q. Specific steps are as follows:

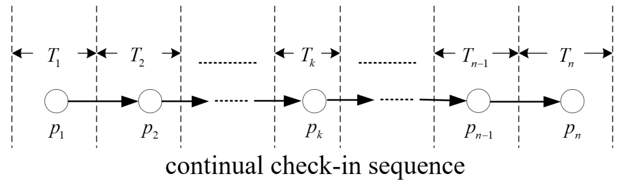


Figure 4: time interval

(1) **Continual Check-in Sequence:** Given a check-in history H_i of user u_i. We look for the longest sub-sequence of continual check-ins in the current check-in history. After we find it, we remove the sub-sequence from the check-in history and repeat the process in the remaining check-in history until the sub-sequence length is less than 2. We refer to this sub-sequence as the continual check-in sequence. All such sub-sequences of user u_i can be described as $SEQ = \{seq_1, ser_2, ..., seq_m\}$. And $LEN = \{len_1, len_2, ..., len_m\}$ is a set of the length of the corresponding continual check-in sequence of SEQ.

(2) **Time Interval Calculation:** For each $seq_j \in SEQ$, we artificially construct conjoint time intervals according to the corresponding check-in time in seq_j. Specifically, given a continual check-in sequence $seq_j = \{p_1, p_2, ...p_n\}$. We consider the following three cases to determine the time intervals and an example is shown in Figure 4.

(a) Based on the first check-in p_1 of seq_j, the first time interval is calculated in Equation2.

$$T_1 = [p_1.t - \frac{p_2.t - p_1.t}{2}, p_1.t + \frac{p_2.t - p_1.t}{2}] \tag{2}$$

(b) Based on the last check-in p_n of seq_j, the last time interval is determined as shown in Equation3.

$$T_n = [p_n.t - \frac{p_n.t - p_{n-1}.t}{2}, p_n.t + \frac{p_n.t - p_{n-1}.t}{2}] \tag{3}$$

(c) Otherwise, the time interval is calculated by Equation4 according to the check-in time of each $p_k \in seq_j$.

$$T_k = [p_k.t - \frac{p_k.t - p_{k-1}.t}{2}, p_k.t + \frac{p_{k+1}.t - p_k.t}{2}] \tag{4}$$

(3) **POIs clustering:** For all POIs belong to same preference, we exploit MeanShift algorithm[7] to cluster these POIs according to their latitude and longitude. Then we get a set of clusters $C = \{c_{ij} : 1 \leq i \leq |F|, 1 \leq j \leq |C_i|\}$, where c_{ij} denotes the jth cluster belonging to the ith preference, and C_i is the collection of clusters that belong to the ith preference.

(4) **Virtual common access sequence construction:** According to each seq_j of target user u_i, we construct virtual common access sequence of user u_i and u_q, denoted by $seq_j^{iq} = \{s_1, s_2, ...s_n\}$, where each $s_k \in seq_j^{iq}$ is a set of POIs. The method of generating s_k

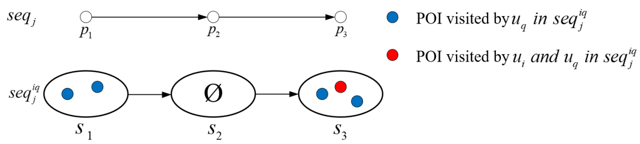


Figure 5: virtual common access sequence of user u_i and u_q

according to $p_k \in seq_j$ is as follows: First, according to the POIs clustering results, find the cluster c_k where $p_k.l$ is located. Then calculate the corresponding time interval T_k according to $p_k.t$. Finally, we look for the POIs in the c_k that u_q visited during the time interval T_k. And s_k is called a virtual common access collection. Obviously, s_k may be empty. Specifically, Figure 5 illustrates an example of constructing a virtual common access sequence.

Through the above construction of the virtual common access sequence, we can see that the sequence well describes that two users access POIs belonging to the same preference in the same geographical area in the same time interval. Therefore, the virtual common access sequence shows the similar historical behaviors between two users. So we can use this virtual common access sequence to calculate the similarity between the two users. The value of the similarity between target user u_i and other user u_q can be calculated according to the following three factors: sequential property, common access property, visited popularity of virtual common access collection.

