Knowl Inf Syst (2018) 56:559–579
https://doi.org/10.1007/s10115-017-1107-4

REGULAR PAPER

A time-aware trajectory embedding model for next-location recommendation

Wayne Xin Zhao[1,2,3] · **Ningnan Zhou**[1] · **Aixin Sun**[4] · **Ji-Rong Wen**[1,2] · **Jialong Han**[4] · **Edward Y. Chang**[5]

Received: 8 April 2016 / Revised: 8 May 2017 / Accepted: 21 September 2017 /
Published online: 4 October 2017

Abstract Next-location recommendation is an emerging task with the proliferation of location-based services. It is the task of recommending the next location to visit for a user, given her past check-in records. Although several principled solutions have been proposed for this task, existing studies have not well characterized the temporal factors in the recommendation. From three real-world datasets, our quantitative analysis reveals that temporal factors play an important role in next-location recommendation, including the periodical temporal preference and dynamic personal preference. In this paper, we propose a Time-Aware Trajectory Embedding Model (TA-TEM) to incorporate three kinds of temporal factors in next-location recommendation. Based on distributed representation learning, the proposed TA-TEM jointly models multiple kinds of temporal factors in a unified manner. TA-TEM also enhances the sequential context by using a longer context window. Experiments show that TA-TEM outperforms several competitive baselines.

Keywords Next-location recommendation · Distributed representation learning · Temporal factors

1 Introduction

With the increasing popularity of location-based social networks (LBSNs), such as Foursquare and Facebook Places, user check-in data in large volume becomes available.

✉ Jialong Han
jialonghan@gmail.com

1 School of Information, Renmin University of China, Beijing, China

2 Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China

3 Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou, China

4 School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore

5 HTC Research & Healthcare, San Francisco, CA, USA

As in September 2015, Foursquare has accumulated over 7 billion check-ins made by 55 million users.[1] The availability of the sheer volume of human check-in data leads to a variety of useful applications for better user experiences with location-based services. One of such tasks is the emerging next-location recommendation [2–4]. Next-location recommendation predicts the successive location(s) for a user to visit given her past check-in records (or trajectory subsequences). Temporal information plays an important role in this task. For example, a user has visited `"subway"` starting from `"home"`, then `"office"` (or working place) should be a suitable recommendation for this user, if the visiting time is in the morning of a working day; relaxing places should be recommended if the visiting time is on weekends.

Existing studies on next-location recommendation mainly model sequential transition patterns based on Markov chain property between consecutive check-ins [18,21]. The sequential transitions are limited to first-order transitions due to data sparsity and computational complexity, and longer sequential context cannot be captured. More importantly, there is a lack of comprehensive and deep consideration of multiple kinds of temporal factors in the recommendation task. In existing studies, user preference is typically considered as static, which does not reflect the evolving characteristics of user interests. For example, a student might check-in more often at universities during semester period while check-in more often at working places when she serves as a summer intern with a company. Her check-in behaviors change in different time periods. The existing studies also ignore the periodical patterns. For example, on weekdays, a user is likely to check-in at the office in the morning while check-in at home in the evening.

The task of next-location recommendation is challenging, which is different from the more widely studied location recommendation task. First, as shown in the above example, a user's check-in behaviors would change over time. Second, even if we could derive the visiting patterns given a user (e.g., `"home"` → `"shop"` → `"lunch"` → `"shop"` → `"dinner"` on weekends), it is still difficult to infer the exact location for each pattern slot, since multiple candidate locations might be suitable to fill in the corresponding slot. We present such an illustrative example in Fig. 1. As shown in the example, for the given user, the next location for dinner should be generated based on multiple kinds of considerations, including her own preference, previously visited locations and other temporal factors. The key point in generating effective location recommendations is to accurately capture the associations between multiple context factors. This calls for a principled way to flexibly characterize multiple factors in the recommendation algorithm. In this paper, our focus is to develop an effective time-aware algorithm for next-location recommendation.

Temporal information is more important to consider in the task of next-location recommendation, which aims to generate successive recommendations based on both the check-in history and the current location. Although temporal factors have been considered in location recommendation tasks [26], to the best of our knowledge, temporal factors have not been well characterized in next-location recommendation.

To incorporate temporal factors in next-location recommendation, we propose a novel Time-Aware Trajectory Embedding Model (TA-TEM). Recent progress in deep learning and neural networks in natural language processing [13] shows that (1) sequential semantic relatedness can be effectively captured by surrounding word contexts and (2) distributed representation is more resistant to data sparsity by using a "dense" vector. Inspired by these findings, we adopt a distributed representation model to characterize check-in data. More specifically, in our model, a check-in is generated based on multiple kinds of temporal contexts, and each kind of temporal context is modeled as a unique distributed vector, i.e., a

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

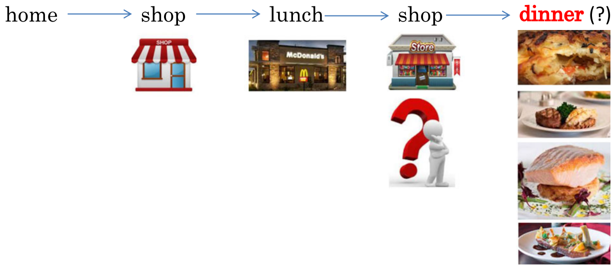

Fig. 1 An illustrative example for the visiting pattern of a user on weekends. She has followed the pattern "home" → "shop" → "lunch" → "shop" → "dinner". Now she would like to make the next visit and has four restaurants as candidates for dinner. For ease of illustration, each restaurant is presented by a representative kind of food

fixed-length embedding vector. In this way, multiple kinds of temporal factors can be integrated in a unified manner. We consider three kinds of temporal factors in the generation of a check-in: (i) the trajectory sequence as sequential context, (ii) both the dynamic and static user preferences for capturing the changes in user interests, and (iii) weekly and daily check-in patterns at the scale of day and hour, respectively. Using such a distributed representation model, it is flexible to encode multiple context factors into a unified context vector using the simple vector aggregation operation. It is more resistant to the data sparsity problem compared to the traditional one-hot representation. Although simple, it is effective to capture the associations between multiple context factors and generate high-quality recommendations.

Another important merit of the proposed model TA-TEM is that it is convenient to analyze the associations between different kinds of contextual information in the embedded space. We derive several interesting findings based on the learned embedding vectors, e.g., $\texttt{Airport+USER}_{\texttt{Aug.}} \approx \texttt{Beach}$ while $\texttt{Airport+USER}_{\texttt{Dec.}} \approx \texttt{Ski Area}$. The contributions of this paper are summarized as follows:

- We show the importance of temporal factors for the task of next-location recommendation through analysis of real-world check-in data, including dynamic user preference and periodical temporal preference.
- We develop a distributed representation model named TA-TEM to jointly model three kinds of temporal factors. In TA-TEM, we also enhance the sequential context by using a longer context window.
- We conduct extensive experiments on three real-world datasets. The experimental results demonstrate the superiority of our model, and we also find interesting qualitative analysis results. We give a detailed performance analysis for TA-TEM and have found that distributed representation models are effective to characterize sequential patterns for check-ins.

