World Wide Web
https://doi.org/10.1007/s11280-018-0596-8

Time-aware metric embedding with asymmetric projection for successive POI recommendation

Haochao Ying[1] · Jian Wu[1] · Guandong Xu[2] · Yanchi Liu[3] · Tingting Liang[1] · Xiao Zhang[4] · Hui Xiong[3]

Received: 26 February 2018 / Revised: 21 April 2018 / Accepted: 24 May 2018

Abstract Successive Point-of-Interest (POI) recommendation aims to recommend next POIs for a given user based on this user's current location. Indeed, with the rapid growth of Location-based Social Networks (LBSNs), successive POI recommendation has become an important and challenging task, since it can help to meet users' dynamic interests based on

This work was finished when Haochao Ying visited University of Technology Sydney.

This article belongs to the Topical Collection: *Special Issue on Big Data Management and Intelligent Analytics*
Guest Editors: Junping Du, Panos Kalnis, Wenling Li, and Shuo Shang

✉ Guandong Xu
Guandong.xu@uts.edu.au

Haochao Ying
haochaoying@zju.edu.cn

Jian Wu
wujian2000@zju.edu.cn

Yanchi Liu
yanchi.liu@rutgers.edu

Tingting Liang
liangtt@zju.edu.cn

Xiao Zhang
tobexiao1@dislab.nju.edu.cn

Hui Xiong
hxiong@rutgers.edu

[1] College of Computer Science & Technology, Zhejiang University, Hangzhou, China

[2] Advanced Analytics Institute, University of Technology Sydney, Sydney, Australia

[3] Management Science & Information Systems, Rutgers University, Newark, NJ, USA

[4] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Published online: 09 June 2018

their recent check-in behaviors. While some efforts have been made for this task, most of them do not capture the following properties: 1) The transition between consecutive POIs in user check-in sequences presents asymmetric property, however existing approaches usually assume the forward and backward transition probabilities between a POI pair are symmetric. 2) Users usually prefer different successive POIs at different time, but most existing studies do not consider this dynamic factor. To this end, in this paper, we propose a time-aware metric embedding approach with asymmetric projection (referred to as MEAP-T) for successive POI recommendation, which takes the above two properties into consideration. In addition, we exploit three latent Euclidean spaces to project the POI-POI, POI-user, and POI-time relationships. Finally, the experimental results on two real-world datasets show MEAP-T outperforms the state-of-the-art methods in terms of both precision and recall.

Keywords Successive POI recommendation · Metric embedding · Asymmetric projection · Temporal influence

1 Introduction

The increasing prevalence of smart mobile devices and the successful development of Location-based Social Networks (LBSNs), such as Gowalla, Foursquare, and Facebook Places, have greatly enhanced the life experience of users [11, 19, 33, 34]. In these platforms, users can check-in at Point-of-Interests (POIs) to show where and when they are, and share their personal experiences with others through comments. Taking Foursquare as an example, more than 10 billion check-ins have been generated by over 50 million users.[1] With such a huge amount of check-in data, how to mine user preferences and recommend right POIs to right users has become an interesting topic, which helps users to explore interesting places and facilitate service providers to launch advertisements to potential target users. This task, known as POI recommendation, has attracted lots of efforts with various recommendation methods being proposed [8, 9, 13, 24, 29].

As the easy collection of user context information (e.g., spatial and temporal information) under the mobile enviroment, successive POI recommendation, which recommends next POIs given a user and his/her current location, has become more pratical and emerging problem [2, 6, 14]. Figure 1 gives an intuitive example. We can observe that it is more rational to recommed recreation venue rather than fitness after user has a dinner. Further, we can analyze where the event (e.g., stampede and traffic jam) will happen in advance if we can predict the next POIs of users [20–22]. However this task is harder than traditional POI recommendation due to following reasons. First, although the interactions between users and POIs are very sparse, the successive check-in interactions are even sparser since one query (user, current location) may have tens of thousands of next candidate POIs. Second, the next POI is largely dependent on the current POI in addition to user preferences. For example, it is easy to imagine that users would prefer a dinner than shopping after hiking or other outdoor activities. Therefore, how to deal with the high sparsity and sequential information is the key to the success of successive POI recommendation.

Recently, approaches have been proposed for successive POI recommendation by tackling the above challenges [2, 4, 16]. For example, Feng et al. [4] attempt to use metric

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

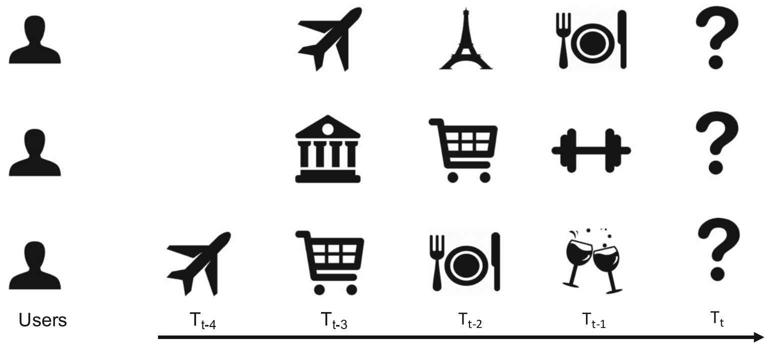


Figure 1 An example of users' check-in sequences

embedding to better model the highly sparse sequential transitions. To be more specific, they project each POI to a point in a low-dimensional latent Euclidean space rather than an independent vector in matrix factorization. However, several properties of check-in behaviors are not considered here: 1) **Asymmetric property.** They assume a consecutive POI-POI transition is intrinsically symmetric, which means reverting previous and next POIs in the latent space will get the same Euclidean distances. However, this assumption does not consistently hold because the consecutive check-in data usually exhibits sequential order (i.e., asymmetric property). For instance, the chance of coffee-office sequence is higher than that of office-coffee. We will present more details in Section 3 to demonstrate this property. 2) **Temporal property.** Check-in data exhibits temporal patterns, e.g., time-aware and periodic. As we know, successive POI recommendation is a time-sensitive task, since different POIs have different popularities at different time slots. For example, restaurants are always checked-in during lunch or dinner time, while also show different popularities in weekdays and weekends, i.e., periodicity.

