A Time-Aware Personalized Point-of-Interest Recommendation via High-Order Tensor Factorization

XIN LI, MINGMING JIANG, HUITING HONG, and LEJIAN LIAO, School of Computer Science, Beijing Institute of Technology

Recently, location-based services (LBSs) have been increasingly popular for people to experience new possibilities, for example, personalized point-of-interest (POI) recommendations that leverage on the overlapping of user trajectories to recommend POI collaboratively. POI recommendation is yet challenging as it suffers from the problems known for the conventional recommendation tasks such as data sparsity and cold start, and to a much greater extent. In the literature, most of the related works apply collaborate filtering to POI recommendation while overlooking the personalized time-variant human behavioral tendency. In this article, we put forward a fourth-order tensor factorization-based ranking methodology to recommend users their interested locations by considering their time-varying behavioral trends while capturing their long-term preferences and short-term preferences simultaneously. We also propose to categorize the locations to alleviate data sparsity and cold-start issues, and accordingly new POIs that users have not visited can thus be bubbled up during the category ranking process. The tensor factorization is carefully studied to prune the irrelevant factors to the ranking results to achieve efficient POI recommendations. The experimental results validate the efficacy of our proposed mechanism, which outperforms the state-of-the-art approaches significantly.

CCS Concepts: • **Information systems** → **Collaborative filtering**; • **Applied computing** → *Sociology*

Additional Key Words and Phrases: Time-aware POI recommendation, tensor factorization, HITS algorithm

ACM Reference Format:
Xin Li, Mingming Jiang, Huiting Hong, and Lejian Liao. 2017. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. ACM Trans. Inf. Syst. 35, 4, Article 31 (June 2017), 23 pages.
DOI: http://dx.doi.org/10.1145/3057283

1. INTRODUCTION

With the widespread use of mobile devices such as global positioning system (GPS)-enabled smartphones, various types of location-based services (LBSs) have become very popular. And accessing these location-based social networks (LBSNs) has become an important part of modern social life [Chow et al. 2010; Yang et al. 2012]. Many LBSN systems, for example, Foursquare,[1] Yelp,[2] and Gowalla,[3] have emerged to help users build connections with their friends, where the users can upload photos, comment, and share their points of interest (POIs) via online check-ins, such as restaurants, cinema,

[1]https://foursquare.com.
[2]https://www.yelp.com.
[3]It was acquired by Facebook on December 2, 2011.

Authors' addresses: X. Li, M. Jiang, H. Hong, and L. Liao, BJ ER Center of HVLIP, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; emails: xinli@bit.edu.cn, jiangmings1992@163.com, xhhszc@163.com, liaolj@bit.edu.cn.

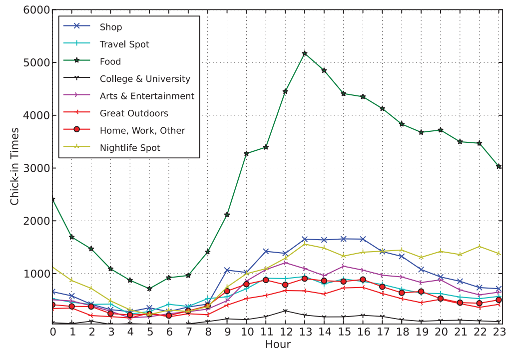
DOI: http://dx.doi.org/10.1145/3057283

tourist spots, and so forth. The major difference from conventional social networks lies in that what users shared via LBSNs is now all geographically related one way or another. With a large volume of data of geographic relations, many tasks could become possible, for example, mapping, route finding, and POI recommendations.

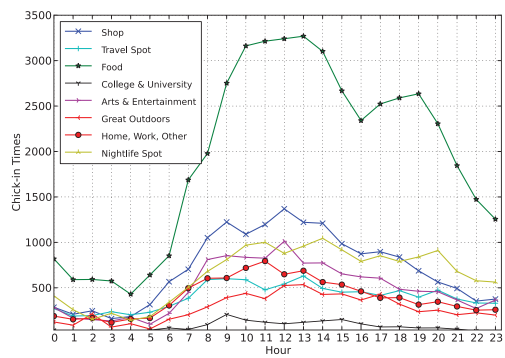
In the literature, location-aware data mining methods utilizing users' digital trajectories have been proposed to discover interesting locations and travel sequences [Zheng et al. 2009]. The trajectory could be predicted based on the extensive WiFi mobility data [Song et al. 2004]. Differing from most of the mobility data, such as GPS/WiFi trajectories, which are collected by the system automatically, the sparseness of the LBSN data is due to the user's voluntary check-in behavior. It is very common in LBSNs that the users just do not update their status while they are on the spot. Such sparseness makes the POI recommendation very challenging. POI recommendation is a very important task in LBSNs to recommend to users the location they are potentially interested in. An effective POI recommendation service can not only improve user viscosity by enhancing the user experience on LBSNs but also benefit advertising agencies in that they can provide an easy way of launching advertisements to target potential clients. Intuitively, POI recommendation is based on mining the individual preference or past behavior. In fact, such data is relatively sparse due to the check-in on a voluntary basis. Furthermore, some successive check-in records span a long period, which poses another challenge for POI recommendation.

Most of the existing recommendation systems (e.g., music recommendation, movie recommendation, and friends recommendation) recommend items that users might be interested in (or likely to listen to, watch, or accept) at a later time. The spanned time could take days, weeks, and even months. However, an ideal POI recommender system must have high standards in terms of real-time response. For example, the system may recommend hotels instead of nightspots or movies if a user just gets off an airplane, even though he or she may enjoy nightlife and watch movies very often. And such a POI recommendation will soon expire when the user arrives at the hotel. Then the nightspots may be among the recommended locations when it is getting late into the evening.

POI recommendation as a special form of item recommendation [Su et al. 2010; Rendle et al. 2010; Rendle and Schmidt-Thieme 2010] has attracted much attention from both academia and industry recently. The amount of data with social structures (e.g., user's residence, friendship, and their comments on the check-in location) has been explored as the contextual cues to fulfill the recommendation task. Conventional recommendation techniques such as collaborative filtering (CF) [Linden et al. 2003; Sarwar et al. 2001] and matrix factorization (MF) [Koren et al. 2009; Cheng et al. 2012] have been adopted for POI recommendation by incorporating the contextual information (e.g., sentimental analysis of comments, limits of distance between the two successive checked-in locations given the spanned time). Ye et al. [2013] proposed to utilize the first-order hidden Markov model (HMM) to capture users' short-term preferences for location prediction. FPMC-LR [Cheng et al. 2013] is the first work to apply factorized personalized Markov chain (FPMC) [Rendle et al. 2010] (proposed for bucket-list recommendations) to the POI recommendation task. Specifically, FPMC-LR considered the next POI recommendation task by taking into account the temporal information and utilized the distance constraint to shorten the candidate list of locations, and then to reduce the computational complexity. Although the authors claim that the temporal relation of locations has been considered, the relations are in fact extracted via a Markov chain that only captures the consecutive ordering relations instead of the sophisticated behavior tendency over time explicitly. We argue that the temporal relation that spans a long time is also an important feature for POI recommendations. For example, some people enjoy a morning run in the park, checking in at various locations



(a) Category Check-in Patterns of NYC



(b) Categorical Check-in Patterns of LA

Fig. 1. Relationship between time interval and check-in location category.

during the daytime, and going to church every Sunday, which spans a relatively longer period of time but forms a regular periodic pattern.

Before we go into the details of our proposed framework for next POI recommendations, here we discuss the unique aspects of POI recommendation compared to the conventional recommendation as follows:

—The types of the checked-in venues are highly related to the time period, as shown in Figure 1, which presents the categorical check-ins of New York City and Los Angeles in Foursquare, respectively (the detailed data description can be seen in Section 4.1). It is obvious that the number of checked-in locations of different types over time has different patterns. For example, the number of check-ins for "Shop" during daytime hours is much higher than that of the evening. The peaks of check-ins for "Food" spots occur at 13:00 p.m. and 19:00 p.m., respectively. In general, the observation of online check-in behavior is consistent with our daily habits.

—The number of venues is large (about several thousands of POIs per square kilometer) within an average range of the user's daily activities, which makes the recommendation time-consuming and accuracy difficult to achieve. In contrast, the user's check-in data is very sparse as the user checked in online on a "voluntary" basis. And the check-in data can only provide the implicit preferences by indicating whether some POIs have been visited by the users.

Table I. Novelty of User Access Location

City	Total Number of Locations	Number of New Locations	Novelty of the Accessed Location
New York	30,459	27,731	91.04%
Los Angeles	20,125	16,515	82.06%

—Human activity is restricted within a local area due to urban transportation. Moreover, people tend to visit nearby locations rather than those in distant areas, as explained in Tobler's first law of geography: "everything is related to everything else, but near things are more related than distant things" [Tobler 1970].
—The transition between POIs is strongly affected by the user's own preference, which is referred to as a long-term individual preference in our article. For example, some prefer to go home right after work, while others are more likely to go to restaurants after work. Thus, capturing users' personalized preferences is crucial for the POI recommendation task.
—Users' next location is highly related to their current location, which is referred to as short-term preference in our article. For example, people that just get off the plane tend to go to hotels rather than theaters.
—People tend to explore new POIs they have not visited before. For the Foursquare datasets (NYC and LA), we partition the data with respect to the time series; the former 80% of data are taken as historical data, and the other 20% as the observation data. Then we define a metric $N_{ob}^{nv}/N_{hist}^{vb} - N_{ob}^{vb}$ to measure the "newness" of the accessed locations to the users, where N_{ob}^{nv} is the number of new locations in the observation dataset. N_{hist}^{vb} is the number of visited locations in the historical data. N_{ob}^{vb} is the number of locations that have been visited in both the observation data and the historical data. We list out the corresponding statistics for the two datasets in Table I. We observe that both the NYC dataset and LA dataset have a higher " newness." It indicates that the users often visit new places they have not been before. Thus, the "newness" of the visited location to the users shall be taken into the consideration for a reliable POI recommendation, which in turn helps to enhance user experience and to increase the commercial brand awareness for advertising. Intuitively, recommending new places to a user according to his or her current categorical interest is a good strategy (e.g., recommending newly opened restaurants to users during lunch time).

