Neural Process Lett (2018) 47:975–992
https://doi.org/10.1007/s11063-017-9681-8

Aggregated Temporal Tensor Factorization Model for Point-of-Interest Recommendation

Shenglin Zhao[1] · Irwin King[1] · Michael R. Lyu[1]

Published online: 12 August 2017

Abstract Point-of-interest (POI) recommendation is an important application in location-based social networks (LBSNs), which mines user check-in sequences to suggest interesting locations for users. Because user check-in behavior exhibits strong temporal patterns—for instance, users would like to check-in at restaurants at noon and visit bars at night. Hence, capturing the temporal influence is necessary to ensure the high performance in a POI recommendation system. Previous studies observe that the temporal characteristics of user mobility in LBSNs can be summarized in three aspects: periodicity, consecutiveness, and non-uniformness. However, previous work does not model the three characteristics together. More importantly, we observe that the temporal characteristics exist at different time scales, which cannot be modeled in prior work. In this paper, we propose an Aggregated Temporal Tensor Factorization (ATTF) model for POI recommendation to capture the three temporal features together, as well as at different time scales. Specifically, we employ a temporal tensor factorization method to model the check-in activity, subsuming the three temporal features together. Next, we exploit a linear combination operator to aggregate temporal latent features' contributions at different time scales. Experiments on two real-world data sets show that the ATTF model achieves better performance than the state-of-the-art temporal models for POI recommendation.

Keywords Point-of-interest recommendation · Temporal influence · Location-based social network · Tensor factorization

✉ Shenglin Zhao
slzhao@cse.cuhk.edu.hk

Irwin King
king@cse.cuhk.edu.hk

Michael R. Lyu
lyu@cse.cuhk.edu.hk

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China

1 Introduction

Point-of-interest (POI) recommendation is an important application in location-based social networks (LBSNs), which mines user historical check-in records and recommends users interesting locations where they would like to check-in in the future. LBSNs have become very popular recently with the explosively increase of smart phones. Traditional online social networks such as Facebook introduce location-based services, e.g., Facebook Places, to evolve to LBSN. In the meantime, LBSNs such as Yelp and Foursquare, serving for special target like restaurant recommendation, emerge and attract a great number of users. In such a context of prosperous LBSNs, POI recommendation is proposed, which recommends a personalized POI list for each user according to their tastes. To this end, POI recommendation creates values in two aspects: (1) helping users to discover interested locations and services in the city, and (2) helping the businesses in LBSNs launch advertisements for target customers. The potential value of POI recommendation attracts much academic attention, which motivates a variety of POI recommendation systems [4,32,38].

To establish a practical POI recommendation system, temporal influence plays an important role. Previous studies show that the user mobility in LBSNs exhibits distinct temporal features [3,4,32]. For example, users always stay in the office in the Monday afternoon, and enjoy entertainments in bars at night. In summary, the temporal features in users' check-in data can be abstracted in three aspects.

- *Periodicity* Users share the same periodic pattern, visiting the same or similar POIs at the same time slot [3,32]. For instance, a user always visits restaurants at noon, so do other users. Hence, the periodicity inspires the time-aware collaborative filtering method to recommend POIs [32].
- *Consecutiveness* A user's current check-in is largely correlated with the recent check-in [2,4]. Gao et al. [4] model this property by assuming that user preferences are similar in two consecutive hours. Cheng et al. [2] assume that two checked-in POIs in a short term are highly correlated in latent feature space.
- *Non-uniformness* A user's check-in preference changes at different hours of a day [4]. For example, at noon a user may visit restaurants while at night the user may have fun in bars.

By capturing the observed temporal features, a variety of systems are proposed to enhance POI recommendation performance [2,4,32], which gain better performance than general collaborative filtering (CF) methods [30]. Nevertheless, previous work [2,4,32] cannot model the three features together. Moreover, an important fact is ignored in prior work that the temporal influence exists at different time scales. For example, in day level, you may check-in POIs around your home in the earning morning, visit places around your office in the day time, and have fun at nightclubs in the evening. In week level, you may stay in the city for work in weekdays and go out for vocation in weekends. Hence, to better model the temporal influence, capturing the temporal features at different time scales is necessary.

In this paper, we propose an Aggregated Temporal Tensor Factorization (ATTF) model for POI recommendation to capture the three temporal features together, as well as at different time scales. We construct a user-time-POI tensor to represent the check-ins as shown in Fig. 1, and then employ the interaction tensor factorization [26] to model the temporal effect. Different from prior work that represents the temporal influence at single scale, we index the temporal information for latent representation at different scales, i.e., hour, week, day, and month. Furthermore, we employ a linear combination operator to aggregate different temporal latent features' contributions, which capture the temporal influence at different

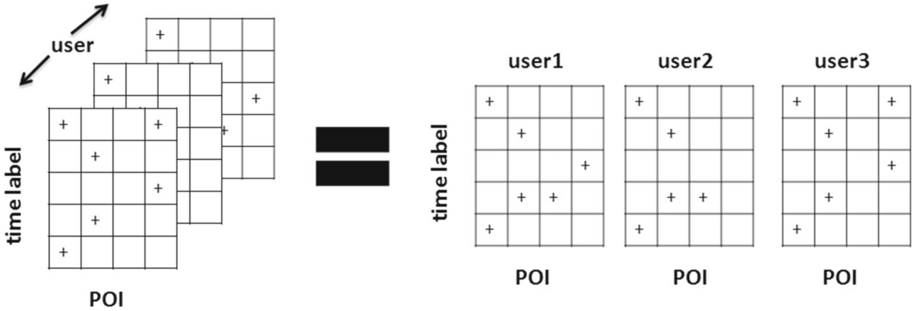


Fig. 1 Tensor illustration for check-ins

scales. Specifically, our ATTF model learns the three temporal properties as follows: (1) periodicity is learned from the temporal CF mechanism; (2) consecutiveness is manifested in two aspects—time in a slot brings the same effect through sharing the same time factor, and the relation between two consecutive time slots can be learned from the tensor model; (3) non-uniformness is depicted by different time factors representing different time slots from each time scale perspective. Moreover, an aggregate operator is introduced to combine the temporal influence at different scales, i.e., hour, week, day, and month, and represent the temporal effect in a whole.