In this paper, to justify the above observations, we first conduct empirical analysis on two public real-world datasets. Specifically, we report successive POI check-in transition distributions from coarse-grained category level to fine-grained POI level to show the existence of asymmetry of POI transitions. Then, to address such asymmetric property, we propose a new time-aware metric embedding approach with asymmetric projection to recommending the most possible POIs for users given their current locations. In particular, two corresponding left and right matrices are designed to project the current and next POI representations into another latent space to differentiate the forward and backward transitions. Apart from asymmetry, two temporal factors, namely periodicity and time interval, are also taken into consideration. Finally, we utilize three types of latent Euclidean spaces (i.e., user preference space, sequential transition space, and time specific space) to model POI-user, POI-POI, and POI-time relationships, respectively. The strengths of all these relationships are fused into a unified way by Euclidean distance in corresponding spaces. To conclude, the main contributions of this paper lay as follows:

- We empirically verify the existence of asymmetric property in successive POI recommendation from two perspectives: category level and POI level.
- We propose a time-aware metric embedding approach with asymmetric projection to learning the representations of POIs, users, and time in latent Euclidean spaces. To model asymmetric sequence information and temporal impacts, we jointly consider

POI-POI transition, POI-user preference, and POI-time periodicity in three different latent spaces.
– We conduct experiments on two real-world datasets to evaluate the effectiveness of our proposed model. Experimental results show our model outperforms four state-of-the-art methods for successive POI recommendation.

The remainder of this paper is organized as follows. We first review the related work in Section 2. Section 3 reports our empirical analysis of check-in data. In Section 4, we introduce and elaborate our model in detail. Section 5 presents the experimental study, followed by conclusions in Section 6.

2 Related work

Comparing with traditional recommendation scenarios (e.g., movie recommendation), the task of POI recommendation faces with severer challenges: 1) The check-in data is implicit user feedback, which brings more noise for modeling user preference. 2) There are many types of contextual information to determine user check-in locations, e.g., social connections, spatial-temporal influence, POI categories, sequential information and so on. In this section, we first introduce traditional POI recommendation and how to model sequential information for this task, and then review existing work on successive POI recommendation.

Traditional point-of-interest recommendation With the rapid growth of accumulated check-in data, traditional POI recommendation, which focuses on recommending the right POIs to the right users, has received much attention in recent years [26–28]. Among the approaches proposed in previous work, matrix factorization is the most popular framework to solve this task. Lian et al. proposed the GeoMF model to seamlessly incorporate spatial clustering phenomenon into weighted matrix factorization [10]. To capture the geographical phenomenon, GeoMF augments user and POI latent factors with activity area vectors of users and influence area vectors of POIs. However, this model cannot easily integrate context information. Based on this observation, Li et al. proposed a ranking based geographical factorization method, namely Rank-GeoMF, which employs the OWPC loss metric to learn the model [9]. In particular, the authors assume that the check-in probability is determined by the interactions between users and targeted POIs, and the ones between users and the neighboring POIs of targeted POIs. In addition to incorporating factors into traditional collaborative filtering, generative graphical model is another mainstream method. Liu et al. proposed a geographical probabilistic factor analysis framework which strategically considers multiple factors, including user preferences, geographical influence, and the user mobility pattern [12]. Yin et al. joint probabilistic matrix factorization and deep learning model to solve the out-of-town and cold-start issues [29]. Moreover, some recent work starts to study sequential influence for POI recommendation [23, 35]. Wang et al. designed a sequential personalized spatial item recommendation framework which introduces a novel latent variable topic-region to learn and fuse sequential influence and personal interests in the latent and exponential space [23]. Compared with this thread, we consider how to embed sequential information for a more challenging task, i.e., successive POI recommendation.

Successive point-of-interest recommendation Sequential influence may help traditional POI recommendation to some extent, but it is a significant factor for successive POI recommendation [15]. Different from traditional POI recommendation, successive

POI recommendation needs to provide a recommendation list based on a given user's recent check-ins, which requires not only the preference modeling from users but also the correlations between POIs [32]. With the rapid rising of deep learning, some jobs have introduced them to solve this task. Liu et al. extended Recurrent Neural Network and modeled local temporal and spatial contexts in each layers [14]. In particular, they replaced the single transition matrix in original RNN with time-specific and distance-specific transition matrices. In addition, most of previous studies employed Markov chain property to model POI-POI transition [2, 4, 6, 31]. Cheng et al. proposed a novel tensor factorization, namely FPMC-LR, to incorporate two observed properties: personalized Markov chains and localized regions [2]. Additionally, He et al. observed that human exhibit distinct latent transition patterns under different contextual scenarios and proposed a unified tensor-based latent model [6]. Feng et al. employed metric embedding to model sequential POI transition [4]. Recently, Zhao et.al considers successive POI recommendation is a time-subtle task and designs a time index scheme [32]. Different from previous work, our model mainly focus on POI-POI asymmetric property in this paper, while they always assume the transition is symmetric intrinsically.

3 Pattern analysis of real-world check-in datasets

Before presenting our approach in detail, we first introduce two real-world datasets used in this paper and then show some important patterns of user behaviors that will be taken into consideration in our model.