Toward an effective POI recommendation, we consider all the aforementioned characteristics and propose a two-step personalized POI recommendation methodology. The system is supposed to recommend different places of interest to users with changes of time and current location. In the first step, we propose an adaptive pairwise interaction tensor factorization (PITF) [Rendle and Schmidt-Thieme 2010] to predict the category list of the next location by considering users' time-varying behavioral trends. The adoption of categorical information can greatly alleviate the sparsity and cold-start issues. A fourth-order tensor factorization is proposed to integrate users' short-term influence, long-term influence, and time-variant preference to effectively predict the category of the next location. The second step is to obtain the location ranking list according to the categorical ranking results from the first step. A metric-based algorithm is proposed to do so. We also propose a bipartite graph-based algorithm for location ranking. Both of them consider the principle of spacial locality mentioned before. The final location ranking list will be obtained based on the recommended categories and our proposed category-to-location algorithms. We conducted extensive experiments to demonstrate the effectiveness of our proposed methodology. Our experimental results show that the methodology we proposed outperforms the state-of-the-art models by a large margin.

The key contributions of the work presented in this article are the following:

—We analyzed the spatial and temporal characteristics for the POI recommendation problem and propose a two-step methodology accordingly with higher prediction accuracy obtained. First, we predict users' preferred categories. Second, we manage to recommend to users the most interesting locations according to the predicted categories.
—A fourth-order tensor model for the category prediction is proposed to capture users' personalized long-term preference, short-term preference, and time-varying behaviors simultaneously, making our approach more personalized and time aware.
—We incorporate a time-decay factor into the tensor-based model to simulate the discontinuous check-in behavior and weigh the importance of the spanning time between two successive check-ins to the POI recommendation. We also show that the time-delay factor has played an important role for POI recommendation.
—The complexity has been sharply reduced due to the fact that the dimensionality of the problem has been reduced from the number of locations to the number of categories. Besides, with the category recommendation incorporated, the system is able to recommend more new venues and thus provide the user a better experience and be useful for advertisers to promote their services.

The rest of the article is organized as follows. In Section 2, we review the work related to POI recommendations. The problem formulation and the proposed methodology are introduced in detail in Section 3. In Section 4, we evaluate the proposed method with comparative experiments on real LBSN data and analyze the performances. Finally, we conclude the article in Section 5.

2. RELATED WORK

POI recommendations based on LBSN data share some common characteristics with related problems like trajectory-based location prediction and item recommendation.

The widely used GPS-enabled devices bring us a large amount of GPS trajectories representing people's location histories. Different location prediction methods based on GPS trajectories have been proposed in the literature. A HITS (Hypertext Induced Topic Search)-based inference model [Zheng et al. 2009] regards an individual's access to a location as a directed link from the user to the location for mining interesting locations and travel sequences from the GPS trajectories. A novel notion of individual life pattern was proposed in Ye et al. [2009], which captures the individual's lifestyle and regularity. They first introduce the life-pattern-normal form (LP-normal form) to formalize the expression of life patterns and then propose the LP-Mine framework to effectively extract life patterns from the raw GPS data. Compared with GPS data, LBSN data is different as LBSN carries more social information (i.e., check-in tips, residence, and comments on particular locations). However, with the rich information of the LBSN social structure, LBSN data is still considered very sparse as the check-in data is updated by the users themselves, unlike the GPS trajectory data, which is collected by the system automatically. The physical distance between two successive check-in records is typically much larger than that of GPS data, which makes POI recommendation more challenging.

Collaborative filtering (CF) [Sarwar et al. 2001] has been adopted for POI recommendation to deal with the user-location check-in matrix by exploring either the user similarity or the location similarity [Chow et al. 2010]. Zheng et al. [2010] proposed to utilize the collective matrix factorization method to mine the interesting locations and activities. The user-related data are pulled together to apply collaborative filtering to find like-minded users and like-patterned activities at different locations. The idea is to utilize the CF model to perceive the similarity between users or between locations, and then to recommend the user with the locations visited by his or her peers. However, such conventional CF-based strategies cannot effectively deal with the cold-start issue (e.g., recommending new POIs to the users).

Some of the existing approaches utilize user profile to exploit user preference settings for POI recommendation. Bayesian network is used to evaluate the matching between user profiles (area, age, income, etc.) and restaurant profiles (price, cuisine, etc.), and then the system recommends restaurants with higher matching scores to the user [Park et al. 2007]. Recommendations based on user profile can be a good solution for cold start. But due to the lack of flexibility, it also sacrifices the accuracy of the recommendation.

Recently many efforts have been put toward POI recommendation by considering more factors, for example, geographical factors, temporal factors, sentiment factors, and social factors. Ye et al. [2011] exploit a power-law distribution to model users' check-in behavior and incorporate such geographical influence for POI recommendation. Zhang and Chow [2016] modeled the geographical influence with a kernel density estimation. FPMC-LR [Cheng et al. 2013] takes into account the geographical constraints of users' movement to shorten the candidate list of users' interested locations.

Recent studies also proposed to analyze the published location-related tweets or users' comments on POIs to facilitate the POI recommendation. Chen et al. [2013b] built a detection model to mine the user interest from short text and established the mapping between location function and user interest. Gao et al. [2015] studied both POI-associated content and users' sentiment information into POI recommendation and reported their enhanced performance. However, semantic analysis itself is a very challenging research issue as most of the comments in LBSN are short and contextually ambiguous.

Some works mainly leverage the temporal influence to enhance the recommendation performance. Yuan et al. [2013] proposed to extend the user-based POI recommendation by incorporating the time factor when computing the similarity between two users in terms of the historical check-ins at time t. Gao et al. [2013] investigated the temporal cyclic patterns of user check-ins in terms of temporal nonuniformness and temporal consecutiveness. Chen et al. [2013a] designed a temporal recommender system and modeled the user behavior based on the intrinsic interests as well as the temporal context. Yin et al. [2013] proposed a temporal context-aware mixture model to model user rating behaviors with the topic model.

Existing studies also showed the importance of social influence in POI recommendation, which can somehow enhance the quality of recommendation for long-distance POI recommendation [Cho et al. 2011]. Cheng et al. [2012] modeled the latent preferences of users and the features of POI by probabilistic matrix factorization with social regularization, where the geographical influence is addressed by utilizing a Multicenter Gaussian Model.

However, the previous POI recommendations all address the problem as the conventional item recommendation with a geographical feature. Recently, the task of next POI recommendation (also known as successive POI recommendation) was proposed, which only recommends the POIs at the successive timestamp. The task is more challenging as the successive POI check-in data is even more sparse. Cheng et al. [2013] first formally define the problem of next personalized POI recommendation, where they conduct a FPMC [Rendle et al. 2010] over a check-in tensor with the movement constraint considered. Zhao et al. [2016] proposed a spatial-temporal latent ranking method to recommend to users the most possible successive POIs by capturing the impact of time.

However, the previous recommendations are all different from our work. This articles focuses on predicting users' next POI categories by considering all the unique features discussed in Section 1 and then predicting locations under the category's distribution. The extensive experiments on two real-world datasets demonstrate that our proposed two-step methodology outperforms the state-of-the-art successive POI recommendation model by a large margin.

3. PROBLEM FORMULATION

In this section, we present the details of our two-step approach: Category Prediction based on Tensor Factorization and Location Recommendation with proposed metrics. We start with the notations and symbols used throughout the article.

3.1. Notation

Let u_i denote the ith user and $\mathcal{U} = \{u_1, u_2, u_3, \ldots, u_{|U|}\}$ denote the set of LBSN users. $\mathcal{L} = \{l_1, l_2, \ldots, l_{|\mathcal{L}|}\}$ denotes the set of locations and $\mathcal{C}$ denotes the set of location categories (e.g., school, restaurant, shopping mall, etc.). We allow one location to possibly belong to multiple categories. $\mathcal{C}_{(l_t)} \subset \mathcal{C}$ and $\mathcal{C}_{(l_{t+1})} \subset \mathcal{C}$ represent the category set of the current location $l_t \in \mathcal{L}$ and the next location $l_{t+1} \in \mathcal{L}$, respectively. $c_{(l_t)} \in \mathcal{C}_{(l_t)}$ and $c_{(l_{t+1})} \in \mathcal{C}_{(l_{t+1})}$ denote the category of the current location and the category of the next location, respectively. To explore the temporal information, we roughly divide 24 hours of a day into five time intervals according to the observation from Figure 1, where there are roughly five patterns in one day; namely, $\mathcal{T} = \{T_1, T_2, T_3, T_4, T_5\}$. $T_1 = \{2, 3, 4, 5\}$, $T_2 = \{6, 7, 8, 9\}$, $T_3 = \{10, 11, 12, 13, 14\}$, $T_4 = \{15, 16, 17, 18\}$, and $T_5 = \{19, 20, 21, 22, 23, 0, 1\}$ represent dawn, morning, noon, afternoon, and evening, respectively. Δt indicates the spanned time between two successive check-ins (check-in timestamp of the current location and that of the next location).