To sum up, we propose the ATTF model to seek a better way to capture the temporal influence for POI recommendation. Note that we have presented our ATTF model for POI recommendation in [36]. In this paper, we extend the work [36] with a comprehensive related work review, an in-depth data empirical study, and more detailed performance analysis. Moreover, we establish an embedding neural network to represent the ATTF model, which gives new insights to understand the proposed model from the neural network perspective. Specifically, this paper makes the following additional contributions. First, we provide an empirical data analysis on the check-in data, which motivates our proposed ATTF model. Second, we demonstrate the three-slice time indexing scheme and show our memory reducing trick via a binary representation transformation. Third, we present the detailed inference procedure of our ATTF model and provide extensive discussion about our proposed model on performance and parameter effect. Last but not the least, we understand the ATTF model from the embedding neural network perspective, providing new insights to interpret the latent ranking models.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first temporal tensor factorization method for POI recommendation, subsuming all the three temporal properties: periodicity, consecutiveness, and non-uniformness.
- We propose a novel model to capture temporal effect in POI recommendation at different time scales. Experimental results show that our model outperforms prior temporal model more than 20%.
- The ATTF model is a general framework to capture the temporal features at different scales, which outperforms single temporal factor model and gains 10% improvement in the top-5 POI recommendation task on Gowalla data.
- We understand the ATTF model from the embedding neural network perspective, verifying the effectiveness of the embedding neural network that is a general framework

for latent factor models, including rating estimation models (e.g., MF [9]) and ranking models (e.g., our ATTF model).

This following of this paper is organized as follows. Section 2 reviews the most relevant work. Section 3 introduces our empirical analysis on check-in data and demonstrates the time labeling scheme. Next, Sect. 4 presents the details of our model. Then, Sect. 5 reports experimental results conducted on two real-world data sets. Finally, Sect. 6 draws the conclusion and gives possible further work in this study.

2 Related Work

In this section, we first review the literature of POI recommendation. Then, we summarize the progress of modeling temporal effect for POI recommendation. Finally, we review the literature of embedding learning and its applications, which inspires us to understand the proposed ATTF model from the neural network perspective.

POI Recommendation Most of POI recommendation systems base on the collaborative filtering (CF) techniques, which can be reported in two aspects, memory-based and model-based. On the one hand, Ye et al. [30] propose the POI recommendation problem in LBSNs solved by user-based CF method, and further improve the system by linearly combining the geographical influence, social influence, and preference similarity. In order to enhance the performance, more advanced techniques are then applied, e.g., incorporating temporal influence [32], and utilizing a personalized geographical model via kernel density estimator [33,34]. On the other hand, model-based CF is proposed to tackle the POI recommendation problem that benefits from its scalability. Cheng et al. [1] propose a multi-center Gaussian model to capture user geographical influence and combine it with social matrix factorization (MF) model [19] to recommend POIs. Gao et al. [4] propose an MF-based model, Location Recommendation framework with Temporal effects (LRT), utilizing similarity between time-adjacent check-ins to improve performance. Lian et al. [14] and Liu et al. [18] enhance the POI recommendation by incorporating geographical information in a weighted regularized matrix factorization model [8]. In addition, some researchers subsume users' comments to improve the recommendation performance [5,13,31]. Other researchers model the consecutive check-ins' correlations to enhance the system [2,16,37,38].

Temporal Effect Modeling In 2011, Cho et al. [3] propose the periodicity of check-in data in LBSNs. People always visit restaurants at noon, so we suffice to recommend users restaurants he/she did not visit at noon. Pearson correlation helps us to recommend similar POIs at the same time slot. However, experiments in [3] depend on dense check-in data not fitting most of users. In 2013, Yuan et al. [32] combine the temporal similarity and non-temporal similarity to measure the similarity better. At the same year, Gao et al. [4] observe the non-uniformness property (a user's check-in preference changes at different hours of a day), and consecutiveness (a user's preference at time t is similar with time $t - 1$). Further, Gao et al. propose LRT model based on MF technique to model the non-uniformness and consecutiveness. Meantime, Cheng et al. [2] propose the Factorized Personalized Markov Chain model [25] with Local Region constraint (FPMC-LR) to capture the consecutiveness, supposing strong correlation between two consecutive checked-in POIs. However, previous work does not model the three features together nor modeling the temporal influence at different scales.

Embedding Neural Network The embedding neural network, e.g., word2vec framework [20], has turned out to be a successful semi-supervised learning method. It is generally used in natural language processing [15,17]. Since the efficacy of the framework in capturing the correlations of items, the embedding neural network is employed to the network embedding [7,23,28], and as well as in recommendation systems [27,29]. Moreover, recent studies [11,12] show that the neural word embeddings can be treated as a kind of matrix factorization method [9]. This equivalence between neural embeddings and the latent factor models inspires us to understand our ATTF model from the embedding neural network perspective. Our interpretation for ATTF model from neural network perspective verifies that the embedding neural network can be treated as a general framework for latent factor models, including rating estimation models [9] and ranking models (e.g., our proposed ATTF model).

3 Preliminaries

In this section, we first analyse the temporal features of user check-in data. Then, we introduce the time labeling scheme that is the prerequisite of our ATTF model. We analyse user check-in data in Foursquare and Gowalla, which demonstrates the similar check-in pattern. In the following, we show the empirical data analysis result based on a randomly selected user in Foursquare.

3.1 Empirical Data Analysis

We leverage empirical data analysis to explore the three temporal properties of check-in data. Our analysis verifies previous discoveries, for instance, the non-uniformness—user check-in preference changes at different time of a day [4]. Moreover, we observe some new properties not covered in prior work, e.g., the non-uniformness exists at different time scales.

Data sparsity is a big concern in previous temporal models. Figure 2a demonstrates a user's check-in pattern in a day. We observe that the user always has many check-ins in the morning and evening, which verifies the periodicity. The check-in activity repeats in the morning and evening. Figure 2b shows the consecutive hour pair similarity,[1] i.e., the check-in similarity between time t and $t-1$ (t means the hour $1, 2, \ldots, 24$). We observe that the user check-in preference has high similarity at some consecutive hours, e.g., between 5 o'clock and 4 o'clock, 8 o'clock and 7 o'clock. However, we also find that at some time (e.g., 9:00), the user has few check-ins and the similarity is zero. Therefore, the sparse data make it too hard to model the periodicity and consecutiveness via a counting method (e.g., Pearson correlation or cosine similarity). As shown in Fig. 2b, most of the similarities are zero. So the consecutiveness cannot be modeled at most of the time. The dilemma of counting methods in face of sparse data motivates us to exploit a latent factor learning model. In our model, we use a time latent factor to represent the temporal effect of a time slot, not modeling the temporal effect from the user or POI perspective. Further, the temporal factor is learned from all users' check-ins at the time slot. Therefore, it overcomes the sparsity problem in counting methods.

We observe that the non-uniformness (e.g., the check-in change characteristics) exists at different time scales in users' check-in data. Following [4], we demonstrate an example of a random user's aggregated check-in activities on his/her top 5 most visited POIs in Fig. 3. Figure 3a verifies the non-uniformness: a user's check-in preference changes at different hours of a day [4]. As shown in Fig. 3a, the most visited POI changes at different hours. For

[1] We use cosine similarity here; other measures like Pearson correlation are also applicable.