3.1 Data description

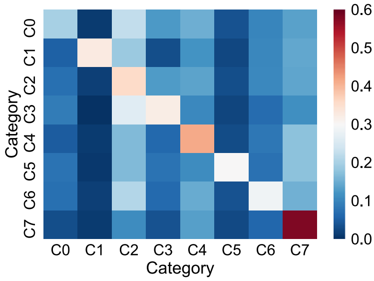
We use two publicly available check-in datasets collected from different real-world LBSN applications: one is from *Foursquare* [25], and the other is from *Gowalla* [3]. The *Foursquare* dataset includes user check-in data from April 12, 2012 to February 16, 2013 in New York City and we remove POIs which have been visited by no more than 5 users, and filter users who have checked-in no more than 10 POIs. For the *Gowalla* dataset, we first choose user check-in records in California using Bing Maps API according to POI latitude and longitude. Then we keep POIs which have been visited by more than 15 users, and choose users who have checked-in more than 20 POIs due to the higher data sparsity than *Foursquare*. Some basic statistics of the two processed datasets are summarized in Table 1.

3.2 Patterns of user behaviors

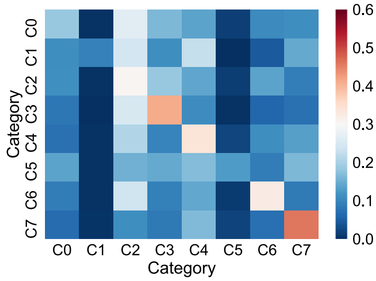
Figure 2a and b report successive POI check-in transition probabilities at category level in weekdays and weekends, respectively, where Y axis is the category of the current POI, while X axis presents the category of the next POI. Note that here we only demonstrate the category transition of *Foursquare* since *Gowalla* dataset misses the category information

Table 1 Statistics of datasets

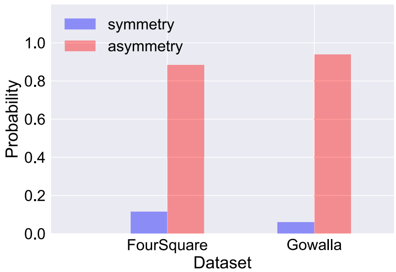
Dataset	#Users	#POIs	#Check-ins	Density	#Avg. check-ins per user
Foursquare	1078	2941	71,622	2.26%	66.44
Gowalla	2166	4047	100,986	1.15%	46.62



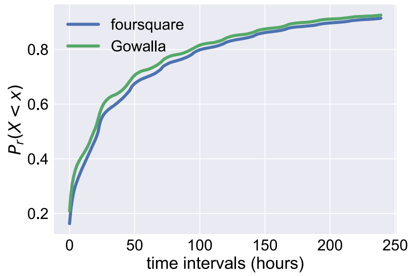
(a) weekday asymmetry pattern at category transition in *Foursquare*



(b) weekend asymmetry pattern at category transition in *Foursquare*



(c) asymmetry pattern at POI level



(d) time intervals of successive check-ins

Figure 2 Mobile patterns of user successive check-in behaviors. **a** and **b** show asymmetry property at category level (C0: Arts & Entertainment, C1: College & University, C2: Food, C3: Nightlife Spot, C4: Outdoors & Recreation, C5: Professional & Other Places, C6: Shop & Service, C7: Travel & Transport). **c** shows asymmetry property at POI level. **d** is the statistical result of time intervals of successive check-in records

of locations. Moreover, in the preprocessing, we remove 6.7% of users' check-ins whose category is home because the information is useless for our task. From these figures, we can see that category transition has asymmetric property on both weekdays and weekends. For example, the transition probability from C1 (College & University) to C4 (Outdoors & Recreation) is much higher than that from C4 to C1. This phenomenon can be easily interpreted that students may go climbing after school but it is quite rare for them to go outdoors first and then be punctual for classes. We further discuss whether successive POI check-in transition probabilities are also asymmetric at POI level, through comparing the probabilities of observing the preceding POI and the next POI given the current one. As shown in Figure 2c, the probability that preceding and next POIs of a current POI are different is much higher than they are same in both *Foursquare* and *Gowalla*. Therefore, successive check-ins exhibit asymmetric property on both category level and POI level. However, existing models usually assume the transitions of successive check-ins are symmetric. This observation triggers us to incorporate asymmetric transition into our model.

Time interval is another significant factor of successive POI recommendation. Figure 2d shows how long users check two successive check-ins by calculating the cumulative distribution function (CDF). We come to the same conclusion as in [16, 32] that consecutive check-ins have strong cooccurence rate: 30.9 and 36.6% of successive check-in records occur in less than 6 hours in *Foursquare* and *Gowalla* respectively. Meanwhile, 47.7 and 43.7% of successive check-ins happen in more than one day. Intuitively, short time intervals can better reveal why users go to the next POI from the current one. For more than 30% of consecutive check-ins, the time intervals are larger than two days in both two datasets. Therefore, a certain proportion of consecutive check-ins may not be influenced by the current POIs, which will not be considered in our model.

4 Time-aware metric embedding with asymmetric projection

In this section, we present the details of our proposed model time-aware metric embedding with asymmetric projection, referred to as MEAP-T, for successive POI recommendation. We first introduce the problem formulation, and then present our model and optimization method.

4.1 Problem formulation

Let U denote the set of users and L denote the set of locations, i.e., POIs. The check-in records of user u is represented as $L_u = \{l_u^1, \ldots, l_u^{u_t-1}\}$, where u_t is the time step when user t is going to visit the next POI and l_u^i is the POI user u checked-in at time step i ($i \in \{1, 2, \ldots, u_t - 1\}$). The goal of successive POI recommendation is to provide a set of POIs for user u at time step t, given his/her historical check-in records L_u. Inspired by the finding that in a short period of time, two successive POIs of a user exhibit strong connections [2], we employ the Markov chain framework to model sequential influence between POIs. Further, considering the complexity of nth-order Markov chain exponentially increases with n and the experimental result shows first-order chain is better than higher-order ones at *Foursquare* dataset [5], here we assume the probability of next POI only relies on the current one [2, 4, 7]. Therefore, we focus on computing the probability that user u will visit POI l given his current location l^c: $p(l|u, l^c)$.