3.2. Time-Aware FPMC for Category Prediction

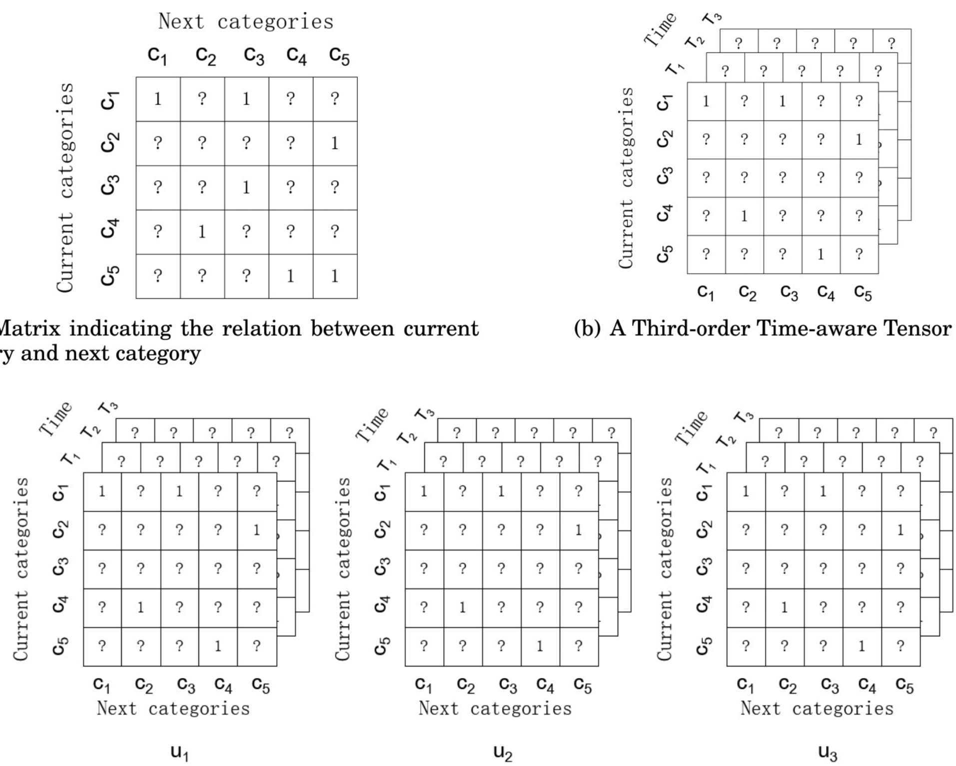
In this section, we will introduce our proposed time-aware FPMC for category prediction in detail, which will be referred to as TA-FPMC hereafter.

3.2.1. TA-FPMC. In our TA-FPMC, we consider only the transitions among the categories of check-ins over time. All categories checked-in in sequence are considered for each user and yield a fourth-order transition tensor $\chi \in [0, 1]^{|\mathcal{U}|\times|\mathcal{T}|\times|\mathcal{C}|\times|\mathcal{C}|}$ (Figure 2).

A first-order Markov chain is used here to indicate the impact from the current category to the next category. Compared with FPMC and FPMC-LR, our TA-FPMC can largely save the time cost due to the reduced dimensionality from the number of locations to the number of categories. We assume that the probability of checking in to a category is the average of the probabilities from each belonging category of the current location. Then the first-order transition probability between the current categories and the next categories can be formalized as

$$P(c_{(l_{t+1})}|\mathcal{C}_{(l_t)}) = \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} P\big(c_{(l_{t+1})}|c_{(l_t)}\big), \tag{1}$$

where $P(c_{(l_{t+1})}|c_{(l_t)})$ corresponds to the probability $p_{u_a,T_b,c_{(l_t)},c_{(l_{t+1})}}$ of user u_a transferring from the current category $c_{(l_t)}$ to the next category $c_{(l_{t+1})}$ at the time interval T_b. Figure 2(a) gives a graphic illustration of the transitions between the two successively checked-in location categories. The transition probability between two categories is labeled as "1" if we observe a transition between these two categories, and otherwise labeled as "?", which yields a transition matrix. Our objective is to somehow utilize Equation (1) to train a model from the observed transitions and then estimate the transition probabilities between the unobserved category pairs.



(a) A Matrix indicating the relation between current category and next category

(b) A Third-order Time-aware Tensor

(c) A Fourth-order Personalized Time-aware Tensor

Fig. 2. A fourth-order tensor construction.

As seen from Figure 1, the visiting trends of location categories show different patterns over time. We argue that the transition probability between the categories is also highly related to the specific time period. In our experiments, we divided the whole day into five different time periods to characterize the pattern. We integrate the temporal information into the previous matrix (Figure 2(a)) to form a third-order tensor (Figure 2(b)).

All users have their own preferences, which leads to different patterns of daily activities. Thus, we further extend the previous third-order tensor by incorporating the personalized preference. Figure 2(c) shows an example of our finally achieved fourth-order tensor.

There are many approximate approaches to recover a tensor with missing values. Tucker Decomposition (TD) and Canonical Decomposition (CD) [Carroll and Chang 1970] are the two common forms of tensor factorization. Here we adopt a special case of Tucker Decomposition, PITF, to model the pairwise interactions among the four modes of the tensor, that is, user, time interval, current category, and next category, given as

$$p_{u_a,T_b,c_{(l_t)},c_{(l_{t+1})}} = \vec{u_a}^{\mathcal{T}} \cdot \vec{T_b}^{\mathcal{U}} + \vec{u_a}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{U}} + \vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}} + \vec{T_b}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{T}} + \vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}} + \vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}. \quad (2)$$

—For the interaction between user and time period: $\vec{u_a}^{\mathcal{T}}$ modeling the user features and $\vec{T_b}^{\mathcal{U}}$ for the time period

—For the interaction between user and current location's categories: $\vec{u_a}^{\mathcal{C}_{(l_t)}}$ modeling the user features and $\vec{c_{(l_t)}}^{\mathcal{U}}$ for current location's categories
—For the interaction between user and next location's categories: $\vec{u_a}^{\mathcal{C}_{(l_{t+1})}}$ modeling the user features and $\vec{c_{(l_{t+1})}}^{\mathcal{U}}$ for current location's categories
—For the interaction between time period and current location's categories: $\vec{T_b}^{\mathcal{C}_{(l_t)}}$ modeling the time period features and $\vec{c_{(l_t)}}^{\mathcal{T}}$ for next location's categories
—For the interaction between time period and next location's categories: $\vec{T_b}^{\mathcal{C}_{(l_{t+1})}}$ modeling the time period features and $\vec{c_{(l_{t+1})}}^{\mathcal{T}}$ for next location's categories
—For the interaction between current location's categories and next location's categories: $\vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}}$ modeling the current location's categories features and $\vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}$ for next location's categories

Equation (1) is then further revised with the personalized preference and temporal information incorporated into the first-order Markov chain model. And we substitute $p(c_{t+1}|c_i \in C_t)$ with Equation (2) and obtain

$$\begin{aligned} P(c_{(l_{t+1})}|\mathcal{C}_{(l_t)}) &= \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} p_{u_a, T_b, c_{(l_t)}, c_{(l_{t+1})}} \\ &= \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} (\vec{u_a}^{\mathcal{T}} \cdot \vec{T_b}^{\mathcal{U}} + \vec{u_a}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{U}} + \vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}} \\ &\quad + \vec{T_b}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{T}} + \vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}} + \vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}). \end{aligned} \tag{3}$$

As the pairwise interactions of $\vec{u_a}^{\mathcal{T}} \cdot \vec{T_b}^{\mathcal{U}}$, $\vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}}$, and $\vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}}$ are independent of $c_{(l_t)}$, Equation (3) can be rewritten as

$$\begin{aligned} P(c_{(l_{t+1})}|\mathcal{C}_{(l_t)}) &= \vec{u_a}^{\mathcal{T}} \cdot \vec{T_b}^{\mathcal{U}} + \vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}} + \vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}} \\ &\quad + \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} (\vec{u_a}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{U}} + \vec{T_b}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{T}} + \vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}). \end{aligned} \tag{4}$$

3.2.2. BPR for TA-FPMC. Due to the sparsity of the dataset, we adopt a ranking approach as suggested in Rendle et al. [2009] to achieve the top-k list of categories that the user is most likely to visit. In dealing with the learning problem, here we adapt Bayesian Personalized Ranking (BPR) for ranking $>_{u,T,c_t}$ over all categories:

$$c_i >_{u_a, T_b, \mathcal{C}_{(l_t)}} c_j \Longleftrightarrow P(c_{(l_{t+1})} = c_i|\mathcal{C}_{(l_t)}) > P(c_{(l_{t+1})} = c_j|\mathcal{C}_{(l_t)}). \tag{5}$$