2 Related work

Our work is mainly related to the topic of location recommendation. Most location recommendation algorithms aim to recommend locations based on users check-in history. The task is often modeled as a collaborative filtering task similar to product and music recommendations. Next-location recommendation aims to generate effective recommendations based on both the check-in history and the current location.

General location recommendation usually exploits user preference and geographical influence by adopting collaborative filtering (CF) techniques, matrix factorization (MF) techniques and generative probabilistic models. The user-based CF framework for location recommendation shows that user-based CF outperforms item-based CF [24]. The item-based CF is also extended by considering the travel distance as penalty [7]. When the MF technique was originally applied to location recommendation, it performed worse than user-based CF and item-based CF methods [17]. Typically, in these approaches, CF and MF methods fit nonzero check-ins and they are likely to suffer from data sparsity [8]. Hence, the weighted MF method is proposed to fit both zero and nonzero check-ins [10]. Latent factors are also exploited by using tensor decomposition such as user-location-activity [28] or generative probabilistic model [11,23,25].

For next-location recommendation, sequential patterns have been widely considered to model the relatedness between surrounding check-in locations. Several studies utilize the Markov chain property to capture the transition patterns, which suffers from the large recommendation space [1]. To address this difficulty, it is proposed to first learn a mixed hidden Markov model to predict the category of user activity at the next step and then predict the most likely location given the estimated category distribution [3]. Factorized personalized Markov chain (FPMC) algorithm is also applied to embed the personalized Markov chains under localized regions [2,18]. FPMC represents each item with two independent vectors but fails to model the relations among multiple items. To overcome this drawback, personalized ranking metric embedding model is proposed to learn personalized check-in sequences by projecting the distance between location pairs and user-location pairs into two different latent spaces [4].

Recently, time-aware location recommendation also receives considerable attention, where temporal information is explored to better suit the task. The user-based CF is first extended to incorporate both the temporal and geographical effects in a linear combination framework [26]. Recently, ordered weighted pairwise classification criterion [19] has been applied to time-aware recommendation. It considers the ranking of locations for each recommendation as a set of pairwise classification problems and emphasizes the top-k positions by assigning higher weights [8]. In addition, textual information has also been utilized when it is available, which includes text descriptions [9] and categories of locations [12].

Although temporal factors have been explored in general location recommendation, very few studies focus on the task of time-aware next-location recommendation. Existing studies on this task typically characterize first-order sequential transition patterns based on Markov chain property, only very short sequential contexts can be modeled. On the model side, these previous studies are mainly developed in the matrix factorization framework or in a probabilistic approach. As a comparison, our work is inspired by the distributed representation models (e.g., `word2vec` [13] and `paragraph2vec` [6]) in natural language processing, and it can capture longer sequential context.

The current work is based on our previous study [29] in the idea of applying distributed representation learning to characterize trajectory sequences. Although our model has been developed in a similar way as that in [29], we have a different research focus. In our current model, we carefully analyze and design several temporal factors tailored to the trajectory sequence modeling, including periodical contexts and static/dynamic user preferences. Based on three real-world datasets, we preform extensive experiments and investigate the effect of the considered temporal factors in the task of next-location recommendation.

3 Observations on trajectories

As shown in [2,18], sequential influence is one of the most important temporal factors to consider in trajectory data, i.e., Markov chain property exist between consecutive check-in points by a user. In this section, we study another two kinds of temporal factors on real-world trajectory data. The same datasets will be used in our experiments.

Datasets. We analyze three public geosocial networking datasets, each consisting of one year check-ins data, namely *Foursquare* [25], *Gowalla* [17] and Foursquare$_{large}$ [22]. The three datasets have different check-in densities, as shown in Table 1. The check-in data in the first Foursquare dataset is much sparser than the other datasets. The details about number of users, check-ins, and locations are also reported in Table 1. From the three datasets, we make the following two observations. We only report the results from Gowalla dataset, and the observations on the other two datasets are similar.

Observation 1 *User preference on check-ins changes over long time periods (e.g., a month).*

Personal preference is of priority to contribute to the location choices in trajectories, which reflects the overall user interests, habits and behavioral patterns. Intuitively, the interests of a user are likely to change after some time (e.g., a month), which might lead to different visiting behaviors in different time periods.

Given a user u, we compute the overlap ratio value (ORV) by averaging the Jaccard similarities of the two sets of locations visited by u in each two consecutive time periods. Here, we set a time period to be a month.

$$\mathrm{ORV}_u = \mathrm{Average}_i\left(JarccardSim(\mathcal{L}^{(u,i)}, \mathcal{L}^{(u,i+1)})\right) \tag{1}$$

In the above equation, $\mathcal{L}^{(u,i)}$ is the set of locations visited by u in the i-th month. We select the top 1000 users with the most number of check-in records, then obtain the mean ORV over all these users. The mean ORV is 0.035 ($\pm$0.02). The small Jaccard similarity value suggests that there is little overlap between the check-in locations of a users in two consecutive months. In other words, user preference for check-in behavior changes in long time periods.

Observation 2 *Periodical check-in patterns are significant.*

Intuitively, a user is likely to have some regular daily and weekly activities, such as having lunch at noon and relaxing on weekends. Thus, the generation of a location is likely to be influenced by the corresponding temporal information, such as the hour in a day and the day in a week.

We first study weekly patterns. We first split the trajectories into multiple intervals by days and then combine the visited locations from all the users with respect to the i-th interval into a location set, $\mathcal{L}^{(i)}$. Since a week consists of seven days, we further group these location sets into seven clusters by day. Then, we can compute the average "intra-" and "inter-" similarity of two location sets from these seven clusters, where "intra-" and "inter-" indicate the two

Table 1 Statistics of our datasets

Dataset	# Users	# Check-ins	# Locations
Foursquare	3667	263,443	80,615
Gowalla	175,777	9,281,803	1,399,974
Foursquare$_{large}$	266,909	33,278,683	3,680,126

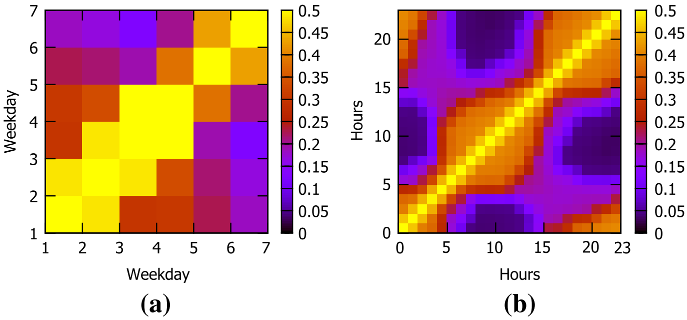

Fig. 2 Heat map for average intra- and inter-similarities for two location sets on different timescales. **a** Weekly patterns, **b** daily patterns

location sets come from the same and different cluster(s). We use the Jaccard coefficient to measure the similarity between two location sets. By enumerating all index pairs in a week, we can derive $7 \times 7 (= 49)$ entries and plot them in a heat map. Shown in Fig. 2a, the Jaccard coefficients of diagonal entries are significantly larger than those of the non-diagonal entries, which indicates the two location sets generated on the time index in a week are more similar than those with different time indices. We can perform the similar analysis based on hour index, and the results are plotted in Fig. 2b, which gives the similar finding that the user trajectory shows interesting temporal patterns. Note that the above analysis is performed by month. The reported results are calculated by averaging the values for multiple months, since using a long period will incorporate the effect of other factors, e.g., the preference variation found in Observation 1.