4.2 Metric embedding with asymmetric projection

Using the first-order Markov chain to learn POI-POI transition, one simple way is to convert the successive check-in transition counts into transition probabilities and then use maximum likelihood estimation to predict the next POI for user u, which is shown as follows:

$$p(l|u, l^c) = \frac{Count(u, l, l^c)}{OCount(u, l^c)} \tag{1}$$

where $Count(u, l, l^c)$ and $OCount(u, l^c)$ denote the numbers of successive transitions from POI l to l^c in L_u and from l^c to all next POIs in L_u, respectively. However, the check-in dataset is very sparse as shown by the densities in Table 1 so it is hard to estimate p precisely.

To overcome the above issue, a further improvement can be made by producing distributed representations for POIs or users. Metric embedding model has been proven a good way to keep the coherent POI-POI or POI-user metric relationships in a latent space [1, 4]. The key assumption of metric embedding is that each relationship is reflected by the Euclidean distance through each latent low-dimensional space (i.e, d-dimensional in this paper). In particular, each user u and POI l have latent positions $X^P(u)$ and $X^P(l)$ in the user Preference space, respectively. The user-POI preference is related to the Euclidean distance $||X^P(u) - X^P(l)||_2$. Meanwhile, each POI l has a latent position $X^S(l)$ in the Sequential transition space. Similarly, the POI-POI consecutive transition probability is reflected from the Euclidean distance $||X^S(l) - X^S(l')||_2$. Note that the stronger a relationship is, the lower the corresponding Euclidean distance is. By combining these two kinds of metric relationships, the transition probability from current POI l^c to a candidate POI l for user u can be defined as follows:

$$p(l|u, l^c) = \frac{e^{-(||X^P(l)-X^P(u)||_2^2+||X^S(l)-X^S(l^c)||_2^2)}}{\sum_{j=1}^{|L|} e^{-(||X^P(l_j)-X^P(u)||_2^2+||X^S(l_j)-X^S(l^c)||_2^2)}} \tag{2}$$

The goal of successive POI recommendation is to provide a ranked POI list for a given user. Therefore, we can drop the normalization term in (2) and simplify it into a ranking task by calculating the two Euclidean distances [4]:

$$D_{u,l^c,l} = ||X^P(l) - X^P(u)||_2^2 + ||X^S(l) - X^S(l^c)||_2^2 \tag{3}$$

After learning from training data, each POI and user is projected to a point in the latent space such that unobserved transitions from l^c to l for user u are assigned meaningful values $D_{u,l^c,l}$. However, representing each POI in the sequential space with only one position will lead to flaws. For any two POIs l_i and l_j, the metric distances from l_i to l_j and from l_j to l_i are the same (i.e., $||X^S(l_i) - X^S(l_j)||_2^2 = ||X^S(l_j) - X^S(l_i)||_2^2$), which means the sequential transition is symmetric. However, recall that in Section 3, asymmetric property of successive check-ins has been demonstrated on both category and POI levels. Such symmetric property takes exactly the opposite point of view from data.

To address the above limitation, an intuitive solution is to assign two distinct representations to each POI, namely entry and exit vectors [1]. The entry vector models the transitions from previous POIs to the concerned POI, while the exit vector models the transitions from the concerned to next POIs. However, in LBSN scenarios, the data is always sparse so that it is very difficult to learn the two full representations simultaneously. Furthermore, each user only checks-in a few POIs and we cannot easily train the exit vectors for those unobserved POIs under the Bayesian Personalized Ranking (BPR) framework, which is a popular pairwise optimization method in recommender systems for implicit feedback data [17]. Hence, we propose to learn two different small projections for current and next POIs, respectively, and the distance $D_{u,l^c,l}$ is modified as follows:

$$D_{u,l^c,l} = ||X^P(l) - X^P(u)||_2^2 + ||LX^S(l) - RX^S(l^c)||_2^2 \tag{4}$$

where L and R are two $d \times d$ matrices. We project the current and next POI representations into another space with the corresponding left and right matrices to meet the asymmetric property. The advantage of using matrix projection is that it will not bring much difficulty for training in spite of data sparsity.

In addition to the asymmetric property, the temporal influence also plays an important role in user check-in behaviors. In this paper, we mainly consider the temporal influence in two-folds. (1) Periodicity [30, 32]. POIs usually have periodic check-in probabilities. For example, a bar may show a daily periodic pattern and is more likely to be visited at night, while a shopping mall may show a weekly periodic pattern and has higher probability to be visited at weekends. (2) Time interval of successive check-ins. In Section 3, we have observed that a certain proportion of consecutive check-ins exhibit long time spanning, which may indicate their irrelevance. To capture the first, we set a time-specific latent space and segment time into fixed-sized time periods. Similar to [30], after dividing each day into 6 time periods and meanwhile discriminating weekdays and weekends, the total number $|T|$ of time periods is equal to 12. After that, our model further considers Euclidean distance $||X^T(l) - X^T(t)||_2^2$ between POI l and time period t in the Time specific space. The intuition is that if a POI is always checked-in at one or a few time periods then the distance between them should be close. For the second characteristic, we assume if the time interval between two POIs is larger than a threshold τ, the influence between the two adjacent POIs vanishes [4]. Hence, our final model is specified by:

$$D_{u,l^c,t,l} = \begin{cases} ||X^P(l) - X^P(u)||_2^2 + ||X^T(l) - X^T(t)||_2^2 + \beta_l, & \text{if } \Delta(l, l^c) > \tau \\ ||X^P(l) - X^P(u)||_2^2 + ||LX^S(l) - RX^S(l^c)||_2^2 & \\ +||X^T(l) - X^T(t)||_2^2 + \beta_l, & \text{otherwise} \end{cases} \quad (5)$$

where $\Delta(l^c, l)$ represents the time interval between two consecutive POIs, and β_l is the bias of POI l. To sum up, we utilize three different latent spaces (i.e., user preference space, sequential transition space and time specific space) to model user preference, POI-POI transition and POI-time relationship, respectively. Note that the number of dimensions of all spaces is set to d for simplicity.