The problem of finding the best top-k ranking $>_{u,T,c_t}$ can be formalized as a problem maximizing the following posterior:

$$P(\Theta| >_{u,T,\mathcal{C}_{(l_t)}}) \propto P(>_{u,T,\mathcal{C}_{(l_t)}} |\Theta)P(\Theta), \tag{6}$$

where $P(>_{u,T,\mathcal{C}_{(l_t)}} |\Theta)$ denotes the likelihood function, $P(\Theta)$ is the prior probability, and Θ represents the model parameters. We also assume that users have independent options for location selection. The likelihood $P(>_{u,T,\mathcal{C}_{(l_t)}} |\Theta)$ is formalized by the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$. By maximizing the posterior, we estimate the model parameters, given as

$$\begin{aligned}
&\arg\max_{\Theta} \prod_{(u,T,\mathcal{C}_{(l_t)},c_i)\in\mathbb{S},(u,T,\mathcal{C}_{(l_t)},c_j)\notin\mathbb{S}} P(>_{u,T,\mathcal{C}_{(l_t)}} |\Theta)P(\Theta)\\
&= \arg\max_{\Theta} \prod_{(u,T,\mathcal{C}_{(l_t)},c_i)\in\mathbb{S},(u,T,\mathcal{C}_{(l_t)},c_j)\notin\mathbb{S}} P\left(P\left(c_{(l_{t+1})}=c_i|\mathcal{C}_{(l_t)}\right)-P\left(c_{(l_{t+1})}=c_j|\mathcal{C}_{(l_t)}\right)>0|\Theta\right)P\left(\Theta\right)\\
&= \arg\max_{\Theta} \prod_{(u,T,\mathcal{C}_{(l_t)},c_i)\in\mathbb{S},(u,T,\mathcal{C}_{(l_t)},c_j)\notin\mathbb{S}} \sigma\left(P\left(c_{(l_{t+1})}=c_i|\mathcal{C}_{(l_t)}\right)-P\left(c_{(l_{t+1})}=c_j|\mathcal{C}_{(l_t)}\right)\right)P\left(\Theta\right), \qquad (7)
\end{aligned}$$

where $\mathbb{S}$ denotes the sampling set. For the prior $P(\Theta)$, we assume that the model parameters are drawn from a Gaussian distribution $\Theta \sim \mathcal{N}(0, \sigma_\Theta \boldsymbol{I})$. Now we can rewrite Equation (7) to derive the optimization criterion for our model. Then an alternative maximum of a posteriori estimation in logarithmic scale is given as

$$\begin{aligned}
&\arg\max_{\Theta} \ln \prod_{(u,T,\mathcal{C}_{(l_t)},c_i)\in\mathbb{S},(u,T,\mathcal{C}_{(l_t)},c_j)\notin\mathbb{S}} P(>_{u,T,\mathcal{C}_{(l_t)}} |\Theta)P(\Theta)\\
&= \arg\max_{\Theta} \sum_{(u,T,\mathcal{C}_{(l_t)},c_i)\in\mathbb{S},(u,T,\mathcal{C}_{(l_t)},c_j)\notin\mathbb{S}} \ln\sigma\left(P\left(c_{(l_{t+1})}=c_i|\mathcal{C}_{(l_t)}\right)-P\left(c_{(l_{t+1})}=c_j|\mathcal{C}_{(l_t)}\right)\right)-\lambda_\Theta\|\Theta\|^2, \qquad (8)
\end{aligned}$$

where λ_Θ is the regularization constant corresponding to σ_Θ. By replacing $P(c_{(l_{t+1})} = c_i|\mathcal{C}_{(l_t)}) - P(c_{(l_{t+1})} = c_j|\mathcal{C}_{(l_t)})$ with Equation (4), $\vec{u_a}^{\mathcal{T}} \cdot \vec{T_b}^{\mathcal{U}}$, $\vec{u_a}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{U}}$ and $\vec{T_b}^{\mathcal{C}_{(l_t)}} \cdot \vec{c_{(l_t)}}^{\mathcal{T}}$ will vanish with no effect on the recommendation results. Then we update Equation (4) as follows:

$$\begin{aligned}
P(c_{(l_{t+1})}|\mathcal{C}_{(l_t)}) = \vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}} + \vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}}\\
+ \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)}\in\mathcal{C}_{(l_t)}} \left(\vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}\right). \qquad (9)
\end{aligned}$$

3.2.3. BPR for TAD-FPMC. The statistics from the LBSN data show that the two successive check-ins often span a large time gap. For example, if the two successive check-in records span a year, we hardly expect much can be learned from the last check-in. Thus, we assume that the intensity of the relation between the two successive check-in locations is decaying over time. And we introduce an attenuation factor $D(\Delta t)$ to our model to indicate the decay of the probability over time. Equation (9) can be rewritten as

$$\begin{aligned}
P(c_{(l_{t+1})}|\mathcal{C}_{(l_t)}) = \vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}} + \vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}}\\
+ \frac{1}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)}\in\mathcal{C}_{(l_t)}} D(\Delta t)\left(\vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}\right). \qquad (10)
\end{aligned}$$

It is worth noting that only $\vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}$ is related to the two successive check-ins. Both $\vec{u_a}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{U}}$ and $\vec{T_b}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{T}}$ have nothing to do with $D(\Delta t)$. Hence, we only insert an attenuation factor $D(\Delta t)$ to the part of $\vec{c_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}} \cdot \vec{c_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}$. The evolved Equation (10) formalizes the approach referred to as TAD-FPMC hereafter.

As the criterion is differentiable, similarly we optimize the model using BPR in which the bootstrapping-based stochastic gradient descent with learning rate *alpha* is adopted. Once the fourth-order tensor is recovered under the criterion, we achieve a

ALGORITHM 1: BPR Learning Algorithm for TAD-FPMC

Input: $\widehat{\mathcal{U}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{U}}, \widehat{\mathcal{T}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{T}}, \widehat{\mathcal{C}_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}$ from $\mathcal{N}(0, \sigma_\Theta \boldsymbol{I})$

Output: $\widehat{\mathcal{U}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{U}}, \widehat{\mathcal{T}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{T}}, \widehat{\mathcal{C}_{(l_t)}}^{\mathcal{C}_{(l_{t+1})}}, \widehat{\mathcal{C}_{(l_{t+1})}}^{\mathcal{C}_{(l_t)}}$

repeat

  draw $(u, T, \mathcal{C}_{(l_t)}, c_i, c_j)$ from $\mathbb{S}$;

  $\delta \leftarrow (1 - \sigma(P(c_{(l_{t+1})} = c_i|\mathcal{C}_{(l_t)}) - P(c_{(l_{t+1})} = c_j|\mathcal{C}_{(l_t)})))$;

  for $f = 1$ *to the number of features* **do**

   $u_f^{\mathcal{C}_{(l_{t+1})}} \leftarrow u_f^{\mathcal{C}_{(l_{t+1})}} + \alpha \left(\delta(c_{i,f}^{\mathcal{U}} - c_{j,f}^{\mathcal{U}}) - \lambda_\Theta u_f^{\mathcal{C}_{(l_{t+1})}}\right)$;

   $T_f^{\mathcal{C}_{(l_{t+1})}} \leftarrow T_f^{\mathcal{C}_{(l_{t+1})}} + \alpha \left(\delta(c_{i,f}^{\mathcal{T}} - c_{j,f}^{\mathcal{T}}) - \lambda_\Theta T_f^{\mathcal{C}_{(l_{t+1})}}\right)$;

   $c_{i,f}^{\mathcal{U}} \leftarrow c_{i,f}^{\mathcal{U}} + \alpha \left(\delta u_f^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{i,f}^{\mathcal{U}}\right)$;

   $c_{i,f}^{\mathcal{T}} \leftarrow c_{i,f}^{\mathcal{T}} + \alpha \left(\delta T_f^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{i,f}^{\mathcal{T}}\right)$;

   $c_{j,f}^{\mathcal{U}} \leftarrow c_{j,f}^{\mathcal{U}} + \alpha \left(-\delta u_f^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{j,f}^{\mathcal{U}}\right)$;

   $c_{j,f}^{\mathcal{T}} \leftarrow c_{j,f}^{\mathcal{T}} + \alpha \left(-\delta T_f^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{j,f}^{\mathcal{T}}\right)$;

   $c_{i,f}^{\mathcal{C}_{(l_t)}} \leftarrow c_{i,f}^{\mathcal{C}_{(l_t)}} + \alpha \left(\frac{\delta}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} D(\Delta t) c_{(l_t),f}^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{i,f}^{\mathcal{C}_{(l_t)}}\right)$;

   $c_{j,f}^{\mathcal{C}_{(l_t)}} \leftarrow c_{j,f}^{\mathcal{C}_{(l_t)}} + \alpha \left(-\frac{\delta}{|\mathcal{C}_{(l_t)}|} \sum_{c_{(l_t)} \in \mathcal{C}_{(l_t)}} D(\Delta t) c_{(l_t),f}^{\mathcal{C}_{(l_{t+1})}} - \lambda_\Theta c_{j,f}^{\mathcal{C}_{(l_t)}}\right)$;

   for $c_{(l_t)} \in \mathcal{C}_{(l_t)}$ **do**

    $c_{(l_t),f}^{\mathcal{C}_{(l_{t+1})}} \leftarrow c_{(l_t),f}^{\mathcal{C}_{(l_{t+1})}} + \alpha \left(\delta \frac{D(\Delta t)}{|\mathcal{C}_{(l_t)}|} (c_{i,f}^{\mathcal{C}_{(l_t)}} - c_{j,f}^{\mathcal{C}_{(l_t)}}) - \lambda_\Theta c_{(l_f)}^{\mathcal{C}_{(l_{t+1})}}\right)$;

   end

  end

until *convergence or reach the maximal number of iterations*;

candidate list of the next move according to the recovered missing value in Tensor. The BPR learning algorithm for TAD-FPMC is detailed as in Algorithm 1.