(1) **Sequential property:** Our approach not only considers the similar historical behavior of two users from one POI to another, but also considers the continuity of similar historical behavior, that is, the length of a virtual common access sequence. The longer the virtual common access sequence of two users, the more related the two users might be.

(2) **Common access property:** Our approach takes into account POIs that two users have commonly accessed. Obviously, the more such POIs are in the virtual common access sequence, the more similar the two users are.

(3) **Visited popularity of virtual common access collection:** Our approach takes into account visited popularity of virtual common access collection. This is similar to inverse document frequency(IDF). Fewer people access POIs in the virtual common access collection of two users, the two users might be more correlated.

Our approach calculates the similarity between target user u_i and other user u_q as follows:

$$sim(u_i, u_q) = \sum_{j=1}^{m} f(len1[j]) sim(seq_j^{iq}) \tag{5}$$

$$sim(seq_j^{iq}) = \frac{g(canum) \sum_{k=1}^{len_j} V(s_k)}{len1[j]} \tag{6}$$

$$V(s) = \begin{cases} \frac{1}{|s|} \sum_{r=1}^{|s|} log \frac{|U|}{n_r}, & s \neq \emptyset \\ 0, & s = \emptyset \end{cases} \tag{7}$$

Specifically, in Equation5, m denotes the number of u_i's continual check-in sequence. $len1[j]$ represents the number of consecutive similar historical behavior, that is, the length of a virtual common access sequence. $canum$ in Equation6 is the number of POIs that two users commonly accessed in the virtual common access sequence seq_j^{iq}. In Equation7, n_r is the number of users visiting rth POI in the virtual common access collection s.

Given the target user u_i and other user u_q, Our method calculates their similarity by summing the similarity scores of all virtual common access sequences of two users in a weighted way. The function $f(len1[j])$ is used to assign larger weights to longer virtual common access sequences, e.g., $f(len1[j]) = 2^{len1[j]-1}$.

The similarity score of a virtual common access sequence, $sim(seq_j^{iq})$, is calculated by summing up the visited popularity $V(s_k)$ of each virtual common access collection s_k contained in seq_j^{iq}. And the $sim(seq_j^{iq})$ is weighted in terms of the number of POIs that two users commonly accessed in the virtual common access sequence seq_j^{iq}, e.g., $g(canum) = 2^{canum}$. Meanwhile, $sim(seq_j^{iq})$ is normalized by $len1[j]$. The algorithm for calculating user similarity is shown in Algorithm 1.

4.2.2 Curve Fitting. Our approach hopes to quantify the impact of geographic factors on people's travel. Because the data of a single user is extremely sparse, it is very difficult for us to analyze the personal effects of geographic factors on a single user. Therefore, we use the calculated user similarity to find users who have similar travel behaviors with a single user, and use these similar users' check-in data as a supplement to fit a curve to personally reflect the relationship between the user's travel distance and travel probability. The details of our method are shown as follows:

(1) Given user u_i and threshold α, user u_q that satisfies $sim(u_i, u_q) > \alpha$ constitutes a similar user group for user u_i . The optimal value of α will be shown in the experimental section.

(2) The purpose of fitting the curve is to reveal the probability of user u_i traveling as the travel distance increases. Obviously, a user has different considerations for travel distance during different time slots in a day. Therefore, the time factor should not be ignored in the curve fitting process. We divide everyday into six time slots determined in section 4.1.

(3) In each time slot, according to the check-in data of user u_i and all users in u_i's similar user group, a curve that reflects the relationship between u_i's travel distance and travel probability is fitted. The specific method is as follows:

   (a) In order to eliminate the discrepancy between users, the check-in times of user u_i and all users in u_i's similar user group are normalized to obtain the check-in frequency of each user.