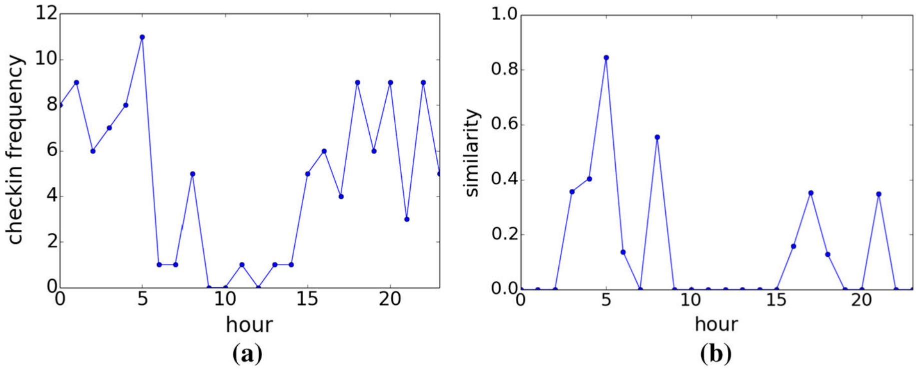


Fig. 2 Sparsity demonstration. **a** A user's check-in pattern in a day, **b** consecutive hour pair similarity

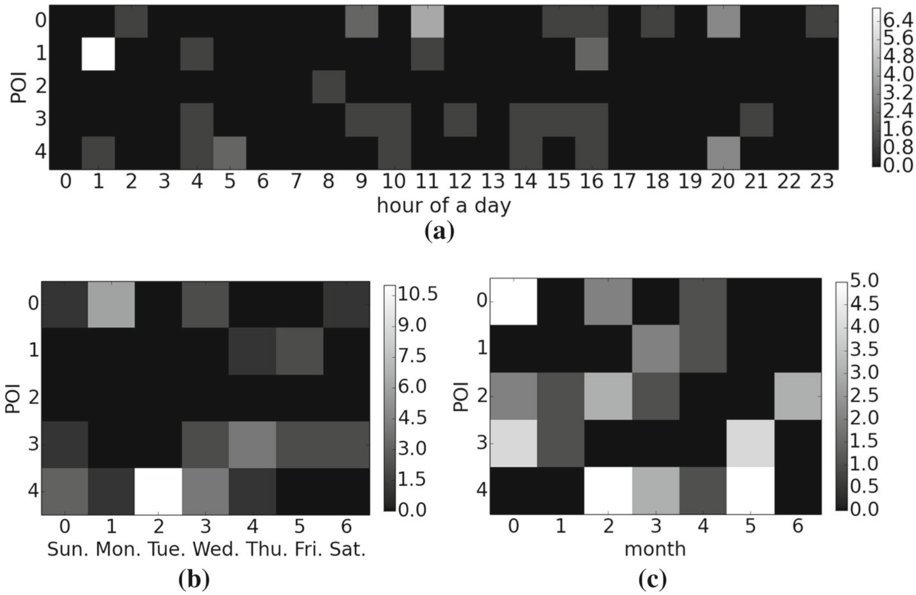
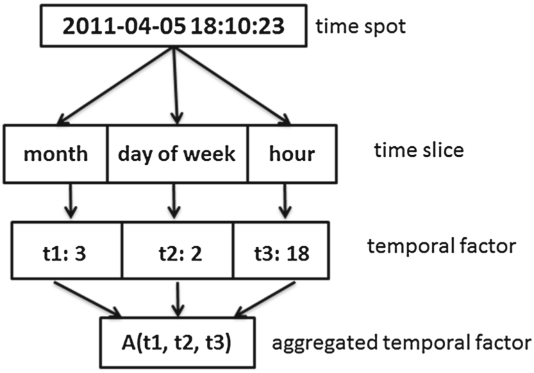


Fig. 3 Demonstration of non-uniformness at different time scales. **a** Non-uniformness in hour of a day, **b** non-uniformness in day of week, **c** non-uniformness in month

example, the most visited POI is POI 1 at 1:00 while the most visited POI is POI 4 at 5:00. Besides, we discover there are other change characteristics. As shown in Fig. 3b, c, a user's check-in preference changes at different months of a year, and among different days of a week as well. The change of check-ins at different time scales depicts the user preference from different perspectives: (1) A user may check-in at POIs around his/her home in the morning, visit places around the office in the day time, and have fun in bars at night. (2) A user may visit more locations around his/her home or office on weekdays. On weekends, he/she may check-in more at some shopping malls or vocation places. (3) At different months, a user may have different customs. For instance, he/she would visit ice cream shops in the months of summer and hot pot restaurants in the months of winter. Hence, only modeling

Fig. 4 Time labeling scheme demonstration



the non-uniformness at a single scale, we cannot capture all temporal features, which need to be formulated at different scales.

3.2 Time Labeling Scheme

Time labeling is a prerequisite of our ATTF model. We use a time latent factor to represent the temporal effect at a specific time, and then learn from a latent factor model. Time labeling scheme determines how to assign a latent factor to specific time. Before diving to the model, we describe the time labeling scheme first.

Figure 4 demonstrates the time labeling scheme. In order to capture temporal features at different time scales, we represent a time spot with several parts and then aggregate their contributions together. According to the empirical data analysis, we consider temporal features in three time scales: month of a year, day of a week, and hour of a day. Now the temporal effect is formulated by three latent time factors. As shown in Fig. 4, we leverage three slices to denote a time spot: month of year, day of week, and hour of day. Further they are depicted by three kinds of different temporal latent vectors respectively. So a time spot t is labeled by a tuple (t_1, t_2, t_3), which satisfies that $t_1 \in \{0, 1, \cdots, 12\}$, $t_2 \in \{0, 1, \cdots, 6\}$, $t_3 \in \{0, 1, \cdots, 23\}$ (we have 12 months in a year, 7 days in a week, and 24 h in a day). Furthermore, we define $T_1 \subset R^{12\times d}$, $T_2 \subset R^{7\times d}$, and $T_3 \subset R^{24\times d}$ to denote the corresponding temporal latent factor matrices, where d is the latent vector dimension.

To aggregate several temporal factors, we define an operator $A(\cdot) : R^d \times R^d \times R^d \to R^d$ to combine different temporal features. Take “2011-04-05 18:10:23” as an example, its label ids at month, day of week, and hour are 3 (April), 2 (Tuesday), 18 (after 18:00). Hence its temporal latent factor is formulated as $A(T_{1,3}, T_{2,2}, T_{3,18})$. It is important to note that our scheme is flexible: we are able to ignore one feature by taking away a slice, or introduce a new feature by adding a new slice.

Memory Reducing Trick We reduce the input data size through a binary coding trick. We employ one label id instead of three to represent the three slices. In detail, we use 4 bits to represent the month, 3 bits to represent the day of week, and 5 bits to represent the hour slot. So the time label id can be represented by an integer of 16 bits. For instance, “2011-04-05 18:10:23” could be coded as “0011 010 10010”, and its label id is 850. When we learn the model, we transform the label id into binary representation and find corresponding label to each slice. After labeling the time, we are able to model the temporal effect from a user-time-POI latent factorization model; see details in Sect. 4.