4.3 Model inference and learning

Our model aims to provide a ranked list of next POIs based on successive check-in probabilities given a user's current location. As we have mentioned, we care more about the ranking order of candidate POIs rather than the probabilities. Following the BPR optimization criterion in [17], we propose a pairwise ranking objective function. We assume that users prefer observed next POIs to unobserved ones and define a ranking operator $>_{u,l^c,t}$ over POIs:

$$l_i >_{u,l^c,t} l_j \Leftrightarrow D_{u,l^c,t,l_i} < D_{u,l^c,t,l_j}, \quad (6)$$

where l_i is the observed next POI at time period t given user u and current POI l^c while l_j is not observed. For each observation $< u, l^c, t, l_i >$ which means user u transfers from current POI l^c to next POI l_i at time period t, we can generate a pairwise preference order $l_i >_{u,l^c,t} l_j$ where l_j is an unobserved POI, i.e., $l_j \notin L_u$. After that, the training set $D_{train} = \{(l_i, u, l^c, t, l_j)\}$ can be obtained. We further assume the independence of the generated pairwise orders. Then we estimate our model by using maximizing a posterior (MAP) and use logistic function to approximate the likelihood of all the pairwise orders:

$$\begin{aligned} \Theta &= \arg\max_{\Theta} \log \prod_{(l_i,u,l^c,t,l_j)\in D_{train}} P(l_i >_{u,l^c,t} l_j|\Theta)P(\Theta) \\ &= \arg\max_{\Theta} \sum_{(l_i,u,l^c,t,l_j)\in D_{train}} \log P(l_i >_{u,l^c,t} l_j|\Theta)P(\Theta) \\ &= \arg\max_{\Theta} \sum_{(l_i,u,l^c,t,l_j)\in D_{train}} \log(\sigma(D_{u,l^c,t,l_j} - D_{u,l^c,t,l_i})) \\ &\quad -\lambda_P(||X^P(U)||^2 + ||X^P(L)||^2) - \lambda_S||X^S(L)||^2 \\ &\quad -\lambda_T(||X^T(L)||^2 + ||X^T(T)||^2) - \lambda_\beta||\beta(L)||^2 \\ &\quad -\lambda_A(||L||^2 + ||R||^2), \end{aligned} \quad (7)$$

where $\Theta = \{X^P(U), X^P(L), X^S(L), L, R, X^T(L), X^T(T), \beta(L)\}$ is the set of parameters, σ is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\lambda = (\lambda_P, \lambda_P, \lambda_S, \lambda_A, \lambda_A, \lambda_T, \lambda_T, \lambda_\beta)^T$ is the regularization parameter vector corresponding to Θ. We adopt stochastic gradient decent (SGD) to learn the parameters for efficiency. The update procedure is carried out as follows:

$$\Theta = \Theta + \alpha\left((1 - \sigma(z))\frac{\partial z}{\partial \Theta} - 2\lambda\Theta\right), \quad (8)$$

where α is the learning rate, $z = D_{u,l^c,t,l_j} - D_{u,l^c,t,l_i}$ and $\lambda\Theta = \{\lambda_i\Theta_i | 1 \le i \le |\Theta|\}$, i.e., scaling parameters Θ with regularization parameters in λ. Note that when learning, bootstrap sampling is exploited to sample the unobserved POI l_j. The detailed leaning algorithm is described in Algorithm 1.

Algorithm 1 Learning Procedure of MEAP-T

Input: check-in data C, time threshold τ, learning rate α, regularization vector λ, number d of dimensions

Output: model parameters Θ

Draw Θ from Normal Distribution $N(0, 0.01)$ except for L and R

Initialize L and R with unit diagonal matrix

repeat

  shuffle the set of observations $\{< u, l^c, t, l_i >\}$

  for *each observation* $< u, l^c, t, l_i >$ **do**

    Randomly draw an unobserved POI l_j from $L \backslash L_u$

    Update $X^P(u)$, $X^P(l_i)$, $X^P(l_j)$

    Update $X^T(t)$, $X^T(l_i)$, $X^T(l_j)$

    Update β_{l_i}, β_{l_j}

    if $\Delta(l, l^c) \le \tau$ **then**

      Update $X^S(l^c)$, $X^S(l_i)$, $X^S(l_j)$, L, R

until *convergence*

return $\Theta = \{X^P(U), X^P(L), X^S(L), L, R, X^T(L), X^T(T), \beta(L)\}$

5 Experimental study

In this section, we conduct an extensive experimental study to answer the following questions: i) How does our approach perform in comparison to baselines and other state-of-the-art models? ii) How does the time interval between consecutive POI check-ins influence successive POI recommendation? And iii) How do the parameters affect the model performance?

5.1 Experimental setup and comparison methods

To fully demonstrate the performance of our model, we perform experiments on two real-world datasets which have been introduced in Section 3. For both datasets, we split the sequential check-ins of each user into three parts: 80% of behavioral records are selected for training, 10% for validating, and 10% for testing according to the check-in time order. The model aims to recommend a list of next POIs for each user, given the user's current location. We choose the next check-ins within successive τ seconds to evaluate model performance from test data. Recall that τ is vanishing threshold for temporal influence in (5). Then, as in prior work [2, 4, 14], we employ two widely used metrics, namely Precision@N and Recall@N, to measure model performance for successive POI recommendation, where N is the number of top-ranked recommendations and we will present the results of N = 5, 10, 15, and 20 for each metric.