4 Time-aware trajectory embedding model

In this section, we present the Time-Aware Trajectory Embedding Model (TA-TEM) for next-location recommendation. The model characterizes three kinds of temporal factors in trajectory modeling. We start with a base model with the first kind of temporal factor, i.e., sequential context. Further, we integrate two kinds of temporal factors based on the two observations made in Sect. 3. The notions used in our model are summarized in Table 2.

4.1 Trajectory embedding: a base model

To model trajectory data, a straightforward approach is to constructing probabilistic models by maximizing the generative probability of trajectory sequences based on some criteria. The most widely adopted approach is to utilizing the sequential Markov chain property for deriving the transition probabilities between check-ins. Due to data sparsity, it is infeasible to estimate the transitions by using standard counting methods. The pioneering study [18] factorized the personalized Markov chain for capturing the Markov property between two consecutive items in a sequence. Furthermore, Cheng et al. [2] extended the previous model by incorporating region localization. The authors observed on two real-world check-in datasets that when two consecutive check-ins occur in a short time period, Markov chain property exists. These methods indeed model first-order sequential transition. However, longer sequential context of a check-in cannot be modeled.

Table 2 List of notations

Notations	Descriptions
u, u_m, t	Index variables for user, user personas, trajectory
ℓ, s, d, h	Index variables for location, time stamp, day, and hour
N_t	The length of a trajectory t
$\mathcal{U}, \mathcal{T}, \mathcal{L}$	The sets of users, trajectories, and locations
$\mathcal{T}^{(u)}$	The set of trajectories generated by user u
$\boldsymbol{\theta}$	An embedding vector for nodes in the binary tree created by hierarchical sampling
$\boldsymbol{v}$	An embedding vector
$\boldsymbol{v}_f$	The embedding vector for a contextual feature f
$\overline{\boldsymbol{v}}_\ell$	The averaged contextual embedding vector for location ℓ
$\boldsymbol{v}_u$, $\boldsymbol{v}_{u_m}$	Embedding vectors for a user and user persona
$\boldsymbol{v}_\ell$, $\boldsymbol{v}_d$, $\boldsymbol{v}_h$	Embedding vectors for location ℓ, day d and hour h
$CL(= 2K)$	The context window length
VS	The number of dimensions for embedding vectors

Inspired by the recent progress of deep learning and neural networks [13,14], we propose to use distributed representation method to characterize the sequential context in trajectory data. Formally, let t be a trajectory sequence generated by user u with N check-ins: $\langle u, \ell_1, s_1\rangle$, $\langle u, \ell_2, s_2\rangle, \ldots, \langle u, \ell_N, s_N\rangle$. We model each location ℓ_j with a VS-dimensional embedding vector $\boldsymbol{v}_{\ell_j}$.

Due to physical limit, road traffic or causality, the generation of the current location is influenced by its surrounding locations in a trajectory (called *sequential influence* in [4]), which can be considered as a specific kind of temporal factor. Not simply modeling the first-order sequential context, we consider modeling the context of surrounding locations in a window of length of $CL = 2K$. Let ℓ_{j-K}:ℓ_{j+K} denote the location sequence $\ell_{j-K}, \ell_{j-K+1}, \ldots,$ ℓ_{j+K} (excluding ℓ_j). Following [29], this idea resembles to that in word embedding models (e.g., word2vec [13]), where the semantics of a word is related to those of the surrounding words. Thus, we have the following:

$$\overline{\boldsymbol{v}}_{\ell_j} = \frac{1}{2K} \sum_{-K\leq k\leq K, k\neq 0, 0\leq j+k<N_t} \boldsymbol{v}_{\ell_{j+k}}, \tag{2}$$

where $\overline{\boldsymbol{v}}_{\ell_j}$ is also a VS-dimensional embedding vector modeling the sequential context of location ℓ_j, which can be easily extended to model other kinds of information as shown later. For locations which do not have $2 \times K$ surrounding locations, the above equation is modified accordingly. For simplicity, we present the general case, where a location has previous K and the successive K locations. With this formulation, we define the objective function for a trajectory t as follows:

$$\frac{1}{N_t} \sum_{j=1}^{N_t} \log Pr(\ell_j | \overline{\boldsymbol{v}}_{\ell_j}), \tag{3}$$

where a multi-class classifier is further deployed to yield each location via a softmax function as follows

$$Pr(\ell_j|\overline{\boldsymbol{v}}_{\ell_j}) = \frac{\exp\left(\overline{\boldsymbol{v}}_{\ell_j}^{\top} \cdot \boldsymbol{v}_{\ell_j}\right)}{\sum_{\ell' \in \mathcal{L}} \exp\left(\overline{\boldsymbol{v}}_{\ell_j}^{\top} \cdot \boldsymbol{v}_{\ell'}\right)}. \tag{4}$$

In the above formulations, the sequential context is characterized as a window with a tunable length of $2 \times K$. Note that, our modeling relaxes the original Markov chain property: the locations in a small sequential-context window are orderless (i.e., as in "bag-of-words" model). Although simple, an important merit of this model is that the increase of window size mainly affects the computational cost of the aggregation process (i.e., deriving $\overline{\boldsymbol{v}}_{\ell_j}$) instead of the exponential objective function.

4.2 Integrating user preference changes

The preference of a user over check-ins is likely to change with time (see Observation 1). To model this factor, we divide the check-ins into months and assume that a user u is associated with a unique "persona" u_m in the m-th month. The term "persona" is used to reflect the varying preferences of a user following [15]. Besides the preference change, we assume that a user also has a general preference over locations which is relatively stable over time to generate regularly visited locations like working offices and home. Then we can have

$$\overline{\boldsymbol{v}}_{\ell_j} = \boldsymbol{v}_u + \boldsymbol{v}_{u_m}, \tag{5}$$

where $\boldsymbol{v}_u$ and $\boldsymbol{v}_{u_m}$ are two VS-dimensional embedding vectors corresponding to the general user preference and the user persona in m-th month, respectively. We still use $\overline{\boldsymbol{v}}_{\ell_j}$ to denote the contexts of location ℓ_j. Note that a user u will correspond to a general vector $\boldsymbol{v}_u$ over the entire time span, and she will correspond to a unique vector $\boldsymbol{v}_{u_m}$ for each month. These two embedding vectors model long-term and short-term visiting preferences, respectively, for a user.

The time granularity for modeling short-term preference can be tuned according to different datasets and application scenarios. Now we adopt a simple fixed-length splitting method to segment the entire time span. We can also consider applying dynamic splitting techniques to derive a better period segmentation, such as burst-based methods [27].

4.3 Integrating periodical patterns

Periodical check-in patterns characterize weekly and daily moving patterns, made in Observation 2. When a user is to generate a trajectory, her behavior is not only influenced by her own preference but also the periodical preferences. Let h_j and d_t be the hour index in a day and the day index in a week for the j-th location in trajectory t. Formally, we have

$$\overline{\boldsymbol{v}}_{\ell_j} = \boldsymbol{v}_{h_j} + \boldsymbol{v}_{d_t}, \tag{6}$$

where $\boldsymbol{v}_{h_j}$ and $\boldsymbol{v}_{d_t}$ are the VS-dimensional embedding vectors for the hour index h_j and the day index d_t, respectively.