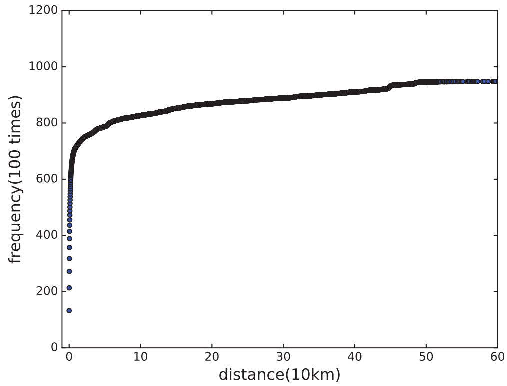
3.3. From Category to Location

3.3.1. Metric-Based Location Recommendation. In this section, we propose a metric-based approach to achieve the top-k location list given the category recommendation results C_p. Figure 3 shows the relation between the spatial distance and the frequency of the visits. We can observe that most of the successive check-ins are distributed within 5 kilometers, which illustrates the spacial locality of check-ins. Here we propose a metric $score_{l,u}$ to evaluate the importance of the location by considering the distance from the candidate of the next locations to the current location, the visiting frequency over the location candidate, and so forth, with respective to user and location, given as

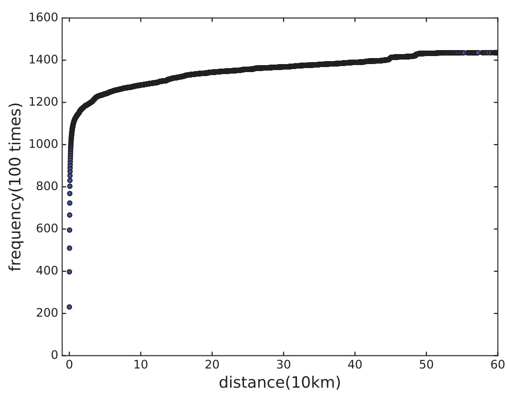
$$score_{l,u} = \left(\sum_{\tau} H_u(\tau, l)\right) \left(\frac{\psi}{Dist(l_t, l)}\right)^{f(C_l \cap C_p)}, \tag{11}$$

where H_u is a $|\mathcal{T}| \times |\mathcal{L}|$ matrix that describes the visiting history of user u. And $H(\tau, l)$ is the number of check-in records on location l during τ. $Dist(l_t, l)$ indicates the distance from l to the current location l_t. ψ is the threshold indicating the importance of the distance. $f(C_l \cap C_p)$ is given as

$$f(C_l \cap C_p) = e^{\sum_{c \in (C_l \cap C_p)} softmax(P(c|C_{(l_t)}))}, \tag{12}$$



(a) LA



(b) NYC

Fig. 3. Check-in frequency versus interval distance.

where $C_l \subset \mathcal{C}$ is the subset of categories that the location l belongs to. The softmax function is used to normalized the weights of c according to its category's ranking position in C_p.

3.3.2. Location Recommendation with Distance-Weighted HITS. To further boost the location recommendation performance by utilizing more user features (e.g., residence, age), we propose an alternative approach, which is referred to as distance-weighted HITS algorithm to predict the preference to the locations and achieve the personalized location ranking list for recommendation.

We regard each user as a hub page and each location as an authority page. A page of high authority is linked by a lot of hub pages, just like a popular POI will be visited by many people. For all of the users $\mathcal{U}$ and the locations $\mathcal{L}_{\mathcal{C}}$, the value of authority and the score of hub can be iteratively updated as follows:

$$u_n.H^{(k+1)} = \sum_{l \in \mathcal{L}_{u_n}} l.A^{(k)} \tag{13}$$

$$l_m.A^{(k+1)} = \sum_{u \in \mathcal{U}_{l_m}} \frac{1}{dist(l_t, l_m)} u.H^{(k)}, \tag{14}$$

ALGORITHM 2: Location Recommendation with Distance-Weighted HITS

Input: $L_{\mathcal{C}}, \mathcal{U}, G, u^*, l_t$
Output: $\vec{\mathcal{L}_{\mathcal{C}}}.A$
$G_{u^*} = findGroup(G, u^*)$;
$\vec{\mathcal{L}_{\mathcal{C}}}.A$ = Unit vector;
$\vec{\mathcal{U}}.H$ = Zero vector;
repeat
  for $u_n \in G_{u^*}$ **do**
    $u_n.H = \sum_{l \in \mathcal{L}_{u_n}} l.A$;
  end
  for $l_m \in \mathcal{L}_{\mathcal{C}}$ **do**
    $l_m.A = \sum_{u \in \mathcal{U}_{l_m} \bigcap G_{u^*}} \frac{1}{dist(l_t, l_m)} u.H$;
  end
until *convergence or reach the maximal number of iterations*;
$\vec{\mathcal{L}_{\mathcal{C}}}.A$ = sortByDescending($\vec{\mathcal{L}_{\mathcal{C}}}.A$);

where $u_n.H$ and $l_m.A$ denote the hub value of user u_n and authority value of location l_m, respectively. $\mathcal{L}_{u_n} \subset \mathcal{L}_{\mathcal{C}}$ denotes the set of locations visited by user u_n. $\mathcal{U}_{l_m}$ denotes the set of users who visited the location l_m. $dist(l_t, l_m)$ denotes the distance between the current location and the candidate location l_m. The hub value and authority value are iteratively updated according to Equation (13) and Equation (14) until convergence. The location ranking list is then achieved by sorting the authority value of locations decreasingly.

Rather than taking all of the users homogeneously, we also exploit users' demographics and frequently visited locations to form groups. In detail, we represent each user with a vector by using the frequency of check-ins on different location categories and the longitude and latitude of users' residence location. We adopt TF-IDF to measure the similarity between users. Then k-means is used to cluster the vectorized users to form groups. With the assumption that the users in the same group share similar interests in POIs and should contribute more to its fellow group members in updating the hub and authority values, we further revised the updating rules with both group information and distance constraint considered, which is detailed in Algorithm 2.

4. EXPERIMENTS

In our evaluation, we investigate the operating efficiency and prediction quality of our proposed methodologies. We address mainly two aspects: category prediction and location recommendation. For category prediction, we intend to justify the importance of the time-delay factor and the significance of temporal information. We also investigate the performance of the PITF model compared with other conventional standard tensor factorization approaches. With respect to location recommendation, we show the effectiveness of incorporating both the distance constraints and the location popularity.

4.1. Datasets

We evaluate our model on the data collected from Foursquare [Bao et al. 2012], which contains users' check-in data of New York and Los Angeles from January 2010 to June 2011. We use the catalog Foursquare provided to infer the mapping from thousands of locations to 249 categories. Eighty percent of the data are used as the training set and 20% of the data are used as the test set. The statistics of the data are detailed as in Table II.

Table II. The Statistics of Data

City	User	Location	Category	Tip
New York	2,581	206,416	249	166,530
Los Angeles	1,604	215,614	249	109,526

Table III. Models for Comparison of Category Prediction

Models	Scale	Describe
Matrix Factorization (MF) [Koren et al. 2009]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{C}\rvert$	MF is a well-known method in collaborative filtering. We take the MF as baseline for the performance comparison.
Probabilistic Matrix Factorization (PMF) [Salakhutdinov and Mnih 2011]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{C}\rvert \times \lvert\mathcal{C}\rvert$	PMF is widely used in traditional recommender systems.
Factorized Personalized Markov Chain (FPMC) [Rendle et al. 2010]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{C}\rvert \times \lvert\mathcal{C}\rvert$	This method formalizes the user's preference as a personalized Markov chain.
TA-FPMC	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{T}\rvert \times \lvert\mathcal{C}\rvert \times \lvert\mathcal{C}\rvert$	TA-FPMC stands for our proposed fourth-order time-aware FPMC model without considering time-decaying influence.
Tucker Decomposition (TD) [Symeonidis et al. 2008]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{T}\rvert \times \lvert\mathcal{C}\rvert \times \lvert\mathcal{C}\rvert$	TD factorizes a higher-order cube into a core tensor and one factor matrix for each dimension.
Canonical Decomposition (CD) [Carroll and Chang 1970]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{T}\rvert \times \lvert\mathcal{C}\rvert \times \lvert\mathcal{C}\rvert$	CD, aka parallel factor analysis (PARAFAC) [Harshman 1970], is an approach for tensor factorization with a model equation of linear complexity.

4.2. Evaluation Metric

Our recommendation task is to provide a list of Top-N recommended categories (locations) among which only at most one will be picked by the user, which makes the upper bound of the precision less than $1/|recommenationlist|$. Instead, we define that if the intersection of the actual categories set and the list of the Top-N recommended categories is not null, or the list of Top-N recommended locations contains the actually visited locations, then this category prediction or location recommendation is correct. The evaluation metric is given as

$$P@N = \frac{\text{the counts of correct predictions}}{\text{the total number of recommendation rounds}}. \tag{15}$$

4.3. Category Prediction

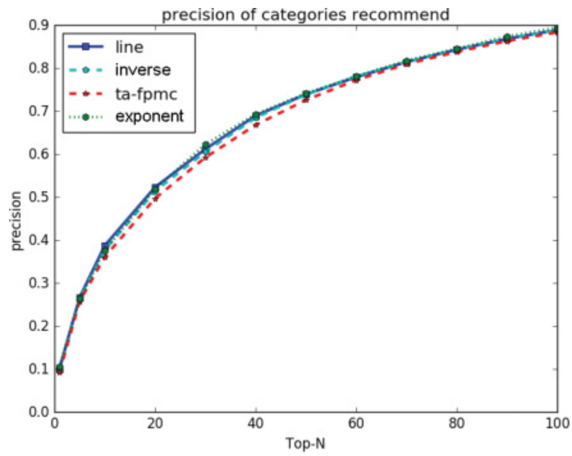
In this section, we compare our TAD-FPMC model for category prediction with a series of state-of-the-art methods and different tensor factorization approaches. The differences among these approaches are summarized in Table III.