4 Methods

In this section, we first demonstrate the ATTF model. Then we give the detailed model inference and learning procedure. Finally, we summarize the model discussion.

4.1 Aggregated Temporal Tensor Factorization Model

Denote that $\mathcal{U}$ is the set of users and $\mathcal{L}$ is the set of POIs. In addition, $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ are the set of months, days of week, and hours respectively. Further we define $\mathcal{T}$ as the set of time label tuples, consisting of elements $t := (t_1, t_2, t_3)$, namely the temporal representation at different scales. The ATTF model estimates the preference of a user u at a POI l given a specific time label t through a score function $f(u, t, l)$, where $u \in \mathcal{U}$ is user id, $t \in \mathcal{T}$ is a time label tuple, and $l \in \mathcal{L}$ is POI id.

We are typically given N training examples $(u_i, t_i, l_i) \in \{1, \ldots, |\mathcal{U}|\} \times \{1, \ldots, |\mathcal{T}|\} \times \{1, \ldots, |\mathcal{L}|\}, i = 1, 2, \ldots, N$, and correspondingly outputs $y_i \in R, i = 1, 2, \ldots, N$. Here, (u_i, t_i, l_i) is the index of a particular element of a user-time-POI tensor, and y_i is the preference score of the user at the POI given the time label. One could simply collate the training data to build a suitable tensor, so the training task turns to fill in the blank entries of the tensor.

We exploit the Pairwise Interaction Tensor Factorization (PITF) [26] model to decompose the user-time-POI tensor. PITF model that turns out to be successful in the ECML/PKDD Discovery Challenge, runs much faster than other tensor factorization methods and has better performance in a large scale prediction task [26]. Thus the score of a POI l given user u and time t is factorized into three interactions: user-time, user-POI, and time-POI, where each interaction is modeled through the latent vector product. Further, we inference the model via Bayesian Personalized Ranking (BPR) criteria [24] that is a general framework to train a recommendation system from implicit feedback. Because prior work [14,37,38] indicates that treating the check-ins as implicit feedback is better than explicit ways for POI recommendation. Since we recommend POIs for users at specific time, any candidate POI has the same user-time interaction. As a result, the preference score is independent of the user-time interaction. Then the score function for a given time label t, user u, and a target POI l could be formulated as :

$$f(u, t, l) = \langle U_u^{(L)}, L_l^{(U)} \rangle + \langle A(T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)}), L_l^{(T)} \rangle, \tag{1}$$

where $\langle \cdot \rangle$ denotes the vector inner product, $A(\cdot)$ is the aggregate operator. Suppose that d is the latent vector dimension, $U_u^{(L)} \in R^d$ is user u's latent vector for POI interaction, $L_l^{(U)}, L_l^{(T)} \in R^d$ are POI l's latent vectors for user interaction and time interaction, $T_{1,t_1}^{(L)}$, $T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)} \in R^d$ are time t's latent vector representations in three aspects: month, day of week, and hour.

Aggregate operator combines the several temporal features together. In this paper, we propose a linear convex combination operator. It is formulated as follows,

$$A(\cdot) = \alpha_1 \cdot T_{1,t_1}^{(L)} + \alpha_2 \cdot T_{2,t_2}^{(L)} + \alpha_3 \cdot T_{3,t_3}^{(L)}, \tag{2}$$

where α_1, α_2, and α_3 denote the weights of each temporal factor, which satisfy $\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\alpha_1, \alpha_2, \alpha_3 >= 0$.

4.2 Learning

We infer the model via BPR criteria [24], which treats the check-in activity as a kind of implicit feedback. Namely, we assume the user prefers the visited POIs than the unvisited. We treat the visited POIs as positive and the unvisited as negative. Then, we suppose that the score of $f(u, t, l)$ at positive observations is higher than the negative POIs, given u and t. Further, we formulate the relation that user u prefers a positive POI l_i than a negative one l_j at time t as follows

$$l_i >_{u,t} l_j. \tag{3}$$

Based on the pairwise preference defined above, we suffice to extract the set of preference constraints from the training examples

$$D_S := \{(u, t, l_i, l_j)|l_i >_{u,t} l_j, u \in \mathcal{U}, t \in \mathcal{T}, l_i, l_j \in \mathcal{L}\}. \tag{4}$$

For simplicity, we denote $y_{u,t,l} = f(u, t, l)$. Then for any quadruple in D_S, it satisfies $y_{u,t,l_i} > y_{u,t,l_j}$. Using a logistic function to model this relation, we get

$$p(l_i >_{u,t} l_j) := \sigma(y_{u,t,l_i} - y_{u,t,l_j}), \tag{5}$$

which measures the probability of l_i is a positive observation and l_j is a negative observation for user u at time t. In Eq. (5), σ is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$.

Suppose the quadruples in D_S are independent of each other, then learning the ATTF model is to maximize the likelihood of all the pair orders

$$\arg\max_{\Theta} \prod_{(u,t,l_i,l_j)\in D_S} p(l_i >_{u,t} l_j), \tag{6}$$

where Θ is the parameters to learn, namely $U^{(L)}$, $L^{(U)}$, $L^{(T)}$, $T_1^{(L)}$, $T_2^{(L)}$, and $T_3^{(L)}$. The objective function is equivalent to minimizing the negative log likelihood. To avoid the risk of overfitting, we add a Frobenius norm term to regularize the parameters. Then the objective function is

$$\arg\min_{\Theta} \sum_{(u,t,l_i,l_j)\in D_S} - ln(\sigma(y_{u,t,l_i} - y_{u,t,l_j})) + \lambda_\Theta ||\Theta||_F^2, \tag{7}$$

where λ_Θ is the regularization parameter.

We leverage the Stochastic Gradient Decent (SGD) algorithm to learn the objective function for efficacy. First, we define $y_{u,t,l_p,l_n} = y_{u,t,l_p} - y_{u,t,l_n}$, which models the pairwise relation in D_S. Further we denote a common part in gradient decent values for all parameters as $\delta = 1 - \sigma(y_{u,t,l_p,l_n})$. As $T_{1,t_1}^{(L)}$, $T_{2,t_2}^{(L)}$, and $T_{3,t_3}^{(L)}$ are symmetric, they have the same gradient form. For simplicity, we use $T_t^{(L)} \in \{T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)}\}$ to represent any of them, $\alpha \in \{\alpha_1, \alpha_2, \alpha_3\}$ to denote corresponding weight, and $A(\cdot)$ to denote $A(T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)})$. Then the updating rule for the parameters is as follows,

```
Input: Training tuples $\{(u_i, t_i, l_i)\}_{i=1,\ldots,N}$
Output: $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$
Initialize $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$
Uniformly sample $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$ check-in tuples from $D_S$ to generate $D_e$ for loss calculation
for iterations do
    //$S$ is the number of sampled check-ins
    for $i \in [1, S]$ do
        Draw $(u, t, l_p)$ uniformly from training tuples
        // $k$ is the number of negative samples
        for $n = 1, 2, \ldots, k$ do
            Draw $(u, t, l_p, l_n)$ uniformly
            $y_{u,t,l_p,l_n} \leftarrow y_{u,t,l_p} - y_{u,t,l_n}$
            $\delta \leftarrow 1 - \sigma(y_{u,t,l_p,l_n})$
            Update parameters according to Eq. (8)
        end
    end
    Estimate the loss defined on $D_e$
end
Return $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$
```