Algorithm 1: User similarity calculation method

Input: u_i: the target user
U: the set of entire users
C: the set of POIs clusters
SEQ: the set of continual check-in sequences of target user u_i, $SEQ = \{seq_1, ser_2, ..., seq_m\}$
LEN: the set of the length of the corresponding continual check-in sequence of SEQ, $LEN = \{len_1, len_2, ..., len_m\}$
Output: the similarity between target user u_i and other user u_q

begin
  for *each* $u_q \in U$ *and* $u_q \neq u_i$ **do**
    Construct sim;
    for *each* $seq_j \in SEQ$ **do**
      Construct an empty array $len1$ and initialize to 0; /*$len1$ stores the length of each virtual common access sequence.*/
      Construct $canum$ and initialize to 0; /*$canum$ stores the number of POIs that two users commonly accessed in the virtual common access sequence.*/
      for *each* $p_k \in seq_j$ **do**
        Find $c_{kj} \in C$ which contains $p_k.l$;
        Calculate T_k according to $p_k.t$;
        Construct $s_k = \{l_r : l_r \in c_{kj}$ and l_r is visited by u_q during $T_k\}$
        Construct an empty array v; /*v stores the visited popularity of each virtual common access collection s_k.*/
        if $s_k \neq \emptyset$ **then**
          $len1[j] = len1[j] + 1$;
          if $p_k.l \in s_k$ **then**
            $canum = canum + 1$;
          end
          Construct an empty array n;
          for *each* $l_r \in s_k$ **do**
            Construct a set B which contains all the users who visited l_r;
            $n[r] = |B|$;
          end
          $v[k] = \frac{1}{|s_k|}\sum_{r=1}^{|s_k|} \log \frac{|U|}{n[r]}$;
        end
        else
          $v[k] = 0$;
        end
      end
      Construct an empty array d;
      $d[j] = \frac{g(canum)\sum_{k=1}^{len_j} v[k]}{len1[j]}$
    end
    $sim = \sum_{j=1}^{m} f(len1[j]) * d[j]$;
  end
end

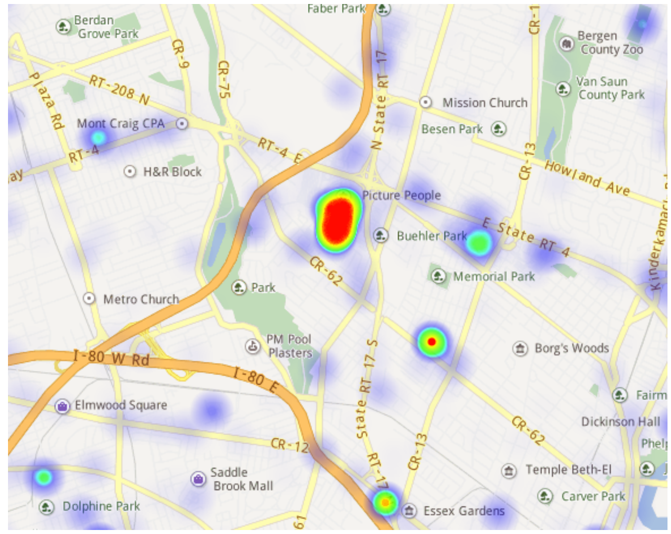


Figure 6: heat map

(b) Accumulate the frequency of check-ins for each POI by a single user to obtain the total frequency of check-ins for that POI. And calculate the sum N_{total} of the check-in frequency of all POIs.
(c) According to the check-in frequency of all POIs, a heat map is generated. As shown in Figure 6. And select the center of the hottest zone as the virtual starting point.
(d) Calculate the Euclidean distance from each POI to the virtual starting point. And let the maximum distance be d_{max}.
(e) Draw a set of concentric circles with the virtual starting point as the center of the circles. The radius r_i of each circle C_i starts from 0 and increments to d_{max} by step k. We choose k to be 500 meters.
(f) The sum N_{C_i} of the check-in frequency of all POIs falling in each circle C_i is calculated. Then we calculate the travel probability according to the set of concentric circles by Equation8.

$$P(v|C_i) = 1 - \frac{N_{C_i}}{N_{total}} \tag{8}$$

(g) Take the radius r_i of each circle C_i in (e) as the abscissa, and the travel probability in (f) as the ordinate, use the least squares method to fit the curve.