Different from user-specific preferences, periodical-related embedding vectors can characterize the temporal visiting patterns of the entire population. Besides, compared with short-term user preference, periodical preference can capture more fine-grained temporal visiting behaviors, which is likely to be useful in next-location recommendation.

4.4 TA-TEM model

To integrate the above three kinds of temporal factors, we define the objective function as follows

$$\sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}^u} \frac{1}{N_t} \sum_{j=1}^{N_t} \log \Pr\Big(\ell_j | \ell_{j-K} : \ell_{j+K}, u, u_m, d_t, h_j\Big), \tag{7}$$

where $\mathcal{U}$ is the set of users, $\mathcal{T}^u$ is the set of trajectories generated by user u and N_t is the length of trajectory t. Each kind of temporal factor is modeled as a unique embedding vector: $\boldsymbol{v}_u$ and $\boldsymbol{v}_{u_m}$ are the embedding vectors corresponding to the preference for u and the m-th persona of u, respectively, $\boldsymbol{v}_h$ and $\boldsymbol{v}_d$ correspond to the daily and weekly preference respectively, and $\boldsymbol{v}_\ell$ corresponds to location ℓ.

Given a target location ℓ_j, the additivity assumption [6,13] can be applied to multiple kinds of contextual information. Then we derive its final embedding vector by integrating Eqs. 2, 5 and 6 as follows

$$\overline{\boldsymbol{v}}_{\ell_j} = \frac{1}{5}\left\{\left(\frac{1}{2K} \sum_{-K \le k \le K, k \ne 0} \boldsymbol{v}_{\ell_{j+k}}\right) + \left(\boldsymbol{v}_u + \boldsymbol{v}_{u_m} + \boldsymbol{v}_{h_j} + \boldsymbol{v}_{d_t}\right)\right\}. \tag{8}$$

This is essentially the simple average aggregation method following [13]. There can be other aggregation methods to integrate different kinds of context information, e.g., max pooling. In our work, we use the average aggregation method for its simplicity and effectiveness. Exploiting other aggregation methods is part of our future work.

Finally, we plug the above embedding vector $\overline{\boldsymbol{v}}_{\ell_j}$ into Eqs. 4 and 3 to obtain the objective function for a single trajectory.

Ranking Locations for Recommendation. Once all embedding vectors are learnt, we generate next-location recommendation as follows. Given the previous K check-ins $\ell_1, \ell_2, \ldots, \ell_K$ of user u and the next time stamp s_{K+1} (corresponding to month m, day d, and hour h), we use the function to rank a candidate $\ell \in \mathcal{L}$:

$$S(u, \ell, s_{K+1}) \propto \left(\boldsymbol{v}_u + \boldsymbol{v}_{u_m} + \boldsymbol{v}_h + \boldsymbol{v}_d + \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{v}_{\ell_i}\right)^\top \cdot \boldsymbol{v}_\ell. \tag{9}$$

For next-basket recommendation, similar recommendation formula can be used by setting different h's within the time period $[s_N, s_N + \gamma]$

$$S(u, \ell, [s_N, s_N + \gamma]) \propto \left(\boldsymbol{v}_u + \boldsymbol{v}_{u_m} + \frac{1}{\gamma} \sum_{h \in [s_N, s_N+\gamma]} \boldsymbol{v}_h + \boldsymbol{v}_d + \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{v}_{\ell_i}\right)^\top \cdot \boldsymbol{v}_\ell. \tag{10}$$

Note that in training both the previous K and successive K locations are used, while in recommendation only the previous K locations can be used. Such a setting is consistent with real data essence, where we assume that trajectory sequences can be entirely obtained in training but only the preceding contexts can be used.

4.5 Parameter learning

In TA-TEM, the parameters to learn are the embedding vectors for various contextual factors: $\{\boldsymbol{v}_u\}, \{\boldsymbol{v}_{u_m}\}, \{\boldsymbol{v}_h\}, \{\boldsymbol{v}_d\}$ and $\{\boldsymbol{v}_\ell\}$. Unlike [13], here, we do not discriminate between the emitting vectors and contextual vectors for the locations for simplicity.

For parameter learning, TA-TEM needs to maximize the log probability defined in Eq. 7 over the trajectories of all users. However, directly optimizing this objective function is impractical because the cost of computing the full softmax for the multi-classifier to predict the current location is extremely high. To address this difficulty, we borrow the idea from the recently proposed *hierarchical softmax* [16] to approximate the full softmax. The hierarchial softmax uses a binary tree representation for every location as its leaves, and each node is explicitly associated with an embedding vector for computing the relative probability to take one branch. Each leaf can be reached by an appropriate path from the root of the tree. In this way, instead of evaluating all the $|\mathcal{L}|$ output nodes to obtain the probability distribution, only about $\log_2(|\mathcal{L}|)$ nodes need to be evaluated.

All parameters are trained by using the stochastic gradient descent (SGD) method. During the training, the algorithm iterates over the locations of all trajectories. At each time, a target location ℓ_j with its context window is used for update. After computing the hierarchical softmax, the error gradient is obtained via backpropagation and we use the gradient to update the parameters in our model. Readers can refer to [29] for the details of gradient derivation and update. Given the vector size of VS, the time complexity for an iteration is $\mathcal{O}(N \cdot VS \cdot VL \cdot \log|\mathcal{L}|)$, where N is the total number of check-ins considered in an iteration and the log terms comes from the binary tree to determine the target location from the $|\mathcal{L}|$ locations in the dataset.

5 Experiments

In this section, we construct the experiments to evaluate the performance of our proposed model TA-TEM and compare it with baseline methods.

5.1 Experimental setting

In this subsection, we present the detailed experimental setting.

5.1.1 Test collection

We use the datasets in Table 1 to construct the test collection. We randomly split the data into training and test sets by trajectories, to avoid the correlation between training and test sets (i.e., part of a trajectory is in training set while the rest of the trajectory is in test set). In our experiments, the first 80% trajectories are used for model training and the rest 20% trajectories are used as test data.

5.1.2 Evaluation metrics

For next-location recommendation, we predict the i-th ($i = 2, \ldots, N-1$) check-in location by assuming that the previous $(i-1)$ locations are known in a trajectory. On the i-th time stamp (i.e., the i-th recommendation), only the actually visited location on this time stamp is considered as a positive case (i.e., ground-truth location). For a recommendation, it will be less meaningful to recommend locations far away from the current location. Hence, we select 5000 nearest locations from the current location as the candidate samples.[2] When the ground-truth location is not included in the candidate samples, we add it into the candidate samples. Then, we rank the candidate locations based on Eq. 9.