Algorithm 1: ATTF model learning algorithm

$$\begin{aligned}
U_u^{(L)} &\leftarrow U_u^{(L)} + \gamma \cdot \left(\delta \cdot \left(L_{l_p}^{(U)} - L_{l_n}^{(U)}\right) - \lambda \cdot U_u^{(L)}\right), \\
L_{l_p}^{(U)} &\leftarrow L_{l_p}^{(U)} + \gamma \cdot \left(\delta \cdot U_u^{(L)} - \lambda \cdot L_p^{(U)}\right), \\
L_{l_p}^{(T)} &\leftarrow L_{l_p}^{(T)} + \gamma \cdot \left(\delta \cdot A(\cdot) - \lambda \cdot L_{l_p}^{(T)}\right), \\
L_{l_n}^{(U)} &\leftarrow L_{l_n}^{(U)} - \gamma \cdot \left(\delta \cdot U_u^{(L)} + \lambda \cdot L_n^{(U)}\right), \\
L_{l_n}^{(T)} &\leftarrow L_{l_n}^{(T)} - \gamma \cdot \left(\delta \cdot A(\cdot) + \lambda \cdot L_{l_n}^{(T)}\right), \\
T_t^{(L)} &\leftarrow T_t^{(L)} + \gamma \cdot \left(\delta \cdot \alpha \cdot \left(L_{l_p}^{(T)} - L_{l_n}^{(T)}\right) - \lambda \cdot T_t^{(L)}\right),
\end{aligned} \tag{8}$$

where γ is the learning rate, λ is the regularization parameter. To train the model, we use the bootstrap skill to draw the quadruple from D_S, following [24]. Algorithm 1 gives the detailed procedure to learn the ATTF model. We aim to learn the latent representations of user, temporal features, and POIs, namely $U^{(L)}, T_1^{(L)}, T_2^{(L)}, T_3^{(L)}, L^{(U)}, L^{(T)}$. Let $|\mathcal{U}|$ denote the number of users, then we generate about $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$ tuples from D_S to generate a set D_e for the loss estimation, namely the negative log likelihood value. We follow [24] to set the number of samples for loss estimation as $\lfloor 100 * \sqrt{|\mathcal{U}|} \rfloor$. In each iteration, we sample S check-ins and then generate negative samples to learn the model. After that, we calculate the loss value over D_e: $\sum_{(u,t,l_i,l_j) \in D_e} -ln(\sigma(y_{u,t,l_i} - y_{u,t,l_j})) + \lambda_\Theta ||\Theta||_F^2$. The convergent condition is satisfied when the loss value for the fixed sampled tuples does not decrease.

Complexity The runtime for predicting a triple (u, t, l) is in $O(d)$, where d is the number of latent vector dimension. The updating procedure is also in $O(d)$. Hence training an quadruple is in $O(d)$, then training an example (u, t, l) is in $O(k \cdot d)$, where k is the number of sampled negative POIs. For each iteration, we sample S training examples. The calculation cost for loss estimation is less than the parameter updating procedure. Therefore, training the model costs $O(I \cdot S \cdot k \cdot d)$, where I is the number of iterations. In practical, I is always small for different datasets, in the range of [5, 30]. We usually treat the number of iterations as a constant, so the complexity of the model training is in $O(S \cdot k \cdot d)$.

4.3 Model Discussion

The ATTF model can be treated as a linear combination of two matrix factorization models which learn user preference and temporal effect respectively, as shown in Eq. (1). The first term depicts the user-POI interaction, which is similar as the low rank matrix factorization for the user-POI matrix through collaborative filtering technique. The second term depicts the time-POI interaction, which acts like leveraging a latent factor model to describe the relations between time labels and POIs. Further the aggregate operator $A(\cdot)$ combines several temporal factors together.

Two points are important to note for our model: (1) The ATTF model and the time labeling scheme are a general framework to subsume several temporal characteristics together. We take three common ones in this work, but it is easy to add others, e.g., different days in a month, workdays and vocations in a year. (2) Even though the model equation for ATTF in POI recommendation suffices to be expressed by a combination of two MF models, it is different from a simple ensemble of two MF model recommendation results because in our case the model parameters are learned jointly. Thus the learned parameters jointly represent the user preference and temporal effect. It better reflects the fact that user check-in behavior is a complex decision under many conditions.

The ATTF model can also be interpreted from the embedding neural network perspective. The embedding network, e.g., word2vec framework [20], has turned out to be a successful semi-supervised learning method in natural language processing [10,22], network embedding [7,23,28], and recommendation systems [27,29]. Moreover, recent studies [11,12] show that the neural word embeddings can be treated as a kind of matrix factorization method [9]. This equivalence between neural embeddings and the latent factor models inspires us to understand our ATTF model from the embedding neural network perspective. Figure 5 demonstrates the equivalent embedding neural network for the ATTF model. The input layer is the one-hot representation for user, POI, and temporal information. The second layer is the embedding layer, which projects the one-hot vector as a continuous latent vector in the Euclidean subspace. Next, we exploit the product and sum operation to represent the check-in preference as $\langle U_u^{(L)}, L_l^{(U)}\rangle + \langle A(T_{1,t_1}^{(L)}, T_{2,t_2}^{(L)}, T_{3,t_3}^{(L)}), L_l^{(T)}\rangle$, equivalent to Eq. (1). Finally, we construct a BPR loss layer to learn the embedding representations.

5 Experiments

We conduct systematical experiments to seek the answers of the following questions: (1) how the proposed ATTF model performs comparing with state-of-the-art models? (2) whether the ATTF model is better than single temporal factor models? (3) how the parameters affect the model performance?

5.1 Data Description and Experiment Setting

Two real-world data sets are used in the experiment: one is Foursquare data from January 1, 2011 to July 31, 2011 provided in [6] and the other is Gowalla data from January 1, 2011 to September 31, 2011 in [35]. We filter the POIs checked-in by less than 5 users and then choose users who check-in more than 10 times as our samples. After the preprocessing, the data sets contain the statistical properties as shown in Table 1. We randomly choose 80% of each user's check-ins as training data, and the remaining 20% for test data. Moreover, we use each check-in (u, t, l) in training data to learn the latent features of user, time, and POI.