4.3 Next New POI Recommendation

Next new POI recommendation module combines preference factors and geographic factors to generate a recommendations list. Specifically, given the current time t_c and the current location l_c of user u_i, our system uses the preference dynamic prediction module to predict the preferences of u_i at the current time t_c. The POIs belonging to these predicted preferences that u_i has not visited before the current time t_c

constitute a recommendation candidate set CS_{u_i,t_c}. Our task is to calculate the ranking score of each POI in the candidate set CS_{u_i,t_c} and then sort the candidate set in descending order according to the ranking score. Finally, the top-n POIs are recommended to user u_i. The ranking score of each POI l_j belonging to CS_{u_i,t_c} is as Equation 9.

$$RS_{l_j}(u_i, t_c, l_c) = pre_{l_j}(u_i, t_c) + tp_{l_j}(u_i, t_c, l_c) \tag{9}$$

where $pre_{l_j}(u_i, t_c)$ is called preference probability, which represents the probability that the preference of POI l_j is selected by user u_i at the current time t_c, and $tp_{l_j}(u_i, t_c, l_c)$ denotes u_i's travel probability according to the distance from the current location l_c to POI l_j at current time t_c. Next, we will show the details of preference probability and travel probability based on user u_i's current time t_c and current location l_c.

Preference Probability: According to the time slot t that the current time t_c belongs to, a preference vector $v_{i,t}$ can be obtained by the approximate tensor $\hat{\chi}$ constructed in the preference dynamic prediction module. And each entry $v_{i,t,j} \in v_{i,t}$ with a positive value represents the probability of each preference j which is chosen by u_i in the time slot t. Then, the preference probability $pre_{l_j}(u_i, t_c)$ can be obtained from $v_{i,t}$ according to the preference of POI l_j.

Travel Probability: According to the current time t_c of u_i, the fitting curve of u_i in the corresponding time slot t can be obtained from the geographic influence curve fitting module. Calculate the distance between u_i's current location l_c and POI l_j. Then, the travel probability $tp_{l_j}(u_i, t_c, l_c)$ can be obtained by using the calculated distance and the selected fitting curve.

5 EXPERIMENTS

In this section we will evaluate the effectiveness of our approach. The experiments are set up as the following.

5.1 Experimental Setting

5.1.1 Datasets. In the experiments, we use the two real large-scale LBSNs datasets: Foursquare and Gowalla. The Foursquare dataset is provided by [22] and includes two cities, New York and Tokyo. And the Gowalla dataset comes from [6], which contains New York. In order to make the recommendation results more accurate, we filter out noise and invalid check-ins for both datasets. The basic statistics of them are shown in Table 1.

Table 1: Dataset Statistic

Dataset	New York (Foursquare)	Tokyo (Foursquare)	New York (Gowalla)
Users	807	1857	257
Venues	38196	60988	9762
Check-ins	225864	571812	97562

5.1.2 Experimental Data Partition. To study the effectiveness of the proposed method, we use the check-ins in the last month as test set, the second-to-last month as tuning set, and the rest months as training set.

5.1.3 Evaluation Metrics. Precision, Recall and F1 Score are used to evaluate the effectiveness of our approach. Precision is the ratio of the number of POIs correctly predicted to the total number of recommended POIs. Recall is the ratio of the number of POIs correctly predicted to the total number of POIs actually accessed. F1 Score is a comprehensive evaluation metric based on Precision and Recall.