Following [25], Hit@k is used to measure the ratio of the number that a ground-truth location can be found in top-k recommendation list. Let n denote the rank of the ground-truth location at each recommendation, the optimal result corresponds to the case where it precedes all the negative locations, i.e., $n = 1$; If $n < k$ we have a hit w.r.t k, (i.e., the ground-truth location can be found in top k ranked positions). Otherwise, we have a miss. The probability of a hit increases with the value of k increasing. The overall Hit@k is the ratio of the number of hits in all the test check-ins:

$$Hit@k = \frac{\#hit\ w.r.t\ k}{\#all\_test\_cases}.$$

For next-basket recommendation, we predict the check-in locations within a time period $[s_i, s_i + \gamma]$ by assuming that the previous i locations ($i = 1, \ldots, N - 1$) are known in a trajectory, where s_i is the time stamp for the i-th check-in and γ is the length of the time period (set to 3 h as in [4]). Following [2,4], two widely used metrics Precision@k and Recall@k (denoted by Pre@k and Rec@k) are adopted. The Pre@k measures how many top-k recommended locations correspond to the ground-truth locations and the Rec@k measures how many ground-truth locations are returned as top-k recommended locations:

$$Pre@k = \frac{tp_u}{tp_u + fp_u}$$
$$Rec@k = \frac{tp_u}{tp_u + tn_u}$$

In the above equations, tp_u is the number of locations contained in both the ground-truth location set and the top-k results produced by algorithms; fp_u is the number of locations in the top-k results by algorithms but not in the ground-truth location set; and tp_u is the number of locations contained in the ground-truth location set but not in the top-k results.

5.1.3 Methods evaluated

We evaluated the following methods for performance comparison.

- PRME [4]: the personalized ranking metric embedding, which projects the distance of locations and user-location pairs into two different latent spaces.
- TA-PRME: the time-aware variant of PRME [4], which splits the training set and trains PRME by months to capture user preference changes.
- FPMC-LR [2]: the state-of-the-art method for next-location recommendation, which augments personalized Markov chain model [18] with the localized region constraint.
- HRM [20]: the recently proposed method for next-basket (purchase) recommendation, which considers both static user preference and sequential context.
- Rank-GeoFM [8]: the state-of-the-art method for time-aware location recommendation, which considers the ranking of locations as a set of pairwise classification problems and assigns more weights to top positions.

[2] To check the recall of the ground-truth location using the above candidate generation method, we compute the hit ratios of the ground-truth location among the 5000 nearest locations on the three datasets as follows: 93, 96 and 98%. It can be seen the majority of the ground-truth locations were recalled using 5000 nearest locations.

- TA-CF [26]: the time-aware location recommendation, which extends user-based CF with a linear combination of temporal and geographical effects.
- TA-TEM: our proposed method, which uses the ranking function in Eqs. 9 and 10 for next-location and next-basket recommendation, respectively.

Among these methods, PRME, FPMC-LR, HRM do not exploit temporal effect while Rank-GeoFM, TA-CF do not exploit sequential effect. TA-PRME considers both temporal and sequential effects. For each user, we mask off her 10% training data as the development set to tune the parameters in all methods. All the baselines are obtained from the original implementations.

The parameter settings for each method are given below: Following [4], in PRMC and TA-PRMC, the time window is set at 3 h and the regulation term $\lambda = 0.3$. Following [2], in FPMC-LR, the time window is set to 3 h and regulation term $\lambda = 0.3$ and side length $d = 40$ km. Following [20], in HRM, max pooling aggregation is used for both levels. Following [8], in Rank-GeoFM, the margin parameter to tolerate missing values $\epsilon = 0.3$, the regulation term $C = 1.0$, geographical effect parameter $k = 300$ and the importance of temporal effect parameter $\beta = 0.2$. Following [26], in TA-CF, the decaying factor and time slot length are set to 1 and 1 h, and the weights for temporal and geographical influence are set to 0.5. For TA-TEM, we adopt the hierarchical softmax algorithm and the vector size and context window length are set to 300 and 8, respectively. We set a small learning rate $\eta = 0.0001$ in our experiments.

5.2 Performance evaluation

Figure 4 shows the results of Hit@k for next-location recommendation with the value of k in {5, 10, 20}. From Fig. 4, we make the following observations: (1) TA-TEM is consistently better than all the baselines; (2) TA-PRME, the time-aware variant of PRME, is worse on Foursquare dataset while better on the other datasets than PRME. TA-PRME is composed of multiple PRME components, and it generally requires more training data; thus, it performs well on the two large datasets (the best baseline on this dataset) but not on the small Foursquare dataset; (3) TA-CF does not perform well since it only models temporal preference and user preference, and it cannot capture sequential visiting behavior, which is important to consider for next-location recommendation; (4) HRM and Rank-GeoMF are two competitive baselines, and they have similar performance on three datasets.

Figure 3 further shows the results of Rec@k and Pre@k for next-basket recommendation. Similar observations can be made as in Fig. 4. Our proposed TA-TEM outperforms all baseline methods.

Among all the baselines, the most relevant method to TA-TEM is HRM, which also uses the distributed representation method for location recommendation.[3] HRM does not consider temporal contexts, although performing better than time-aware recommendation method TA-PRME on small Foursquare dataset; it is worse than TA-PRME on the large Gowalla dataset, which indicates the importance of temporal contexts. TA-PRME trains PRME components by fixed-length time periods, which reduces the amount of available training data for each PRME component. In TA-TEM, the general user embedding vector is trained using all the data; only the dynamic user preference is trained using the data in each time period. Our model is built on the distributed representation method trained with longer sequential context using dense vectors, which are more effective to alleviate data sparsity. As we can see, both HRM and

[3] It was originally proposed for next-basket recommendation in shopping, and we have slightly modified it to adapt to the current two tasks.

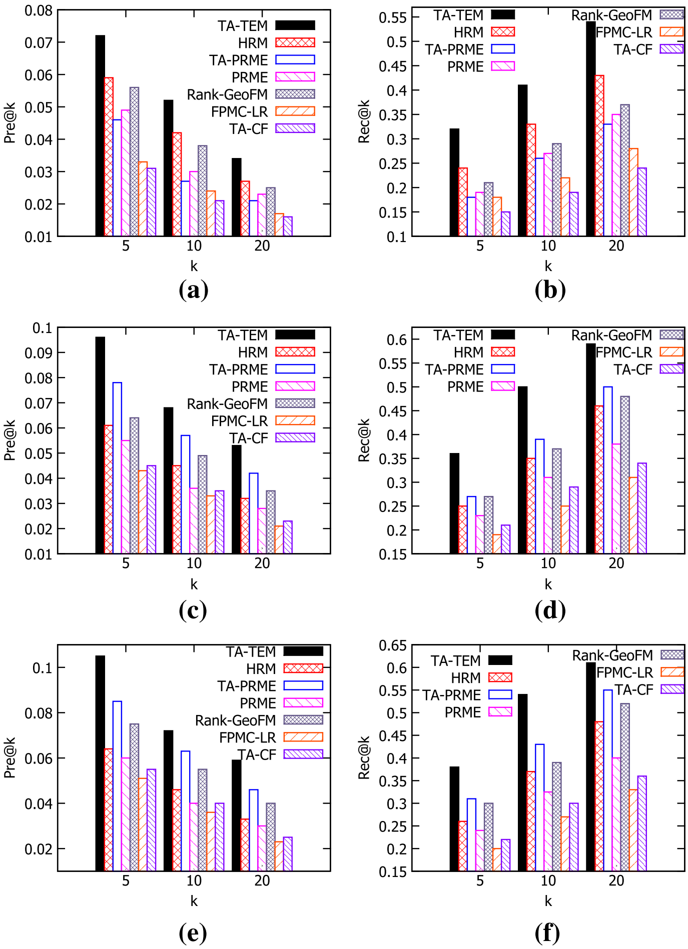

Fig. 3 Performance comparison for next-basket recommendation. The improvements of our method over all the baselines are statistically significant in terms of paired t test with p value < 0.01. **a** Pre@k on Foursquare, **b** Rec@k on Foursquare, **c** Pre@k on Gowalla, **d** Rec@k on Gowalla, **e** Pre@k on Foursquare$_l$, **f** Rec@k on Foursquare$_l$