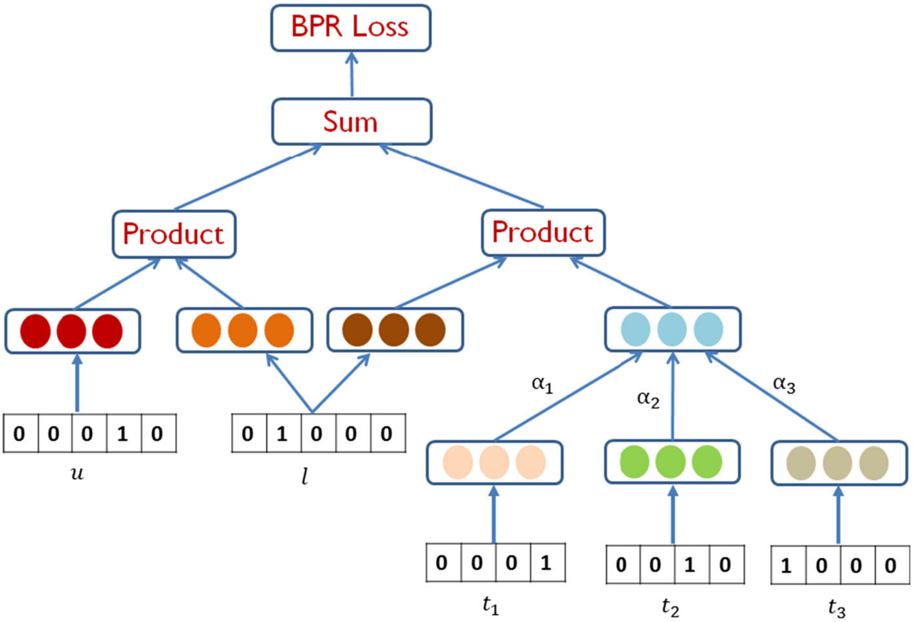


Fig. 5 Embedding neural Network for ATTF model

Table 1 Statistics of data sets

Source	Foursquare	Gowalla
#users	10,180	3,318
#POIs	16,561	33,665
#check-ins	867,107	635,600
Avg. #check-ins each user	85.2	191.6
Avg. #POIs each user	24.3	104.1
Avg. #users each POI	14.9	10.3
Density	0.0015	0.003

Density means the fraction of checked-in entries over all entries in user-POI matrix

Then given the (u, t), we estimate the score value of different candidate POIs, select the top N candidates, and compare them with check-in tuples in test data.

5.2 Performance Metrics

In this work, we leverage three metrics to evaluate the model performance–*precision*, *recall*, and *F-score*. The precision and recall in the top-K recommendation system are denoted as P@K and R@K respectively. P@K measures the ratio of recovered POIs to the K recommended POIs, and R@K means the ratio of recovered POIs to the set of POIs in the testing data. For each user $u \in \mathcal{U}$, $\mathcal{L}^T(u)$ denotes the set of correspondingly visited POIs in the test data, and $\mathcal{L}^R(u)$ denotes the set of recommended POIs. Then the definitions of P@K and R@K are formulated as follows

$$P@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}^R(u) \cap \mathcal{L}^T(u)|}{K}, \tag{9}$$

$$R@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}^R(u) \cap \mathcal{L}^T(u)|}{|\mathcal{L}^T(u)|}. \tag{10}$$

Further, *F-score* is the harmonic mean of precision and recall. So the *F-score* is defined as

$$F\text{-}score@K = \frac{2 * P@K * R@K}{P@K + R@K}. \tag{11}$$

5.3 Baselines

We compare our ATTF model with state-of-the-art collaborative filtering (CF) methods and POI recommendation methods incorporating temporal effect. Prior work [37,38] indicates that treating the check-ins as implicit feedback is better to recommend POIs. Hence we exploit Weighted Regularized Matrix Factorization (WRMF) [8,21] and Bayesian Personalized Ranking Matrix Factorization (BPR-MF) [24] as comparative CF methods. To illustrate the efficacy of our ATTF model, we compare it with LRT [4] and FPMC-LR [2] which are state-of-the-art POI recommendation methods incorporating temporal effect.

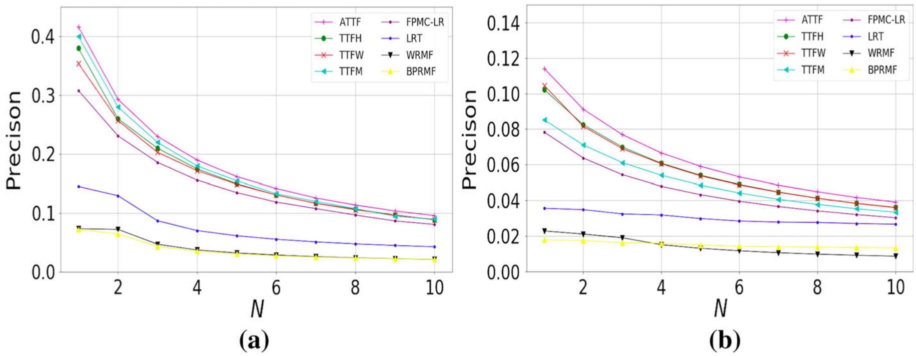
- *WRMF* The WRMF model is designed for processing large scale implicit feedback data sets. We define the weight mapping of user u_i at POI l_j as $w_{i,j} = (1 + 10 \cdot C_{i,j})^{0.5}$, where $C_{i,j}$ is the check-in counts, following the setting in [18].
- *BPR-MF* The BPR-MF model is a popular MF-based recommendation method to learn the pairwise relation, in which users prefer the observed items than the unobserved.
- *LRT* The LRT model is designed to modeling the “non-uniformnes” and “consecutiveness” in a matrix factorization model for POI recommendation.
- *FPMC-LR* The FPMC-LR model adds the Local Region constraint (i.e., geographical information) in the Factorized Personalized Markov Chain (FPMC) model [25]. FPMC-LR incorporates the geographical information and temporal consecutiveness through a local region constraint and the FPMC model respectively.

Moreover, to demonstrate the advantage of ATTF in aggregating several temporal latent factors, we also compare with three single temporal latent factor models: **TTFM**, **TTFW**, and **TTFH**. They are typically PITF model, that correspondingly considering the month, day of week, and hour as a temporal latent factor. Because these three models are the subset of our ATTF model, we attain their results by setting the corresponding weight as 1, and others as 0 in ATTF.