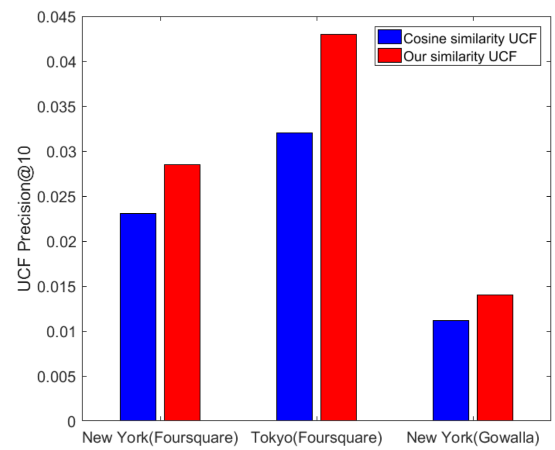


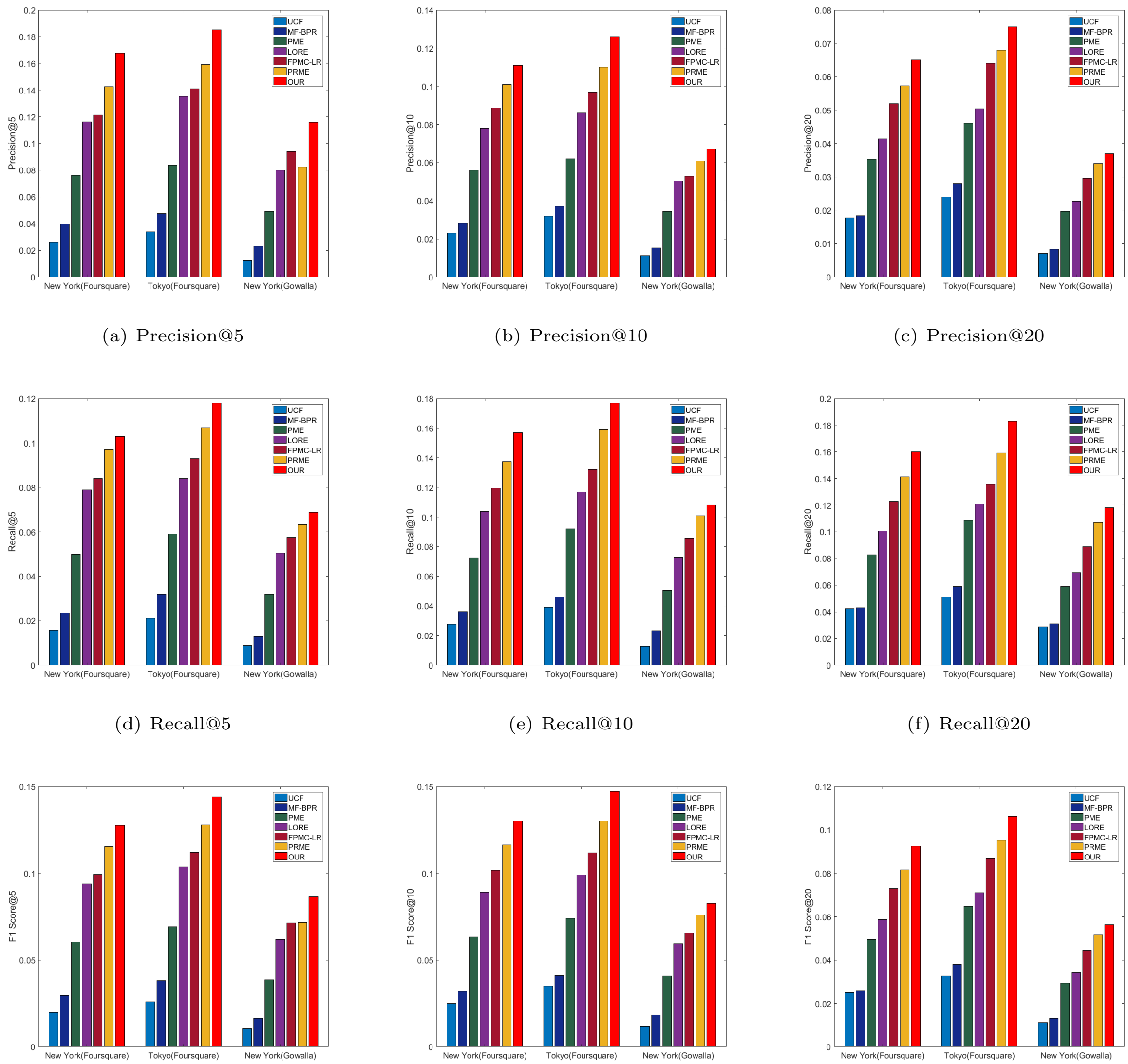
Figure 7: The validity of our user similarity