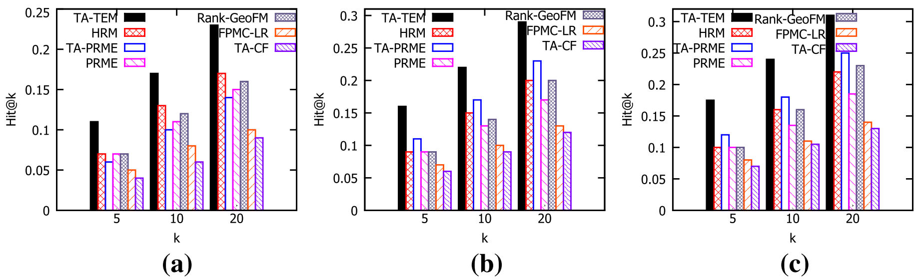

Fig. 4 Performance comparison for next-location recommendation. The improvements of our method over all the baselines are statistically significant in terms of paired t test with p value < 0.01. **a** Foursquare, **b** Gowalla, **c** Foursquare$_{large}$

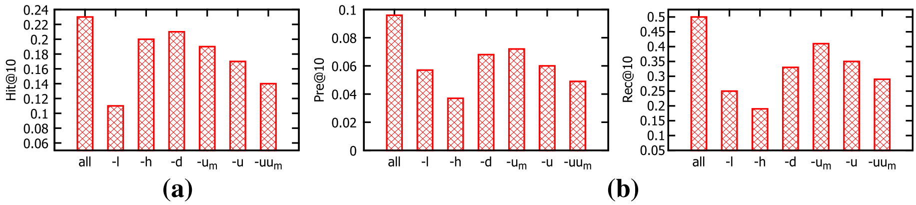

Fig. 5 Effect of contexts on Gowalla dataset (l, h, d, u_m and u denote the contexts of location, hour, day, dynamic user preference and general user preference, respectively. "–" indicates that the corresponding kind of contexts is removed). **a** Next-location recommendation, **b** next-basket recommendation

TA-TEM are less sensitive to the size of training data, and they perform relatively stable when the training size varies. Based on this merit from distributed representation, our proposed TA-TEM characterizes both dynamic personal preference (*Observation* 1) and periodical temporal preference (*Observation* 2), which leads to the best performance on both datasets.

5.3 Detailed analysis of TA-TEM

In this subsection, we present the detailed analysis of our model TA-TEM.

5.3.1 Impact of temporal factors

To measure the effect of different temporal factors, we examine the performance of the proposed method TA-TEM by excluding each feature individually. Figure 5 presents the effect of different contexts for next-location and next-basket recommendations, respectively. For next-location recommendation, as shown in Fig. 5a, all the considered contexts are helpful. Among them, the contexts for user preference (e.g., $-(u + u_m)$) are important consider because our task is essentially a personalized recommendation problem. Recall that in Eqs. 9 and 10, we have included the previous K locations for the successive location recommendation(s). Accordingly, location context is the most important factor to improve the system performance, and it shows that the next location heavily depends on the previous locations. Reported in [13], distributed embedding models are very powerful in capturing the sequential relatedness, e.g., semantic relatedness between words. Our finding further confirms this, indicating that such models are also effective to model check-in sequence data.

Figure 5b shows similar trends to Fig. 5a. The only difference is that the contexts related to temporal factors (i.e., $-(h)$) are more important than location context (i.e., $-(l)$) for next-basket recommendation. In next-location recommendation, we focus on a single location for the next recommendation, while in next-basket recommendation usually contains many successive locations in a short period. It shows that using (preceding) location contexts is more effective to identify a single next location, but less effective to capture a set of successive locations.

5.3.2 Parameter tuning

TA-TEM has two important parameters, i.e., the vector size VS and context length $CL(=2K)$. We have empirically found $VS = 300$ and $CL = 8$ lead to a good performance by using a development set. We tune vector size VS and context length CL. By reusing the previous settings for vector size VS and context length CL, we tune each with the other fixed. Figure 6a,

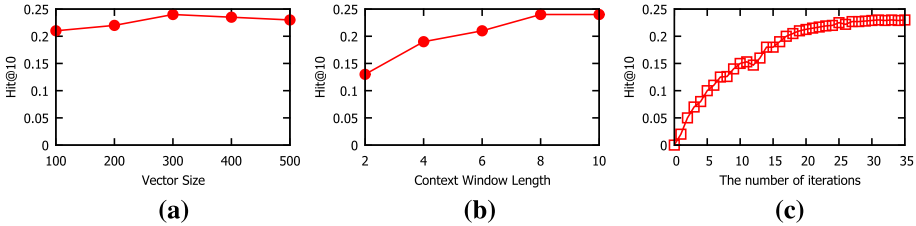

Fig. 6 Parameter tuning for next-location recommendation on Gowalla dataset. **a** Varing VS ($CL = 8$), **b** varying CL ($VS = 300$), **c** varying *iter*

b shows the tuning results on next-location recommendation. The results show that (1) TA-TEM is relatively stable when vector size $VS \in [200, 350]$ and achieves the optimal when vector size $VS = 300$ and (2) the performance improvement becomes less significant when context length $CL > 8$, which indicates that context length $CL = 8$ is sufficient for good performance. It is interesting to see that the performance obtained with $CL = 8$ is much better than that with $CL = 2$, suggesting that a relatively long context windows length is needed. Our tuning results are indeed similar to those in word embedding, where more details can be found in [5].

Our learning method with hierarchical softmax is an iterative algorithm. We finally check how the performance varies with the iteration number, i.e., $iter$. Figure 6 shows that an iteration number of 30 gives good performance, and the performance is relatively stable when $iter \geq 30$. Overall, our learning algorithm has fast convergence speed in practice.

The findings on the next-basket location recommendation task is similar, which is omitted.

5.3.3 Impact of sequential modeling

As a difference, we adopt the distributed representation method to characterize a longer sequential-context window than the personalized Markov chain model [2,18] (denoted by *FPMC-LR*). Here, we examine whether our formulation of sequential contexts can yield better performance or not compared to the method *FPMC-LR*. We prepare four variants with TA-TEM: (1) only using location contexts in a long window ($CL = 8$), denoted by $T_{l,L}$; (2) only using location contexts in a short window ($CL = 2$), denoted by $T_{l,S}$; (3) using location and (static) user contexts in a long window ($CL = 8$), denoted by $T_{l+u,L}$; (4) using location and (static) user contexts in a short window ($CL = 2$), denoted by $T_{l+u,S}$. The last two variants incorporate user contexts due to the fact that FPMC also exploits user contexts. In TA-TEM, setting $CL = 2$ gives a similar effect of characterizing first-order transitions in FPMC-LR. Table 3 shows the comparison of our variants against FPMC-LR. Consistent with the finding in Fig. 6b, we can observe that a long window size yields a much better performance in TA-TEM. Our variant $T_{l,L}$ outperforms FPMC-LR even without user contexts. The findings show the effectiveness of modeling longer sequential context with the distributed representation.