5.4 Experimental Results

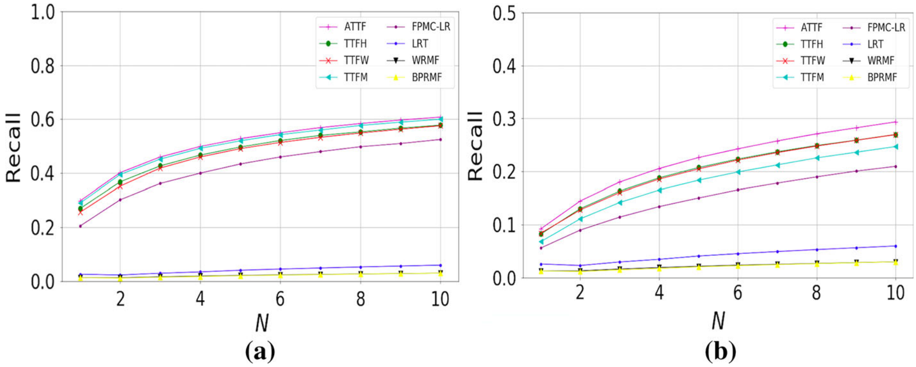
5.4.1 Performance Comparison

In the following, we demonstrate the performance comparison on precision, recall and F-score. We set the latent factor dimension as 60 for all compared models. We leverage grid search method to find the best weights in ATTF model. α_1, α_2, and α_3 are constrained in the range of [0, 1]. In the grid search method, we first change α_1 from zero to one with step size 0.1. Then, for each α_1 value, for instance $\alpha_1 = 0.1$, we change α_2 from zero to $1 - \alpha_1$ with step size 0.1. α_3 can be calculated by $1 - \alpha_1 - \alpha_2$. The grid search method tries all value combinations with step size 0.1 satisfying the constraints $\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\alpha_1, \alpha_2, \alpha_3 >= 0$. As a result, the ATTF model on Foursquare data achieves the best result when $\alpha_1 = 0.7$, $\alpha_2 = 0.1$, and $\alpha_3 = 0.2$, while the ATTF model on Gowalla data achieves the best when $\alpha_1 = 0.2$, $\alpha_2 = 0.1$, and $\alpha_3 = 0.7$.



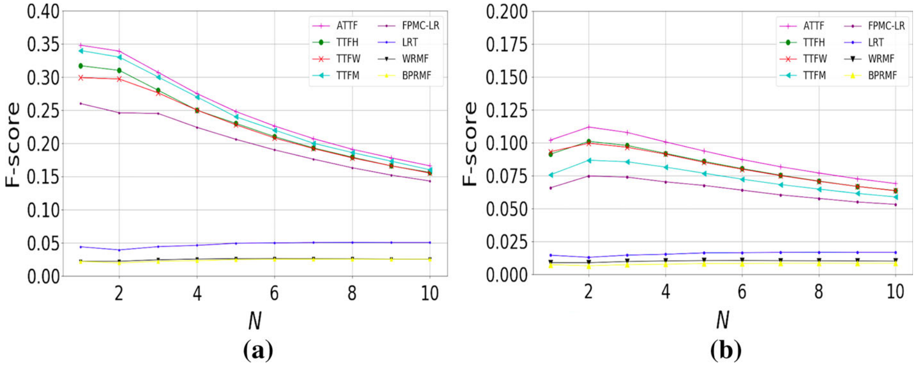
(a) (b)

Fig. 6 Precision on Foursquare and Gowalla. **a** Foursquare, **b** Gowalla



(a) (b)

Fig. 7 Recall on Foursquare and Gowalla. **a** Foursquare, **b** Gowalla



(a) (b)

Fig. 8 F-score Foursquare and Gowalla. **a** Foursquare, **b** Gowalla

Figures 6, 7, and 8 show the experimental results for Foursquare and Gowalla data on measurement precision, recall, and F-score respectively. We see that (1) Our proposed ATTF model outperforms state-of-the-art CF methods and POI recommendation models. Compared with the best state-of-the-art competitor in POI recommendation area (e.g., FPMC-LR), the ATTF model gains more than 20% enhancement on Foursquare data, and more than 36% enhancement on Gowalla data for all three measures, Precision@5, Recall@5, and F-

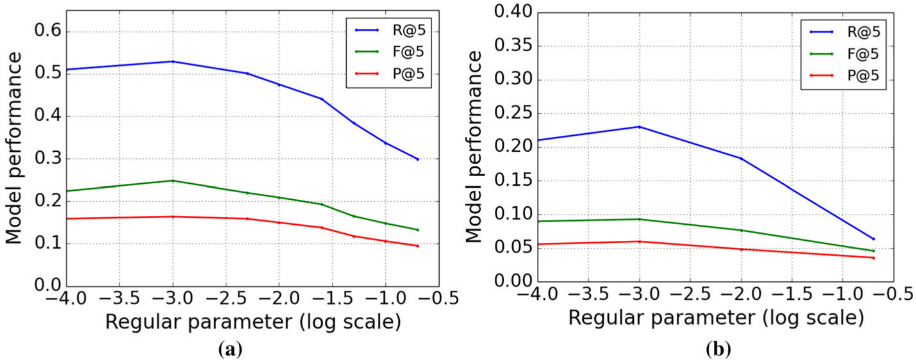


Fig. 9 The effect of regularization parameter λ. **a** Foursquare, **b** Gowalla

score@5. We observe that models perform better on Foursquare data set than Gowalla data set, even though it is sparser. The reason lies in that Gowalla data contain much more POIs and a large candidate POI set makes the recommendation harder. (2) The ATTF model outperforms single temporal factor models. Compared with best single temporal factor model, the ATTF model gains about 3% enhancement on Foursquare data, and about 10% improvement on Gowalla data in a top-5 POI recommender system. So when data are denser, the ATTF model gets advantages. Because the ATTF model uses a tuple to represent a time spot, which gives more precise information. Dense data strengthen this precise labeling scheme. In addition, different weight assignments on both data give us two interesting insights: (i) When data are sparse, the temporal feature on month dominates the POI recommendation performance. Because check-ins on hour or day of week are sparse as shown in Fig. 3, then the corresponding characteristics are not easily caught. The Foursquare data set has high weight on month temporal factor. However, when data are denser, check-ins on hour are not so sparse. So the temporal characteristic on hour of day becomes prominent. (ii) We usually pay much attention on temporal characteristics on hour of day and day of week. Our experimental results indicate that the temporal characteristic on month is important, especially for sparse data. (3) Our proposed ATTF model, single temporal factorization models (e.g., TTFM, TTFW, and TTFH), and FPMC-LR perform much better than other competitors, especially at recall measure. They try to recommend POIs at more specific situations, which is the key point to improve performance. Our models recommend a user POIs at some specific time, and FPMC-LR recommends POIs given a user's recent checked-in POIs; while, the other three models give general recommendations.

5.4.2 Parameter Effect

The regularization parameter and latent vector dimension are two important factors affecting the model performance. We explore how they affect the proposed model in the condition of other parameters fixed.

Figure 9 demonstrates the effect of regularization parameter on model performance. For simplicity, we set the same parameter λ for all latent vectors. The regularization part does not significantly affect the model. The model achieves best performance at 0.001. With the increasing of λ, the performance decreases.