4.3.1. Impact of $D(\Delta t)$. With the assumption that the longer the interval of two successive check-ins, the smaller the impact of the last check-in to the next check-in, we adopt three different time-decaying functions to simulate the time-varying impact as follows:

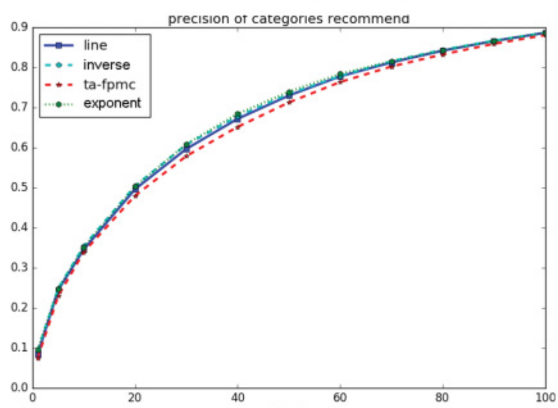
$$D_1(\Delta t) = e^{-\lambda_1 \Delta t} \tag{16}$$

$$D_2(\Delta t) = \frac{1}{1 + \lambda_2 \Delta t} \tag{17}$$

$$D_3(\Delta t) = 1 - \lambda_3 \Delta t, \tag{18}$$



(a) Category prediction of LA



(b) Category prediction of NY

Fig. 4. Comparison for the time-decaying factors.

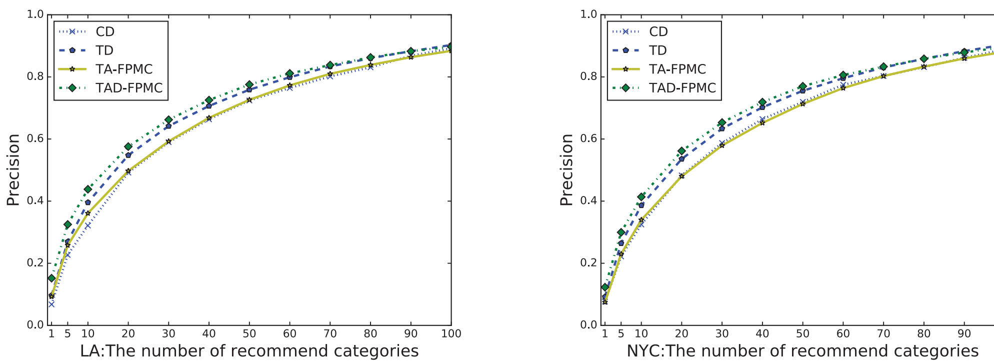
where $\lambda > 0$. For the legend of Figure 4, " exponent," "inverse," and "line" stand for TAD-FPMC incorporated with the time-decaying functions of Equation (16), Equation (17) and Equation (18), respectively. As shown in Figure 4, for both LA data and NYC data, the performance achieved by " exponent," "line," and "inverse" appear similar, but all slightly outperform TA-FPMC, which indicates the effectiveness of the incorporation of the time-decaying factor.[4] Hereafter, we use the exponential function exclusively to the TAD-FPMC model simulating the time-decaying feature for POI recommendation in the following experiments.

4.3.2. Analysis of Tensor Factorization Models. In this article, we adopt PITF to address tensor factorization, which will show its effectiveness compared with two classic tensor factorization approaches: - CD and TD. The illustrations of TD, CD, and PITF in terms of time complexity, space complexity, and factorization approach are tabulated in Table IV, where D is the number of cases in the training set, and K is the dimension of hidden vectors. Compared to CD, PITF drops the runtime from $O(D \cdot K^4)$ to $O(D \cdot K)$. For the

[4]The empirical settings of λ_1, λ_2, and λ_3 are 0.4, 0.1, and 0.02, respectively.

Table IV. Tensor Factorization Models

	Time Complexity	Space Complexity	Factorization
CD	$O(D \cdot K)$	$O((\lvert U\rvert + \lvert T\rvert + \lvert C\rvert) \cdot K)$	$S_{pqrt} = \sum_f A_{pf} \cdot B_{qf} \cdot C_{rf} \cdot D_{tf}$
TD	$O(D \cdot K^4)$	$O(K^4 + (\lvert U\rvert + \lvert T\rvert + \lvert C\rvert) \cdot K)$	$S_{pqrt} = \sum_{ijkl} A_{ijkl} \cdot B_{pi} \cdot C_{qj} \cdot D_{rk} \cdot E_{tl}$
PITF	$O(D \cdot K)$	$O((\lvert U\rvert + \lvert T\rvert + \lvert C\rvert) \cdot K)$	$S_{pqrt} = \sum_f \left(A_{pf} \cdot D^A_{tf} + B_{qf} \cdot D^B_{tf} + C_{rf} \cdot D^C_{tf}\right)$



(a) Category prediction of LA (b) Category prediction of NY

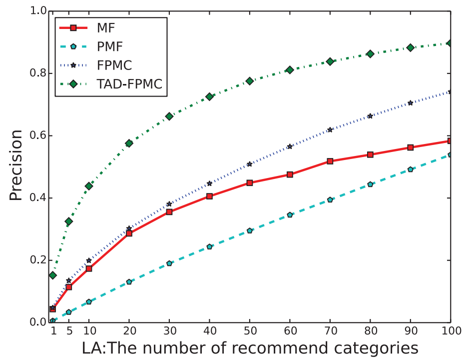
Fig. 5. Category predication comparison versus tensor factorization models.

Table V. Performance Comparison for Tensor Factorization Models

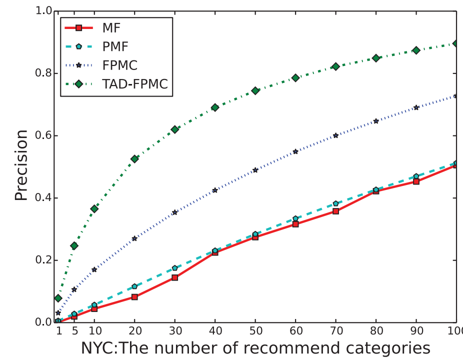
Metrics	LA				NYC			
	CD	TD	TA-FPMC	TAD-FPMC	CD	TD	TA-FPMC	TAD-FPMC
P@1	0.0677	0.0964	0.0928	**0.1519**	0.0767	0.0921	0.0747	**0.1230**
Improve	124.37%	57.57%	63.69%		60.37%	33.55%	64.66%	
P@5	0.2270	0.2695	0.2580	**0.3250**	0.2221	0.2642	0.2298	**0.2996**
Improve	43.17%	20.59%	25.97%		34.89%	13.40%	30.37%	
P@10	0.3216	0.3957	0.3610	**0.4382**	0.3249	0.3863	0.3397	**0.4136**
Improve	36.26%	10.74%	21.39%		27.30%	7.07%	21.75%	
P@20	0.4920	0.5477	0.4974	**0.5756**	0.4829	0.5357	0.4801	**0.5615**
Improve	16.99%	5.09%	15.72%		16.28%	4.82%	16.95%	
P@50	0.7242	0.7588	0.7262	**0.7753**	0.7195	0.7552	0.7130	**0.7699**
Improve	7.07%	2.19%	6.77%		7.00%	1.95%	7.98%	
P@100	0.8920	0.9027	0.8839	**0.8971**	0.8887	0.9040	0.8812	**0.8965**
Improve	0.57%	−1.29%	1.49%		0.88%	−0.83%	1.74%	

factorization, we can find that both TD and CD use the sum of several parameters' product to recover the original value, and this makes TD and CD more difficult to converge than PITF.

For prediction evaluation, we show the performance comparison among three factorization models in Figure 5 and Table V. Note that both TA-FPMC and TAD-FPMC models are based on PITF. In our experiment, TD obtains better results than TA-FPMC. A possible reason is that TD decomposes more thoroughly at the expense of much more running time. The TD model needs about 2 days to achieve a good prediction quality, while the PITF-based model (TA-FPMC/TAD-FPMC) is able to obtain relatively good performance within 3 hours in our datasets. Therefore, the PITF-based model is an ideal choice for tradeoff between the running time and prediction precision. Moreover, TAD-FPMC consistently outperforms other methods.



(a) Category prediction of LA



(b) Category prediction of NY

Fig. 6. Comparison of category prediction.

Table VI. Performance Comparison of Category Prediction

Metrics	LA				NYC			
	MF	PMF	FPMC	TAD-FPMC	MF	PMF	FPMC	TAD-FPMC
P@1	0.0433	0.0057	0.0477	**0.1519**	0.0016	0.0060	0.0310	**0.1230**
Improve	250.81%	2564.91%	218.45%		7587.50%	1950.00%	296.77%	
P@5	0.1142	0.0336	0.1351	**0.3250**	0.0197	0.0283	0.1063	**0.2996**
Improve	184.59%	867.26%	140.56%		1620.81%	958.65%	181.84%	
P@10	0.1734	0.0666	0.1992	**0.4382**	0.0444	0.0571	0.1700	**0.4136**
Improve	152.71%	557.96%	119.98%		831.53%	624.34%	143.29%	
P@20	0.2863	0.1305	0.3023	**0.5756**	0.0822	0.1160	0.2699	**0.5615**
Improve	101.05%	341.07%	90.41%		583.09%	384.05%	108.04%	
P@50	0.4486	0.2949	0.5088	**0.7753**	0.2744	0.2843	0.4893	**0.7699**
Improve	72.83%	126.90%	52.38%		180.58%	170.81%	57.35%	
P@100	0.5834	0.5389	0.7412	**0.8971**	0.5053	0.5127	0.7280	**0.8965**
Improve	53.77%	66.47%	20.03%		77.42%	74.86%	23.15%	

4.3.3. Experimental Comparisons Among Prediction Models. Figure 6 shows the performance comparison with the state-of-the-art approaches for category prediction. It is obvious that TAD-FMPC outperforms FPMC, PMF, and MF by a large margin, which further verifies that the incorporation of the temporal information is crucial for POI recommendation. The detailed performance and its corresponding improvement over different settings of the size of candidates are tabulated in Table VI.