5.3.4 Efficiency analysis

In this part, we analyze the efficiency of the learning algorithm for our proposed TA-TEM with *hierarchical softmax*. Our TA-TEM model is implemented with the open source toolkit

Table 3 Comparisons of sequential modeling on next-location recommendation

Datasets	*Hit*@10				
	$\mathrm{T}_{l,S}$	$\mathrm{T}_{l,L}$	$\mathrm{T}_{l+u,S}$	$\mathrm{T}_{l+u,L}$	FPMC-LR
Foursquare	0.05	0.09	0.08	0.11	0.07
Gowalla	0.08	0.12	0.11	0.15	0.10

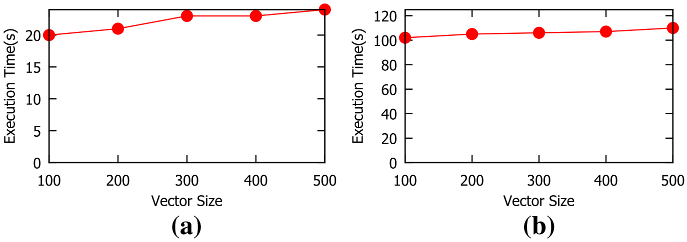

Fig. 7 Average per-iteration training time of TA-TEM with hierarchical softmax (in s). **a** Foursquare dataset, **b** Gowalla dataset

`Gensim`[4] in Python. We run the program (using six active threads simultaneously) in the server with Intel(R) Xeon(R) CPU E7-4830 v3 2.1 GHz with 24 cores in Ubuntu 14.04 LTS.

The time complexity includes both training cost and test cost. As shown in Sect. 4.5, the time complexity for per-iteration training is $\mathcal{O}(N \cdot VS \cdot VL \cdot \log |\mathcal{L}|)$, where N is the total number of check-ins considered in an iteration and the log term comes from the binary tree to determine the target location from the $|\mathcal{L}|$ locations in the dataset. When the dataset is given, VS and VL are two parameters affecting the algorithm efficiency. As shown in Fig. 6b, VL is typically set to a relatively small value, e.g., $VL \leq 10$. Hence, we only study how the time complexity varies with the increasing of VS. We report the average training time of an iteration in Fig. 7: The running time is relatively stable and slowly increases.[5]

During test, the cost for recommending to a user using Equation 9 is about $\mathcal{O}(N' \cdot \log N' \cdot C)$, where N' is the number of candidate locations for recommendation, and C is time cost for the evaluation of a candidate location, including $(CL+7) \times VS$ additions, CL divisions and VS multiplications. Compared to training, the time cost for test is small: TA-TEM can produce recommendations for about 10,000 users per second on Gowalla dataset when $N' = 5000$. It shows our algorithm can be easily extended for efficient online recommendation after offline training.

5.3.5 Qualitative analysis with embedding vectors

A major merit of our proposed method is that various kinds of contextual information are projected into the same low-dimensional space, in which the association between different context features can be measured by using a simple cosine similarity. Table 4 presents several examples on context association. In each example, a *query context* is given and then the top five associated locations are obtained. We can observe that (1) hour and day contexts identify locations with similar temporal characteristics, e.g., traffic-based locations for morning while relax-based locations for weekends; (2) the user dynamic preference (i.e., "USER1$_{\texttt{Dec}}$." vs.

[4] https://radimrehurek.com/gensim/index.html.

[5] Due to multi-threading techniques, the time cost does not show a strict linear increase with the increasing of VS.

Table 4 Sample examples for query context with the corresponding top five related locations on the Gowalla dataset

Type	Query context	Top five related locations
Hour	9 pm	Coffee Shop$_{\text{food}}$, McDonalds$_{\text{food}}$, Light Rail$_{\text{travel}}$, UBS$_{\text{office}}$, Luxury Hotel$_{\text{house}}$
	6 am	Employ$_{\text{office}}$, Baron$_{\text{food}}$, Vaksal$_{\text{home}}$, East Train$_{\text{travel}}$, Square Plaza$_{\text{shop}}$
Day	Monday	Library$_{\text{college}}$, Metro$_{\text{travel}}$, UBS$_{\text{office}}$, Coffee Shop$_{\text{food}}$, Gym$_{\text{sport}}$
	Sunday	Walmart$_{\text{shop}}$, Museum$_{\text{art}}$, Theater$_{\text{movie}}$, Nature Preserve$_{\text{relax}}$, Bookstore$_{\text{shop}}$
$u_m/u + l$	USER1$_{\texttt{Aug.}}$ + Airport	Komos Beach$_{\text{relax}}$, Greenpark Hotel$_{\text{hotel}}$, Lake Lenexa$_{\text{relax}}$, Rental Car$_{\text{travel}}$, Fish Bar$_{\text{food}}$
	USER1$_{\texttt{Dec.}}$ + Airport	Ski Area$_{\text{relax}}$, Ski shop$_{\text{shop}}$, Cameron's Pub$_{\text{food}}$, The Grosvenor$_{\text{hotel}}$, Snow Bus$_{\text{travel}}$
	USER1 + Airport	Starbucks$_{\text{food}}$, Gate$_{\text{travel}}$, Duty Free Shops$_{\text{shop}}$, Bus Station$_{\text{travel}}$, Modern Hotel$_{\text{house}}$

The attached label for each location is marked in subscripts. "USER1" is a randomly selected (anonymous) user for illustration

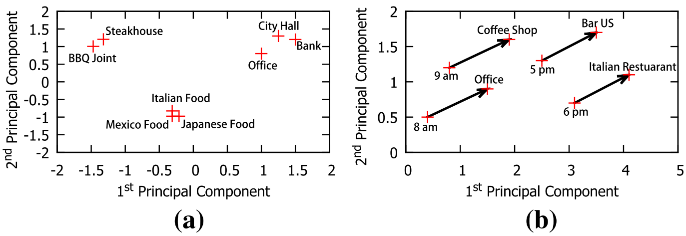

Fig. 8 Visualization examples with the learned vectors using principal component analysis. **a** Location vectors, **b** location versus temporal vectors

"USER1$_{\texttt{Aug.}}$") leads to meaningful recommendation results. For example, given the previous location labeled as "Airport", the same user "USER1" has shown different visiting behaviors, visiting ski-related places in winter while beach-related locations in summer. Interestingly, as a comparison, the general user preference (i.e., "USER1") tends to capture her common visiting behaviors in travel, such as shopping in "Duty Free Shop."

In order to obtain a better understanding of the learned vectors, we further present two visualization examples. The learned vectors are projected into a two-dimensional space using principal component analysis (PCA). In Fig. 8a, it can be observed that locations with similar functionality are indeed projected into close positions in the latent space (e.g., office-based locations are clustered in the top corner). Figure 8b further presents the relation between temporal vectors and location vectors. It is interesting to see each location vector is indeed related to some specific temporal vector.