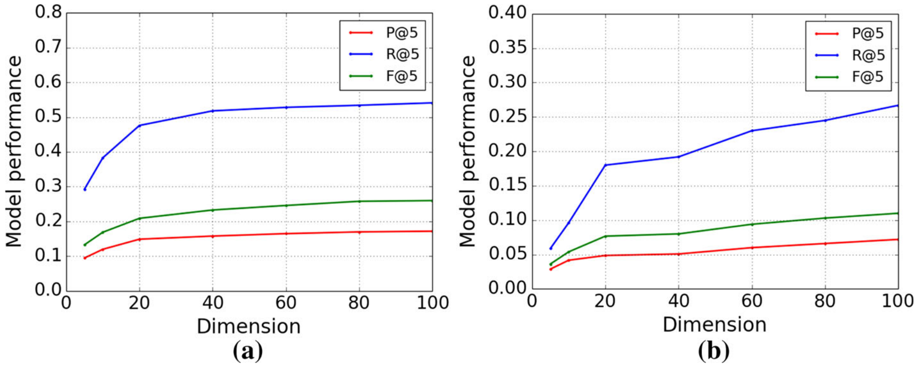


Fig. 10 The effect of latent factor dimension. **a** Foursquare, **b** Gowalla

Figure 10 demonstrates how the latent vector dimension affects the model. The performance of ATTF steadily rises with the increase of latent vector dimension. For the trade-off of high performance and low computation cost, we suggest to set dimension $d = 60$.

6 Conclusion and Future Work

In this paper, we propose the ATTF model for POI recommendation. The proposed model introduces time factor to model the temporal effect in POI recommendation, subsuming all the three temporal properties: periodicity, consecutiveness, and non-uniformness. Moreover, the ATTF model captures the temporal influence at different time scales through aggregating several time factors' contributions. Experimental results on two real-world data sets show that the ATTF model outperforms state-of-the-art models. Our model is a general framework to aggregate several temporal characteristics at different scales. In this work, we only consider three temporal characteristics: hour, week, day, and month. In the future, we may add more in this model, e.g., splitting 1 day into several slots rather than in hour. In addition, we may try nonlinear aggregate operators, e.g., max, in our future work.

Acknowledgements The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14203314 and No. CUHK 14234416 of the General Research Fund), and 2015 Microsoft Research Asia Collaborative Research Program (Project No. FY16-RES-THEME-005).

References

1. Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: Proceedings of the twenty-sixth AAAI conference on artificial intelligence. AAAI Press, pp 17–23
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 twenty-third international joint conference on artificial intelligence. AAAI Press, pp 2605–2611
3. Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1082–1090

4. Gao H, Tang J, Hu X, Liu H (2013) Exploring temporal effects for location recommendation on location-based social networks. In: Proceedings of the 7th ACM conference on recommender systems. ACM, pp 93–100
5. Gao H, Tang J, Hu X, Liu H (2015)Content-aware point of interest recommendation on location-based social networks. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence. AAAI Press, pp 1721–1727
6. Gao H, Tang J, Liu H (2012) gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In: Proceedings of the 21st ACM international conference on information and knowledge management. ACM, pp 1582–1586
7. Grover A, Leskovec J(2016) node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 855–864
8. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: 2008 eighth IEEE international conference on data mining. Ieee, pp 263–272
9. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
10. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31th international conference on machine learning, vol 14, pp 1188–1196
11. Levy O, Goldberg Y (2014) Neural word embedding as implicit matrix factorization. In: Proceedings of the 27th international conference on neural information processing systems. MIT Press, pp 2177–2185
12. Li Y, Xu L, Tian F, Jiang L, Zhong X, Chen E (2015) Word embedding revisited: a new representation learning and explicit matrix factorization perspective. In: Proceedings of the 25th international joint conference on artificial intelligence, pp 3650–3656
13. Lian D, Ge Y, Zhang F, Yuan NJ, Xie X, Zhou T, Rui Y (2015) Content-aware collaborative filtering for location recommendation based on human mobility data. In: 2015 IEEE international conference on data mining (ICDM). IEEE, pp 261–270
14. 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: ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 831–840
15. Liu P, Qiu X, Huang X (2015) Learning context-sensitive word embeddings with neural tensor skip-gram model. In: Proceedings of the 25th international joint conference on artificial intelligence
16. Liu X, Liu Y, Aberer K, Miao C (2013) Personalized point-of-interest recommendation by mining users' preference transition. In: Proceedings of the 22nd ACM international conference on information & knowledge management. ACM, pp 733–738
17. Liu Y, Liu Z, Chua TS, Sun M (2015) Topical word embeddings. In: Proceedings of the 29th AAAI conference on artificial intelligence
18. Liu Y, Wei W, Sun A, Miao C (2014) Exploiting geographical neighborhood characteristics for location recommendation. In: ACM international conference on conference on information and knowledge management, pp 739–748
19. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the fourth ACM international conference on web search and data mining. ACM, pp 287–296
20. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26:3111–3119
21. Pan R, Zhou Y, Cao B, Liu NN, Lukose R, Scholz M, Yang Q (2008) One-class collaborative filtering. In: Proceedings of the 2008 eighth IEEE international conference on data mining. IEEE Computer Society, pp 502–511
22. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: EMNLP, vol 14, pp 1532–1543
23. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710. ACM
24. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461
25. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on world wide web. ACM, pp 811–820
26. Rendle S, Schmidt-Thieme L (2010) Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the third ACM international conference on web search and data mining. ACM, pp 81–90

27. Tang D, Qin B, Liu T, Yang Y (2015) User modeling with neural network for review rating prediction. In: IJCAI, pp 1340–1346
28. Tang J, Qu M, Wang M, Zhang M, Yan, J, Mei, Q (2015) Line: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. ACM, pp 1067–1077
29. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S (2016) Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management. ACM, pp 15–24
30. Ye M, Yin P, Lee WC, Lee DL (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 325–334
31. Yin H, Sun Y, Cui B, Hu Z, Chen L (2013) Lcars: a location-content-aware recommender system. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 221–229
32. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 363–372
33. Zhang JD, Chow CY (2013) igslr: personalized geo-social location recommendation: a kernel density estimation approach. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 334–343
34. Zhang JD, Chow CY (2015) Geosoca: exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 443–452
35. Zhao S, King I, Lyu MR (2013) Capturing geographical influence in poi recommendations. In: International conference on neural information processing. Springer, pp 530–537
36. Zhao S, Lyu MR, King I (2016) Aggregated temporal tensor factorization for point-of-interest recommendation. In: International conference on neural information processing. Springer
37. Zhao S, Zhao T, King I, Lyu MR (2017) Geo-teaser: geo-temporal sequential embedding rank for point-of-interest recommendation. In: Proceedings of the 26th international conference on world wide web companion, pp 153–162
38. Zhao S, Zhao T, Yang H, Lyu MR, King I (2016) Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation. In: Thirtieth AAAI conference on artificial intelligence