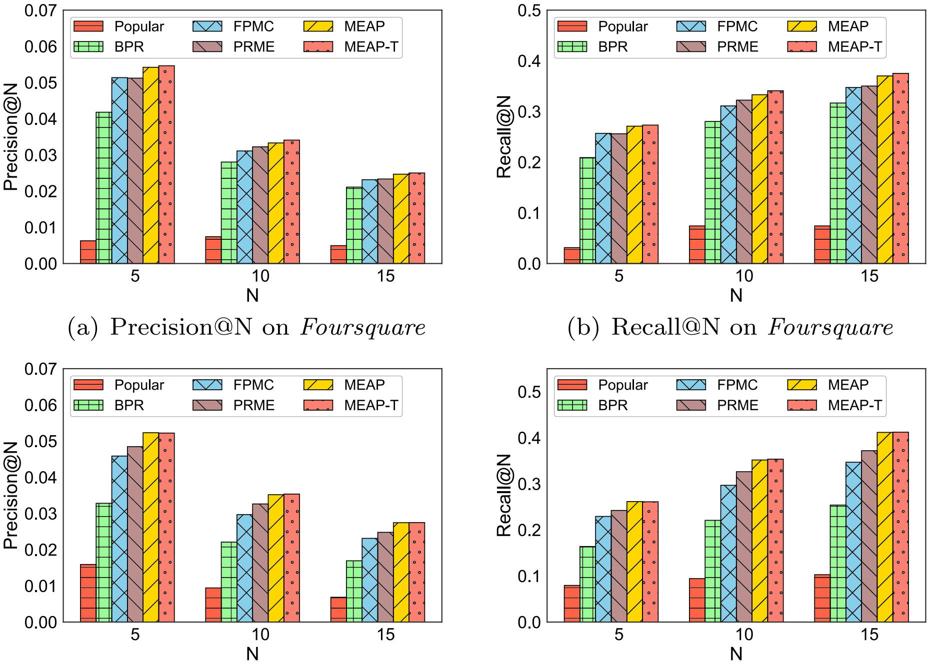
In the experiments, we compare our model with a series of state-of-the-art algorithms in successive POI recommendation as follows:

- **Popular.** The top ranked POIs based on popularity in the training set are selected as recommendation for each user.
- **BPR.** As check-in records can be treated as user implicit feedback, we introduce BPR, a state-of-the-art algorithm for recommendation tasks based on implicit feedback. This method only takes user preference into consideration and we choose Matrix Factorization as the underlying predictor [17].
- **FPMC.** This method considers user preference and sequence information simultaneously through Canonical Decomposition. Specifically, it combines personalized matrix factorization and non-personalized first-order Markov chains to provide the next basket recommendation [18].
- **PRME.** This method embeds POI presentation and user presentation into two spaces: POI sequential transition space and user preference space for successive POI recommendation. Since we only focus on the sequence transition, we do not consider the location constraints in the comparison [4].
- **MEAP.** This is our simplified algorithm without considering the temporal influence, i.e., the distance is defined by (4).
- **MEAP-T.** This method further incorporates temporal influence into MEAP. Therefore, three different latent spaces, namely personalized user preference, POI-POI sequential transition, and POI-time relationship, are modeled through embedding learning.

Finally, we list some important hyperparameters for reproducibility. After tuning hyperparameters in the validation set, the regularization and number of dimensions of BPR and FPMC are set to 0.001 and 100 on both datasets, respectively. We fix the number of dimensions to 60, component weight to 0.2, and regularization term to 0.001 for PRME on both datasets. For MEAP, the number of dimensions is set to 100 and regularizations are set as $\lambda_P = \lambda_\beta = 10^{-6}$, $\lambda_A = \lambda_S = 10^{-7}$ on both datasets. The number of dimensions in MEAP-T is also 100 and regularizations are set as $\lambda_P = \lambda_\beta = 10^{-4}$, $\lambda_A = \lambda_S = 10^{-5}$ on both datasets, and regularization λ_T is set to be 10^{-3} on *Foursquare* and 0.005 on *Gowalla*, respectively. The learning rate is set to 0.01 for all methods.

5.2 Comparison of performance

In Figure 3, we report the overall performance of all recommendation approaches with $\tau = 21600$ (sec.) on both *Foursquare* check-ins in New York City and *Gowalla* check-ins in California, respectively. We can observe that Popular gets much lower precision and recall than all other counterparts, indicating that this naive approach is insufficient for successive POI recommendation. Moreover, FPMC and PRME consistently perform much better than BPR. For example, FPMC improves BPR by 22.28 and 39.73% with Recall@5 on *Foursquare* and *Gowalla*, respectively. This is because BPR only considers personalized user-POI preference in latent space, while FPMC and PRME combine user preference and POI transition together. Therefore, sequential information is a significant factor for successive POI recommendation tasks. On the other hand, PRME is better than FPMC in most cases on *Foursquare* and is much better on *Gowalla*. Specifically, PRME outperforms FPMC by 3.67 and 9.94% with Recall@10 on *Foursquare* and *Gowalla*, respectively, possibly because PRME embeds POIs and users as single points in latent spaces, while FPMC