6 Conclusion

In this paper, we study the task of time-aware next location recommendation. We develop a novel Time-Aware Trajectory Embedding Model (TA-TEM) to jointly characterize sequential patterns, general temporal patterns and dynamic user preference. These three factors are characterized by using the distributed learning methods, i.e., embedding vectors. Such a representation method is flexible to integrate multiple kinds of contextual information in a unified way and is more resistant to data sparsity. Extensive experiments on two real-world geo-based datasets have demonstrated the effectiveness of our proposed method. Our work presents a simple yet effective approach for next-location recommendation, which sets up a new direction. Following this direction, more complicated distributed representation methods can be developed to improve the next-location recommendation task.

Currently, our model is trained in an offline way. As future work, we will consider how to develop the efficient way to train our models in an online way. Our work also sheds light on the application of distributed representation models to trajectory data. In future, we will explore more advanced deep learning models such as deep convolutional neural networks for modeling trajectory data.

Acknowledgements The authors thank the anonymous reviewers for their valuable and constructive comments. The work was partially supported by National Natural Science Foundation of China under the Grant Number 61502502, Beijing Natural Science Foundation under the Grant Number 4162032, the National Key Basic Research Program (973 Program) of China under Grant No. 2014CB340403, and the open fund with the Grant Number MJUKF201703 from Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University).

References

1. Altman RM (2007) Mixed hidden markov models: an extension of the hidden markov model to the longitudinal data setting. J Am Stat Assoc 102(477):201–210
2. Cheng C, Yang H, Lyu MR, King I (2013) Where you like to go next: successive point-of-interest recommendation. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013, pp 2605–2611
3. Cheng H, Ye J, Zhu Z (2013) What's your next move: user activity prediction in location-based social networks. SDM 13:171–179
4. Feng S, Li X, Zeng Y, Cong G, Chee YM, Yuan Q (2015) Personalized ranking metric embedding for next new POI recommendation. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015, pp 2069–2075
5. Lai S, Liu K, He S, Zhao J (2016) How to generate a good word embedding. IEEE Intell Syst 31(6):5–14
6. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31th international conference on machine learning, ICML 2014, Beijing, China, 21–26 June 2014, pp 1188–1196
7. Levandoski JJ, Sarwat M, Eldawy A, Mokbel MF (2012) Lars: a location-aware recommender system. In: ICDE '12, pp 450–461
8. Li X, Cong G, Li X-L, Pham T-AN, Krishnaswamy S (2015) Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In: SIGIR'15, pp 433–442
9. Li Y, Nie J, Zhang Y, Wang B, Yan B, Weng F (2010) Contextual recommendation based on text mining. In: COLING'10, pp 692–700
10. Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: KDD'14, pp 831–840
11. Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: KDD'13, pp 1043–1051
12. Liu X, Liu Y, Aberer K, Miao C (2013) Personalized point-of-interest recommendation by mining users' preference transition. In: CIKM '13, pp 733–738
13. Mikolov T, Chen K, Corrado GS, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of workshop at ICLR
14. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS '13, pp 3111–3119
15. Mimno D, McCallum A (2007) Expertise modeling for matching papers with reviewers. In: KDD '07, pp 500–509

16. Morin F, Bengio Y (2005) Hierarchical probabilistic neural network language model. In: Proceedings of AISTATS, pp 246–252
17. Noulas A, Scellato S, Lathia N, Mascolo C (2012) A random walk around the city: new venue recommendation in location-based social networks. In: Proceedings of PASSAT, pp 144–153
18. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation. In: WWW '10, pp 811–820
19. Usunier N, Buffoni D, Gallinari P (2009) Ranking with ordered weighted pairwise classification. In: ICML '09, pp 1057–1064
20. Wang P, Guo J, Lan Y, Xu J, Wan S, Cheng X (2015) Learning hierarchical representation model for nextbasket recommendation. In: SIGIR '15, pp 403–412
21. Wu X, Liu Q, Chen E, He L, Lv J, Cao C, Hu G (2013) Personalized next-song recommendation in online karaokes. In: RecSys '13, pp 137–140
22. Yang D, Zhang D, Bingqing Q (2016) Participatory cultural mapping based on collective behavior data in location-based social networks. ACM TIST 7(3):30
23. Ye M, Liu X, Lee W-C (2012) Exploring social influence for recommendation: a generative model approach. In: SIGIR '12, pp 671–680
24. Ye M, Yin E, Lee W-C, Lee D-L (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: SIGIR '11, pp 325–334
25. Yin H, Sun Y, Cui B, Hu Z, Chen L (2013) Lcars: a location-content-aware recommender system. In: KDD '13, pp 221–229
26. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: SIGIR '13, pp 363–372
27. Zhao WX, Jiang J, He J, Shan D, Yan H, Li X (2010) Context modeling for ranking and tagging bursty features in text streams. In: Proceedings of the 19th ACM conference on information and knowledge management, CIKM 2010, Toronto, Ontario, Canada, 26–30 October 2010, pp 1769–1772
28. Zheng VW, Cao B, Zheng Y, Xie X, Yang Q (2010) Collaborative filtering meets mobile recommendation: a user-centered approach. In: AAAI '10
29. Zhou N, Zhao WX, Zhang X, Wen JR, Wang S (2016) A general multi-context embedding model for mining human trajectory data. IEEE Trans Knowl Data Eng 28(8):1945–1958

Wayne Xin Zhao received Ph.D. degree from Peking University in 2014. He is currently an associate professor in the School of Information, Renmin University of China. His research interests are web text mining and natural language processing. He has published several referred papers in international conferences and in journals such as ACL, EMNLP, COLING, ECIR, CIKM, SIGIR, SIGKDD, AAAI, IJCAI, ACM TOIS, ACM TKDD, ACM TIST and IEEE TKDE.

Ningnan Zhou received BEng degree in computer science from Renmin University of China, China, in 2012. He is currently working toward Ph.D. degree in School of Information, Renmin University of China. His research mainly focuses on web mining and database. He is a student member of the IEEE.

Aixin Sun is an Associate Professor in School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received Ph.D. from the same school in 2004. His research interests include information retrieval, text mining, social computing, and multimedia. His papers appear in major international conferences like SIGIR, KDD, WSDM, ACM Multimedia, and journals including DMKD, TKDE, and JASIST.

Ji-Rong Wen is a Professor and the Dean in the School of Information at Renmin University of China (RUC). He was awarded the prestigious National 1000 Plan Expert of China. He is the director of Big Data Research Centre and is leading Big Data Analytics and Intelligence Lab in RUC. His main research interest lies in big data management, information retrieval (especially web search), data mining and machine learning.

Jialong Han is a postdoctoral research fellow at School of Computer Science and Engineering, Nanyang Technological University. He earned his Ph.D. degree from Renmin University of China in 2015, under the supervision of Prof. Ji-Rong Wen. He obtained his B.E. degree also from Renmin University of China in 2010. His research interests include graph data mining and management, as well as their applications on knowledge graphs.

Edward Y. Chang is the President of HTC Research & Healthcare, heading software and hardware future technology research and development. Prior to his HTC post, Ed was a director of Google Research and was a full professor of Electrical Engineering at the University of California, Santa Barbara (UCSB). Ed has served on ACM (SIGMOD, KDD, MM, CIKM), VLDB, IEEE, WWW, and SIAM conference program committees and co-chaired several conferences including MMM, ACM MM, ICDE, and WWW. He is a recipient of the NSF Career Award, IBM Faculty Partnership Award, and Google Innovation Award.