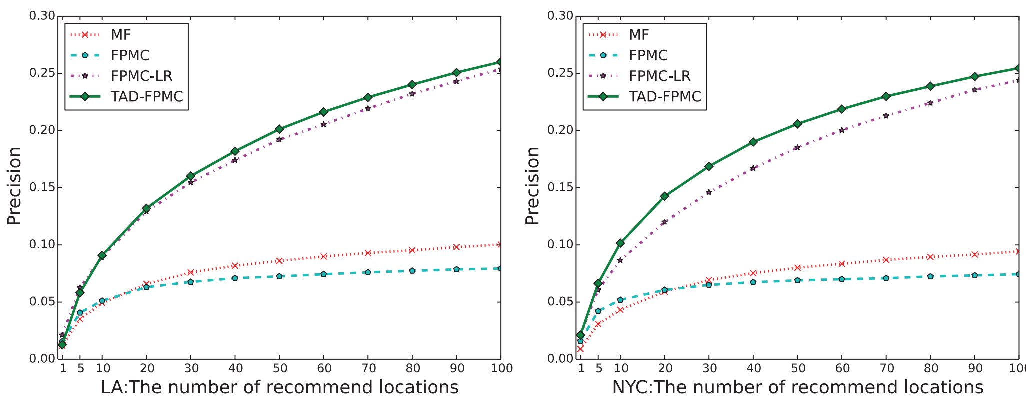
4.4. Location Recommendation

This section mainly consists of four parts for experimental evaluation. First, we further verify the importance of temporal information and distance constraint to the POI recommendation. Second, we propose a series of HIT-based POI ranking algorithms for location recommendation given the predicted categories (second step of our framework). Third, we conduct a more detailed analysis over our proposed two-step methodology. The recommendation ability for new POIs is also evaluated. The description of compared approaches in our experiments is listed in Table VII.

4.4.1. TAD-FPMC for Location Prediction. Note that our proposed TAD-FPMC can be adopted to predict the location directly if the tensor is constructed from the checked-in locations instead of the checked-in categories. Figure 7 shows the experimental results of location prediction achieved by applying TAD-FPMC over checked-in tensor $\hat{\chi} \in [0,1]^{|\mathcal{U}|\times|\mathcal{T}|\times|\mathcal{L}|\times|\mathcal{L}|}$ and that of other state-of-the-art approaches. TAD-FPMC

Table VII. Models for Comparison to Location Recommendation

Models	Scale	Describe
Matrix Factorization (MF)	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{L}\rvert$	MF is a well-known method in collaborative filtering. We take the MF as baseline for the performance comparison.
Factorized Personalized Markov Chain (FPMC)	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{L}\rvert \times \lvert\mathcal{L}\rvert$	This method formalizes the user's preference as a personalized Markov chain for one-step POI recommendation.
FPMC-LR [Cheng et al. 2013]	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{L}\rvert \times \lvert N_d(\mathcal{L})\rvert$	$N_d(\mathcal{L})$ is the neighbor location set filtered by distance d. FPMC-LR is introduced for locatized region.
TAD-FPMC	$\lvert\mathcal{U}\rvert \times \lvert\mathcal{T}\rvert \times \lvert\mathcal{L}\rvert \times \lvert\mathcal{L}\rvert$	We turn the TAD-FPMC we proposed into the model for one-step location recommendation directly.



(a) Location prediction of LA (b) Location prediction of NYC

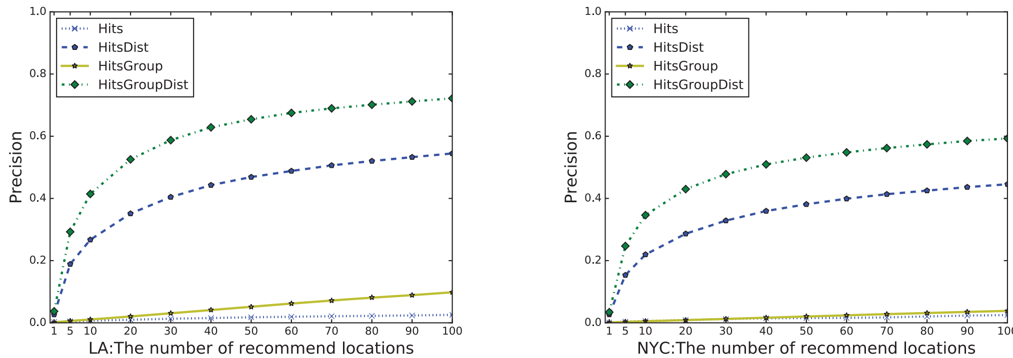
Fig. 7. Comparison of location prediction with other models.

outperforms the other approaches in both LA and NYC datasets. FPMC-LR improves FPMC over at least 30%. This implies that the distance constraint and the temporal information play important roles when performing successive personalized POI recommendations.

4.4.2. HITs for Location Recommendation. Given the predicted category list, we shall conduct the second step of our proposed framework: location recommendation. Here we propose to adapt HITs with variants of combinations of group information and distance constraints incorporated to obtain the final location ranking list. The groups are achieved by k-means considering users' demographics and frequently visited locations.

Figure 8 shows the performance comparison among the HITS-based ranking algorithms. For both datasets, HitsDist performs far better than Hits and HitsGroup, illustrating the importance of considering the distance constraint. With further incorporation of group information, we can see that HitsDistGroup outperforms the other three alternatives by a large margin. It illustrates the effectiveness of considering both distance constraints and location popularity for location recommendation. The detailed comparisons are listed as in Table VIII.

4.4.3. Analysis of Two-Step Framework. Figure 9 shows the experimental results of the location prediction achieved by TAD-FPMC directly and the results achieved by the two-step POI recommendation (TAD-FPMC-HitsGroupDist, TAD-FPMC-Metric). It's obvious that TAD-FPMC-HitsGroupDist and TAD-FPMC-Metric achieve better performance than TAD-FPMC, which outperforms in Figure 9. It further verifies that

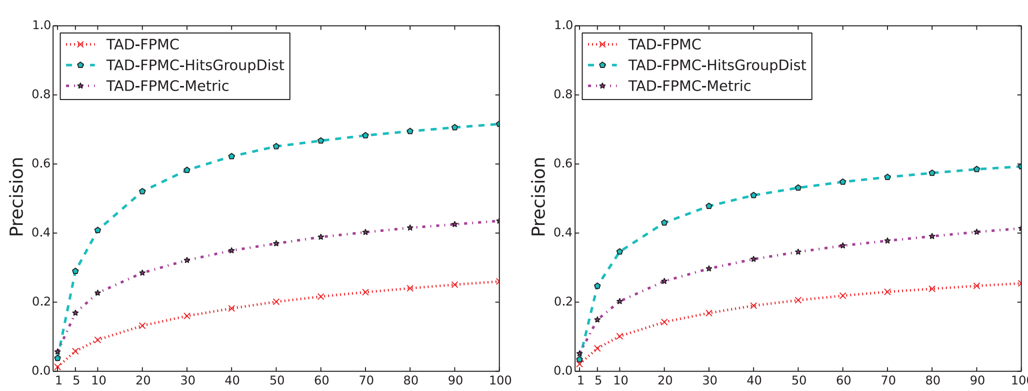


(a) LA (b) NYC

Fig. 8. Performance Comparison among Hits-based Location Ranking methods.

Table VIII. Comparison of Four Hits Methods

Metrics	LA				NYC			
	Hits	HitsGroup	HitsDist	HitsDistGroup	Hits	HitsGroup	HitsDist	HitsDistGroup
P@1	0.0023	0.0012	0.0265	**0.0370**	0.0009	0.0007	0.0266	**0.0342**
Improve	1,508.70%	2,983.33%	39.62%		3,700.00%	4,785.71%	28.57%	
P@5	0.0047	0.0060	0.1886	**0.2927**	0.0035	0.0032	0.1533	**0.2469**
Improve	6,127.66%	4,778.33%	55.20%		6,954.29%	7,615.63%	61.06%	
P@10	0.0065	0.0105	0.2671	**0.4143**	0.0054	0.0054	0.2196	**0.3462**
Improve	6,273.85%	3,845.71%	55.11%		6,311.11%	6,311.11%	57.65%	
P@20	0.0099	0.0205	0.3513	**0.5255**	0.0088	0.0086	0.2865	**0.4299**
Improve	5,208.08%	2,463.41%	49.59%		4,785.23%	4,898.84%	50.05%	
P@50	0.0178	0.0511	0.4687	**0.6544**	0.0154	0.0206	0.3812	**0.5311**
Improve	3,576.40%	1,180.63%	39.62%		3,348.70%	2,478.16%	39.32%	
P@100	0.0256	0.0981	0.5446	**0.7218**	0.0249	0.0383	0.4456	**0.5928**
Improve	2,719.53%	635.78%	32.54%		2,280.72%	1,447.78%	33.03%	



(a) Location prediction of LA (b) Location prediction of NYC

Fig. 9. Comparison of location prediction between one-step and two-step recommendation.