(a) Precision@N on *Foursquare* (b) Recall@N on *Foursquare*

(c) Precision@N on *Gowalla* (d) Recall@N on *Gowalla*

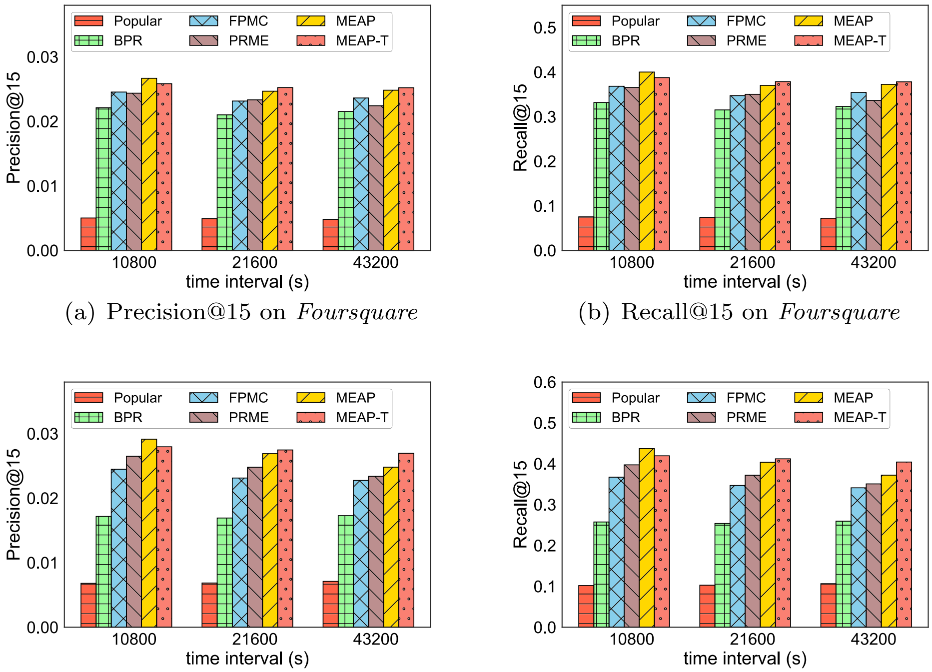
Figure 3 Overall performance comparison on *Foursquare* and *Gowalla*

represents them as independent vectors. This enables PRME to capture latent relationships of POI-POI and user-POI more naturally and precisely [1, 4].

Our proposed model MEAP performs much better than FPMC and PRME. Specifically, MEAP improves (FPMC, PRME) by (6.57%, 5.71%) on *Foursquare* and by (18.73%, 10.73%) on *Gowalla* with Precision@15. As we have discussed earlier in Section 3, successive POI check-in data possesses the characteristic of asymmetry property. Note that MEAP successfully models this property through metric embedding with asymmetric projection, while PRME assumes the symmetric property. This result indicates that asymmetry property should be taken into consideration to improve performance during embedding learning. Finally, MEAP-T achieves the best performance on both datasets, although it is only slightly better than MEAP on *Gowalla*. This shows that temporal influence is beneficial for successive POI recommendation to some extent.

5.3 The impacts of time interval

To explore the impacts of time interval, i.e., τ, we demonstrate the performance of all methods at different time intervals (i.e., 10,800 s., 21,600 s., 43,200 s.) in Figure 4. Due to the space constraint, we only show the results of Precision@15 and Recall@15, and the performances with different Ns are quite similar. Note that there is only one next POI as ground-truth given a user and his current POI in test data. Hence, Precision and Recall display the same trend.



(a) Precision@15 on *Foursquare* (b) Recall@15 on *Foursquare*

(c) Precision@15 on *Gowalla* (d) Recall@15 on *Gowalla*

Figure 4 The impacts of time interval τ on *Foursquare* and *Gowalla*

We can observe that Popular and BPR have similar performances at different time intervals. This is because they do not consider sequential information at all. Likewise, FPMC also presents similar results at various time intervals on both datasets. The reason may be that FPMC utilizes all training data to train the model regardless of time intervals between current and next POIs. The performances of PRME and MEAP decrease with the increase of time interval, especially on *Gowalla*. This observation reveals that POI sequential transition becomes weaker with larger time interval. Surprisingly, MEAP performs better than MEAP-T with $\tau = 10800$. One possible reason is that our datasets are sparse and there is not enough data to train MEAP-T when $\tau = 10800$. With the increase of τ, MEAP-T outperforms MEAP, which means that temporal factor improves the performance when τ is selected reasonably.

5.4 The impacts of the number of latent dimensions

We further investigate the impacts of the number of latent dimensions d, which is an important parameter when learning a latent ranking-based model. Figure 5 demonstrates the result on Recall@15. When the number of dimensions is less than 40, the performance increases fast, while it rises steadily from $d > 40$. This is because higher dimensionality can better embed POI-POI, POI-user, and POI-time relationships. In the experiments, we set the number of dimensions to 100 for the trade-off of recommendation quality and computation cost.

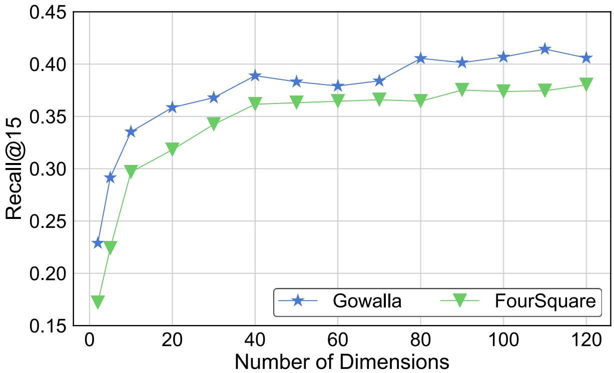


Figure 5 The impact of the number of latent dimensions on *Foursquare* and *Gowalla* under the metric of Recall@15

6 Conclusions and future work

In this paper, we studied the task of successive POI recommendation. We first demonstrated the existence of asymmetric property in successive check-in data. Then a novel time-aware metric embedding algorithm was developed, which incorporated asymmetric POI-POI transition and temporal influence. In particular, we embedded user preference, asymmetric POI-POI transition, and POI-time relationship into three distinct Euclidean spaces. We conducted comprehensive experiments on two real-world datasets to evaluate the performance of our proposed model. The results have demonstrated the superiority of our model compared with baseline methods.

Several issues need further investigations. First, given the observation that MEAP outperforms MEAP-T at low time interval τ, how to embed time influence more effectively for sparse data lies in our future study. Second, there are other types of contextual information, e.g., geography and category. It would be interesting to fuse these types of information to further improve the performance of successive POI recommendation.

Acknowledgments This research was partially supported by the Ministry of Education of China under grant of No.2017PT18, the Natural Science Foundation of China under grant of No. 61379119 and No. 61672453, the WE-DOCTOR company under grant of No. 124000-11110, the Zhejiang University Education Foundation under grant of No. K17-511120-017, and the Australia Research Council under grant of No. LP140100937.