5.1.4 Comparative Approaches. We use six representative POI recommendation approaches as baseline methods for next new POI recommendation. The six comparative approaches are shown as follows:

- UCF [26]: User-based collaborative filtering.
- MF-BPR [18]: Matrix Factorization with Bayesian personalized ranking from implicit feedback.
- PME [21]: The personalized metric embedding, which projects users and POIs in a common latent space. We use this method to implement next new POI recommendation as a baseline method.
- LORE [33]: LORE predicts the probability of a user visiting a location by Additive Markov Chain (AMC) with Location-Location Transition Graph.
- FPMC-LR [5]: A personalized next POI recommendation algorithm of embedding the personalized Markov chains and the localized regions.
- PRME [11]: A personalized ranking metric embedding method for next new POI recommendation.

5.2 Recommendation Effectiveness

5.2.1 The Effectiveness of User Similarity. We designed a novel personalized user similarity calculation method that considers the three factors including sequential property, common access property and visited popularity of virtual common



(a) Precision@5 (b) Precision@10 (c) Precision@20

(d) Recall@5 (e) Recall@10 (f) Recall@20

(g) F1 Score@5 (h) F1 Score@10 (i) F1 Score@20

Figure 8: Comparison with baselines under different datasets

access sequence. We use UCF to verify the validity of our user similarity calculation method. The recommendation results are compared by using cosine similarity and our similarity in UCF, respectively. Figure 7 shows that the Precision of UCF using our user similarity in two datasets is significantly higher than UCF using cosine similarity. This is because our user similarity has directionality and considers multiple factors such as preference, time, geography and sequence that better reflect the relationship between users.

5.2.2 Comparison of Methods. We compared the Precision ,Recall and F1 Score of our approach with baseline methods in two datasets. Through repeated experiments, we finally determined that the optimal value of α is 3. Figure 8 shows the comparison results when the time threshold in definition 3 is 6 hours. It is important to note that our approach's Precision and Recall are relatively low for three reasons. The first reason is compared to general POI recommendation, next new POI recommendation is a more complex and difficult task, and it faced even more difficult challenges. The second reason is that our ground truth is looking for new POIs that the user has visited within 6 hours after each check-in. New POIs refer to the POIs that the user has not visited before this check-in. Therefore, our ground truth is more stringent.

Table 2: The effect of time threshold

Time Threshold (hour)	Pre@5	Pre@10	Pre@20	Pre@5	Pre@10	Pre@20	Time Threshold (hours)	Pre@5	Pre@10	Pre@20
	New York (Foursquare)			Tokyo (Foursquare)				New York (Gowalla)		
2	0.106	0.066	0.037	0.113	0.075	0.046	1	0.058	0.031	0.017
4	0.112	0.073	0.042	0.129	0.088	0.053	**5**	**0.116**	**0.066**	**0.036**
6	**0.168**	**0.111**	**0.065**	**0.185**	**0.126**	**0.075**	10	0.067	0.041	0.023
8	0.134	0.089	0.052	0.155	0.104	0.064	15	0.063	0.039	0.020
10	0.072	0.048	0.028	0.083	0.061	0.039	20	0.060	0.033	0.018
∞	0.061	0.041	0.025	0.074	0.052	0.031	∞	0.055	0.029	0.014

The third reason is that user check-in history data in two datasets is extremely sparse.

The experimental results show that the performances of UCF and MF-BPR are extremely poor. The main reason is that these two approaches are more focused on mining users preference information, and do not effectively utilize the geographical influence and sequential influence of users' check-in behavior. Besides, UCF's performance is the worst among all the approaches because it suffers other problems except the issues described above. One problem is that cosine similarity can not accurately reflect the relationship between users, and another problem is that collaborative filtering is easy to be disturbed by data sparsity. The PME results are not acceptable because the learning of sequential transition and user preference would be interfered with each other in a common latent space.