Table IX. Performance Comparison for Location Prediction (LA)

Mertrics	MF	FPMC	FPMC-LR	TAD-FPMC	TAD-FPMC-Metric	TAD-FPMC-HitsGroupDist
P@1	0.0118	0.0155	0.0212	0.0127	0.0410	**0.0370**
Improve	213.56%	138.71%	74.53%	191.34%	−9.76%	
P@5	0.0351	0.0406	0.0625	0.0582	0.1338	**0.2927**
Improve	733.90%	620.94%	368.32%	402.92%	118.76%	
P@10	0.0491	0.0511	0.0895	0.0909	0.1894	**0.4143**
Improve	743.79%	710.76%	362.91%	355.78%	118.74%	
P@20	0.0658	0.0630	0.1291	0.1319	0.2529	**0.5255**
Improve	698.63%	734.13%	307.05%	298.41%	107.09%	
P@50	0.0860	0.0724	0.1920	0.2012	0.3487	**0.6544**
Improve	660.93%	803.87%	240.83%	225.25%	87.67%	
P@100	0.1003	0.0793	0.2538	0.2601	0.4299	**0.7218**
Improve	619.64%	810.21%	184.40%	177.51%	67.90%	

Table X. Performance Comparison for Location Prediction (NYC)

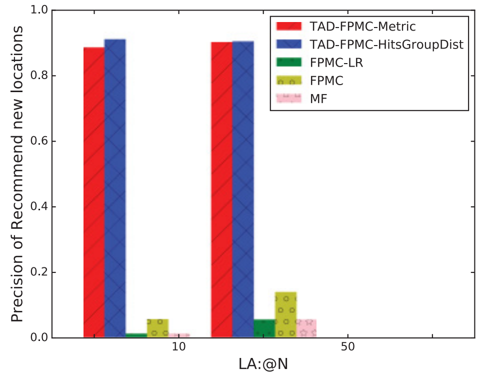
Metrics	MF	FPMC	FPMC-LR	TAD-FPMC	TAD-FPMC-Metric	TAD-FPMC-HitsGroupDist
P@1	0.0089	0.0159	0.0213	0.0211	0.0331	**0.0342**
Improve	284.27%	115.09%	60.56%	62.09%	3.32%	
P@5	0.0307	0.0421	0.0607	0.0663	0.1154	**0.2469**
Improve	704.23%	486.46%	306.75%	272.40%	113.95%	
P@10	0.0432	0.0518	0.0865	0.1015	0.1651	**0.3462**
Improve	701.39%	568.34%	300.23%	241.08%	109.69%	
P@20	0.0590	0.0605	0.1201	0.1425	0.2203	**0.4299**
Improve	628.64%	610.58%	257.95%	201.68%	95.14%	
P@50	0.0800	0.0689	0.1851	0.2059	0.3043	**0.5311**
Improve	563.88%	670.83%	186.93%	157.94%	74.53%	
P@100	0.0942	0.0744	0.2439	0.2546	0.3753	**0.5928**
Improve	529.30%	696.77%	143.05%	132.84%	57.95%	

the two-step methodology we proposed to categorize the locations can greatly alleviate data sparsity issues. In general, TAD-FPMC-HitsGroupDist performs better than TAD-FPMC-Metric; we argue that the gain comes from the utilization of group information in HitsGroupDist. However, when the size of the candidate list is set as 1, $N = 1$, TAD-FPMC-Metric is a good choice for POI recommendation.

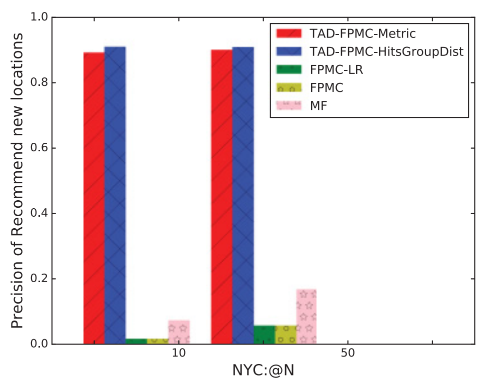
The experimental results of the location prediction for all models are shown in Table IX and Table X.

For case study, we found that one location that belongs to multiple categories has a better chance to be the next visiting location. For example, there is one test tip: $user : 1, Time : 5, categories : 207$ in our dataset. Our obtained candidate location list is l_{58797} and l_{11994}. l_{11994} belongs to two categories, 207 and 123, while l_{58797} only belongs to one category, 207. The location l_{11994} is the observed target venue, although l_{58797} is much nearer to the current location. This is reasonable as people intuitively tend to go to some centers providing high availability for multiple types of services and products just for convenience. The observation further validates the effectiveness of the category incorporation for POI recommendation.

Figure 10 shows the performances of recommending new locations for users of the recommender systems. It is very obvious that both of our proposed approaches TAD-FPMC-Mectric and TAD-FPMC-HGD perform far better than other comparative approaches with respect to recommending new POIs. Our guess is that the ability to effectively recommend new locations accounts for the effectiveness of the category prediction.



(a) New location rate of LA



(b) New location rate of NYC

Fig. 10. Comparison of new location rate.

We ran all our experiments on a machine with a Core i7-6700K 4.0GHz 8HT and the memory size of 32GB. Compared with the other models, TAD-FPMC is found to be the most efficient one. The running time of TAD-FPMC is around 3 hours, while FPMC and FPMC-LR consume over 35 hours. The efficiency benefits from the reduced dimensionality from the number of locations to the number of categories. More specifically, FPMC and FPMC-LR repeated their training process over $C^2_{|L|}$ pairs of the training data, while TAD-FPMC works toward $C^2_{|C|}$ pairs.

5. CONCLUSIONS

In this article, we have proposed a two-step approach to tackle POI recommendation. First, a fourth-order tensor-based TAD-FPMC model is presented for category prediction, which considers the influence of users' time-variant categorial preference and the decaying importance along with the spanned time period between two successive check-in records. Second, we propose a metric-based location prediction approach and a Distance-Weighted HITS algorithm, respectively, to rank the locations selected according to a previously achieved preferred category list. Both algorithms take the location's popularity and the distance constraint into account. Our experimental results obtained based on the Foursquare datasets show that a higher accuracy can be achieved compared to several existing state-of-the-art models and at the same time achieved with a higher efficiency. The second step of our proposed framework, location recommendation, is designed heuristically according to the distance constraint, group interest, and the category distribution, but has proved its effectiveness via extensive experiments. We would like to further investigate it in a more rigorous way in our future work.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful comments. This work has been partially supported by NSFC under Grant No. 61300178 and the National Program on Key Basic Research Project under Grant No. 2013CB329605.

REFERENCES

Jie Bao, Yu Zheng, and Mohamed F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 199–208.

J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika* 35, 3 (1970), 283–319.

Chen Chen, Hongzhi Yin, Junjie Yao, and Bin Cui. 2013a. Terec: A temporal recommender system over tweet stream. *Proceedings of the VLDB Endowment* 6, 12 (2013), 1254–1257.

Yan Chen, Jichang Zhao, Xia Hu, Xiaoming Zhang, Zhoujun Li, and Tat-Seng Chua. 2013b. From interest to function: Location estimation in social media. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence.* 180–186.

Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *26th AAAI Conference on Artificial Intelligence*.

Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. AAAI Press, 2605–2611.

Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1082–1090.

Chi-Yin Chow, Jie Bao, and Mohamed F Mokbel. 2010. Towards location-based social networking services. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 31–38.

Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 93–100.

Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence.* 1721–1727.

Richard A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. University of California at Los Angeles, Los Angeles, CA (1970), 1–84.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.

Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. 2007. Location-based recommendation system using bayesian users preference model in mobile devices. In *Ubiquitous Intelligence and Computing*. Springer, 1130–1139.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 811–820.

Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. ACM, 81–90.

Ruslan Salakhutdinov and Andriy Mnih. 2011. Probabilistic matrix factorization. Citeseer.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, 285–295.

Libo Song, David Kotz, Ravi Jain, and Xiaoning He. 2004. Evaluating location predictors with extensive Wi-Fi mobility data. In *23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Vol. 2. IEEE, 1414–1424.

Ja-Hwung Su, Hsin-Ho Yeh, Philip S. Yu, and Vincent S. Tseng. 2010. Music recommendation using content and context information mining. *IEEE Intelligent Systems* 25, 1 (2010), 16–26.

Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2008. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM Conference on Recommender Systems*. ACM, 43–50.

Waldo R. Tobler. 1970. A computer movie simulating urban growth in the detroit region. *Economic Geography* 46 (1970), 234–240.

De-Nian Yang, Chih-Ya Shen, Wang-Chien Lee, and Ming-Syan Chen. 2012. On socio-spatial group query for location-based social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 949–957.

Jihang Ye, Zhe Zhu, and Hong Cheng. 2013. Whats your next move: User activity prediction in location-based social networks. In *Proceedings of SIAM International Conference on Data Mining (SDM'13)*.

Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 325–334.

Yang Ye, Yu Zheng, Yukun Chen, Jianhua Feng, and Xing Xie. 2009. Mining individual life pattern based on location history. In *10th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009 (MDM'09)*. IEEE, 1–10.

Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. LCARS: A location-content-aware recommender system. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. 221–229.

Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 363–372.

Jia-Dong Zhang and Chi-Yin Chow. 2016. TICRec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. *IEEE Transactions on Services Computing* 9, 4 (2016), 633–646.

Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R. Lyu, and Irwin King. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *30th AAAI Conference on Artificial Intelligence*.

Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 1029–1038.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 791–800.

Received August 2016; revised December 2016; accepted February 2017