References

1. Chen, S., Moore, J.L., Turnbull, D., Joachims, T.: Playlist prediction via metric embedding. In: Proceedings of the 18th International Conference on Knowledge Discovery and Data Mining, pp. 714–722. ACM (2012)
2. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In: IJCAI, vol. 13, pp. 2605–2611 (2013)
3. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090. ACM (2011)
4. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI, pp. 2069–2075 (2015)

5. He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: Proceedings of the 16th International Conference on Data Mining, pp. 191–200. IEEE (2016)
6. He, J., Li, X., Liao, L., Song, D., Cheung, W.K.: Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 137–143. AAAI Press (2016)
7. Kurashima, T., Iwata, T., Irie, G., Fujimura, K.: Travel route recommendation using geotags in photo sharing sites. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 579–588. ACM (2010)
8. Kurashima, T., Iwata, T., Hoshide, T., Takaya, N., Fujimura, K.: Geo topic model: joint modeling of user's activity area and interests for location recommendation. In: Proceedings of the Sixth International Conference on Web Search and Data Mining, pp. 375–384. ACM (2013)
9. Li, X., Cong, G., Li, X.L., Pham, T.A.N., Krishnaswamy, S.: Rank-geofm: a ranking based geographical factorization method for point of interest recommendation. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 433–442. ACM (2015)
10. Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., Rui, Y.: Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining, pp. 831–840. ACM (2014)
11. Lisi, C., Shang, S., Zhiwei, Z., Xin, C., Christian, S.J., Panos, K.: Location-aware top-k term publish/subscribe. In: Proceedings of the 34th International Conference on Data Engineering, pp. 230–241. IEEE (2018)
12. Liu, B., Fu, Y., Yao, Z., Xiong, H.: Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining, pp. 1043–1051. ACM (2013)
13. Liu, B., Xiong, H., Papadimitriou, S., Fu, Y., Yao, Z.: A general geographical probabilistic factor model for point of interest recommendation. IEEE Trans. Knowl. Data Eng. **27**(5), 1167–1179 (2015)
14. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: AAAI, pp. 194–200 (2016)
15. Liu, Y., Liu, C., Liu, B., Qu, M., Xiong, H.: Unified point-of-interest recommendation with temporal interval assessment. In: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining. ACM (2016)
16. Noulas, A., Scellato, S., Lathia, N., Mascolo, C.: Mining user mobility features for next place prediction in location-based services. In: Proceedings of the 12th International Conference on Data Mining, pp. 1038–1043. IEEE (2012)
17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp. 452–461. AUAI Press (2009)
18. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web, pp. 811–820. ACM (2010)
19. Shang, S., Ding, R., Yuan, B., Xie, K., Zheng, K., Kalnis, P.: User oriented trajectory search for trip recommendation. In: Proceedings of the 15th International Conference on Extending Database Technology, pp. 156–167. ACM (2012)
20. She, J., Tong, Y., Chen, L., Cao, C.C.: Conflict-aware event-participant arrangement and its variant for online setting. IEEE Trans. Knowl. Data Eng. **28**(9), 2281–2295 (2016)
21. Tong, Y., She, J., Meng, R.: Bottleneck-aware arrangement over event-based social networks: the max-min approach. World Wide Web **19**(6), 1151–1177 (2016)
22. Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., Ye, J., Lv, W.: The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662. ACM (2017)
23. Wang, W., Yin, H., Sadiq, S., Chen, L., Xie, M., Zhou, X.: Spore: a sequential personalized spatial item recommender system. In: Proceedings of the 32nd International Conference on Data Engineering, pp. 954–965. IEEE (2016)
24. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24. ACM (2016)
25. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. IEEE Trans. Syst. Man Cybern. Syst. Hum. **45**(1), 129–142 (2015)

26. Yin, H., Cui, B., Sun, Y., Hu, Z., Chen, L.: Lcars: a spatial item recommender system. ACM Trans. Inf. Syst. (TOIS) **32**(3), 11 (2014)
27. Yin, H., Cui, B., Zhou, X., Wang, W., Huang, Z., Sadiq, S.: Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. ACM Trans. Inf. Syst. (TOIS) **35**(2), 11 (2016)
28. Yin, H., Zhou, X., Cui, B., Wang, H., Zheng, K., Nguyen, Q.V.H.: Adapting to user interest drift for poi recommendation. IEEE Trans. Knowl. Data Eng. **28**(10), 2566–2581 (2016)
29. Yin, H., Wang, W., Wang, H., Chen, L., Zhou, X.: Spatial-aware hierarchical collaborative deep learning for poi recommendation. IEEE Trans. Knowl. Data Eng. **29**(11), 2537–2551 (2017)
30. Zhang, W., Wang, J.: Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1221–1230. ACM (2015)
31. Zhang, J.D., Chow, C.Y., Li, Y.: Lore: exploiting sequential influence for location recommendations. In: Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 103–112. ACM (2014)
32. Zhao, S., Zhao, T., Yang, H., Lyu, M.R., King, I.: Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 315–321. AAAI Press (2016)
33. Zheng, K., Shang, S., Yuan, N.J., Yang, Y.: Towards efficient search for activity trajectories. In: Proceedings of the 29th International Conference on Data Engineerin, pp. 230–241. IEEE (2013)
34. Zheng, K., Su, H., Zheng, B., Shang, S., Xu, J., Liu, J., Zhou, X.: Interactive top-k spatial keyword queries. In: Proceedings of the 31st International Conference on Data Engineering, pp. 423–434. IEEE (2015)
35. Zhou, N., Zhao, W.X., Zhang, X., Wen, J.R., Wang, S.: A general multi-context embedding model for mining human trajectory data. IEEE Trans. Knowl. Data Eng. **28**(8), 1945–1958 (2016)