Both LORE and FPMC-LR have relatively good performance. LORE predicts the probability of a user visiting a location by Additive Markov Chain (AMC). LORE fuses sequential influence with geographical influence and social influence into a unified recommendation framework. FPMC-LR not only exploits the personalized Markov chain in the check-in sequence, but also takes into account users' movement constraint, i.e., moving around a localized region. The commonality between LORE and FPMC-LR is that they all effectively take advantage of the sequential influence of users' check-in behavior and both use Markov chains. This implies that the use of sequential influence is very important for the next new POI recommendation, and also illustrates the validity of the Markov chain. FPMC-LR is superior to LORE. The reason is that the use of localized regions information can not only greatly reduce the computational cost, but also discard the noise information to improve the recommendation. PRME performed very well. This is because PRME uses two latent spaces to embed user preferences and sequential patterns, respectively. PRME exploits the metric embedding method for the recommendation, which avoids drawbacks of the Matrix Factorization technique.

The performance of our approach in Precision, Recall and comprehensive evaluation metric F1 Score is always better than the baseline methods mainly for the following reasons: Our recommendation process simulates user travel decision-making process. Our approach integrate user preferences, sequential influence and geographical influence into a unified recommendation process. And our recommendation system can dynamically predict the user's preferences based on the change of time and weigh two factors of preference factors and geographic factors when users decide destinations. Furthermore, we designed a novel personalized user similarity calculation method and built a curve for each single user to quantitatively measure the probability of POI visiting w.r.t. the travel distance in different time slots.

5.2.3 Impact of Time Threshold. Table 2 shows the impact of time threshold in definition 3 on our approach in two datasets. We observe that as the time threshold increases, the performance of our approach first increases rapidly and then gradually decreases. This is because the continual check-in sequences generated when the time threshold is too small are short in length and small in number, which is not enough to reflect users' travel information and similar relationships between users. And the continual check-in sequences generated when the time threshold is too large mask the travel information of users at the next time interval. Moreover, our approach performed best with a time threshold of 6 hours in Foursquare and a time threshold of 5 hours in Gowalla. This indicates that the appropriate time threshold can well reflect the user's travel information and the similar relationship between users and the sequential influence of users' check-in behavior in these two datasets plays the most important role when the time threshold is 6 hours and 5 hours respectively.

6 CONCLUSIONS

In this paper, we design a novel and effective next new POI recommendation system by simulating user travel decision-making process. Our system considers two important factors that influence people's choice of travel destination: preference factors and geographic factors, and integrates them into a unified recommendation process.

There are still some directions worth exploring and improving in the future. For example, how to better utilize heterogeneous information to construct a more accurate user similarity calculation method. And how to more accurately predict the change of users' preferences in different time slots is a very difficult and valuable research direction. These are the focus of our future research.

REFERENCES

[1] Hakan Bagci and Pinar Karagoz. 2016. Context-Aware Friend Recommendation for Location Based Social Networks using Random Walk. In *International Conference Companion on World Wide Web*. 531–536.

[2] Betim Berjani and Thorsten Strufe. 2011. A recommendation system for spots in location-based online social networks. In *The Workshop on Social Network Systems*. 4.

[3] Xin Cao, Gao Cong, and Christian S Jensen. 2010. Mining significant semantic locations from GPS data. *Proceedings of the Vldb Endowment* 3, 1 (2010), 1009–1020.

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

[5] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In *International Joint Conference on Artificial Intelligence*.

[6] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility:user movement in location-based social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, Usa, August*. 1082–1090.

[7] D. Comaniciu and P. Meer. 2002. Mean Shift: A Robust Approach Toward Feature Space Analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 603–619.

[8] Chaoran Cui, Jialie Shen, Liqiang Nie, Richang Hong, and Jun Ma. 2017. Augmented Collaborative Filtering for Sparseness Reduction in Personalized POI Recommendation. *Acm Transactions on Intelligent Systems and Technology* 8, 5 (2017), 71.

[9] Lin Cui, Pi Dechang, and Li Hui. 2016. Periodic and successive point-of-interest recommendation under dual social group influences with matrix factorization. *Revista de la Facultad de Ingenieria* (2016), 255–269.

[10] Madhuri Debnath, Praveen Kumar, and Ramez Elmasri. 2016. Preference-aware POI recommendation with temporal and spatial influence. In *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016*. 548–553.

[11] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *International Conference on Artificial Intelligence*. 2069–2075.

[12] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *ACM Conference on Recommender Systems*. 93–100.

[13] Jing He, Xin Li, Lejian Liao, Dandan Song, and William K. Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *Thirtieth AAAI Conference on Artificial Intelligence*. 137–143.

[14] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2013. Mining User Mobility Features for Next Place Prediction in Location-Based Services. In *IEEE International Conference on Data Mining*. 1038–1043.

[15] Shokirkhon Oppokhonov, Seyoung Park, and Isaac K. E Ampomah. 2017. Current location-based next POI recommendation. In *the International Conference*. 831–836.

[16] Makbule Gulcin Ozsoy, Faruk Polat, and Reda Alhajj. 2016. Time preference aware dynamic recommendation enhanced with location, social network and temporal information. In *Ieee/acm International Conference on Advances in Social Networks Analysis and Mining*. 909–916.

[17] Xing Yi Ren, Mei Na Song, and Jun De Song. 2017. Context-Aware Point-of-Interest Recommendation in Location-Based Social Networks. *Chinese Journal of Computers* (2017).

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. (2012), 452–461.

[19] Zhao Shenglin, Zhao Tong, Yang Haiqin, R. Lyu Michael, and King Irwin. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. 315–321.

[20] Yali Si, Fuzhi Zhang, and Wenyuan Liu. 2017. CTF-ARA: An adaptive method for POI recommendation based on check-in and temporal features. *Knowledge-Based Systems* 128 (2017).

[21] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. In *ACM Conference on Recommender Systems*. 137–140.

[22] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2014. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems Man and Cybernetics Systems* 45, 1 (2014), 129–142.

[23] Mao Ye, Peifeng Yin, and Wang Chien Lee. 2010. Location recommendation for location-based social networks. In *ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Acm-Gis 2010, November 3-5, 2010, San Jose, Ca, Usa, Proceedings*. 458–461.

[24] Mao Ye, Peifeng Yin, Wang Chien Lee, and Dik Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *International Acm Sigir Conference on Research and Development in Information Retrieval*. 325–334.

[25] Yonghong Yu, Gao Yang, and Wang Hao. 2016. A Ranking Based Poisson Matrix Factorization Model for Point-of-Interest Recommendation. *Journal of Computer Research and Development* (2016).

[26] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 363–372.

[27] Zhengwu Yuan and Haiguang Li. 2016. Location recommendation algorithm based on temporal and geographical similarity in location-based social networks. In *Intelligent Control and Automation*. 1697–1702.

[28] Jia Dong Zhang and Chi Yin Chow. 2013. iGSLR: personalized geo-social location recommendation:a kernel density estimation approach. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems*. 334–343.

[29] Jia Dong Zhang and Chi Yin Chow. 2015. CoRe: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations. *Information Sciences* 293 (2015), 163–181.

[30] Jia Dong Zhang and Chi Yin Chow. 2015. GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 443–452.

[31] Jia Dong Zhang and Chi Yin Chow. 2015. Spatiotemporal Sequential Influence Modeling for Location Recommendations: A Gravity-based Approach. *Acm Transactions on Intelligent Systems and Technology* 7, 1 (2015), 1–25.

[32] Jia Dong Zhang and Chi Yin Chow. 2017. TICRec: A Probabilistic Framework to Utilize Temporal Influence Correlations for Time-aware Location Recommendations. *IEEE Transactions on Services Computing* 9, 4 (2017), 633–646.

[33] Jia Dong Zhang, Yanhua Li, and Yanhua Li. 2014. LORE: exploiting sequential influence for location recommendations. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems*. 103–112.

[34] Zheng, Yu, Zhang, Lizhu, Xie, Xing, Ma, and WeiYing. 2009. Mining interesting locations and travel sequences from GPS trajectories. *Proceedings of Www* (2009), 791–800.

[35] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, Usa, April*. 1029–1038.