Qian TY, Liu B, Hong L *et al.* Time and location aware points of interest recommendation in location-based social networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(6): 1219–1230 Nov. 2018. DOI 10.1007/s11390-018-1883-7

# Time and Location Aware Points of Interest Recommendation in Location-Based Social Networks

Tie-Yun Qian[1], *Member, CCF, ACM, IEEE*, Bei Liu[1]
Liang Hong[2,*], *Senior Member, CCF, Member, ACM, IEEE*, and Zhen-Ni You[1]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*School of Information Management, Wuhan University, Wuhan 430072, China*

E-mail: {qty, liubei, hong, znyou}@whu.edu.cn

Received August 27, 2017; revised September 21, 2018.

**Abstract** The wide spread of location-based social networks brings about a huge volume of user check-in data, which facilitates the recommendation of points of interest (POIs). Recent advances on distributed representation shed light on learning low dimensional dense vectors to alleviate the data sparsity problem. Current studies on representation learning for POI recommendation embed both users and POIs in a common latent space, and users' preference is inferred based on the distance/similarity between a user and a POI. Such an approach is not in accordance with the semantics of users and POIs as they are inherently different objects. In this paper, we present a novel translation-based, time and location aware (TransTL) representation, which models the spatial and temporal information as a relationship connecting users and POIs. Our model generalizes the recent advances in knowledge graph embedding. The basic idea is that the embedding of a <time, location> pair corresponds to a translation from embeddings of users to POIs. Since the POI embedding should be close to the user embedding plus the relationship vector, the recommendation can be performed by selecting the top-$k$ POIs similar to the translated POI, which are all of the same type of objects. We conduct extensive experiments on two real-world datasets. The results demonstrate that our TransTL model achieves the state-of-the-art performance. It is also much more robust to data sparsity than the baselines.

**Keywords** point of interest (POI) recommendation, location-based social network (LBSN), time and location aware

## 1 Introduction

Location-based social networks (LBSNs), such as Foursquare①, Yelp②, and Facebook Places③, are becoming pervasive in our daily lives. Users on LBSN like to share their experiences with their friends for points of interest (POIs), e.g., restaurants and museums. The location-based service providers have collected a huge amount of users' check-in data, which facilitates the recommendation of POIs to unvisited users. The POI recommendation is of high value to both the users and companies, and thus has attracted much attention from researchers in recent years[1−5].

More importantly, while time and location together play a critical role in determining users' activities in LBSN, rare work has modeled their joint effects. Considering only one factor will deteriorate the predictive accuracy. For instance, whether a student may go to a school cafeteria or to a food court in a mall at lunch time depends on whether he/she is on campus or outside. It is not suggested for a system to recommend the same restaurant to a user in different places at the same

---

Regular Paper
This work was supported by the National Natural Science Foundation of China under Grant Nos. 61572376 and 91646206, and the National Key Research and Development Program of China under Grant No. 2016YFB1000603.
*Corresponding Author
①https://foursquare.com, Sept. 2018.
②https://www.yelp.com, Sept. 2018.
③https://www.facebook.com/places/, Sept. 2018.

time. This example shows the ineffectiveness of using one type of information but ignoring the other. However, taking both time and location into consideration exaggerates the data sparsity.

In this paper, we propose a novel translation-based, time and location aware (TransTL) model, which captures the joint effects of spatial and temporal information. Our model has the following distinct characteristics.

- TransTL takes location and time as a whole to determine the users' choice of POIs.
- TransTL embeds a spatiotemporal pair <time, location> as a relationship connecting users and POIs. By considering the time and the location simultaneously, our model can be successfully applied to real-time POI recommendation. Furthermore, distributed representations of TransTL are very effective in solving the problem of data sparsity.

Existing approaches that embed both users and POIs in a common latent space are unnatural because users and POIs are inherently different objects. In contrast, our TransTL model generalizes recent advances in knowledge graph embedding[6]. A user $u$ reaches an interested POI $v_q$ via an edge $tl$ denoting the <time, location> pair, i.e., $\vec{\boldsymbol{u}} + \vec{\boldsymbol{tl}} \approx \vec{\boldsymbol{v}}_q$. With this transformation, we can do recommendation for $u$ by selecting the top-$k$ POIs similar to POI $v_q$, which are all of the same type of objects with similar semantics.

TransRec by He *et al.*[7] represents the user as a relation vector to capture the transition from the previous item to the next item. This is the most recent work related to our proposed TransTL model. Both TransRec[7] and our TransTL are based on knowledge graph embedding technique[6–8]. However, TransRec mainly focuses on the sequential effects. Its rationale is that there is a sequential continuity between the previous item and the next item. This is reasonable in basket recommendation. However, in the POI recommendation, if the previous visited POI is far away from the next location, or the last check-in occurred several hours ago, such a sequential continuity will not take effects any more. Our TransTL model, on the other hand, is time- and location-aware, and thus is more appropriate for POI recommendation than TransRec.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 gives the problem definition and preliminary. Section 4 introduces our novel framework. Section 5 presents the experimental evaluation. Section 6 concludes the paper.

## 2 Related Work

Most existing studies mainly focused on leveraging spatial information due to the well-known strong correlation between users' activities and geographical distances[9–12]. For example, Ye *et al.*[13] proposed a Bayesian collaborative filtering (CF) algorithm to explore the geographical influence. Cheng *et al.*[14] captured the geographical influence by modeling the probability of a user's check-in on a location as a multi-center Gaussian model and then combined it into a generalized matrix factorization model. Lian *et al.*[15] adopted a weighted matrix factorization framework to incorporate the spatial clustering phenomenon.

Similar to the geo-spatial information, time is another important factor in POI recommendation. Ye *et al.*[16] found the periodic temporal property that people usually go to restaurants at around noon and visit clubs at night. Yuan *et al.*[17] developed a CF-based model to integrate temporal cyclic patterns. Cheng *et al.*[18] explored the temporal sequential patterns for personalized POI recommendation by using the transition probability of two successive check-ins of a user. Zhao *et al.*[19] proposed an Aggregated Temporal Tensor Factorization (ATTF) model for POI recommendation to capture the three temporal features together at different time scales.

Existing studies have exploited spatial or temporal influences mainly using CF[13,17] and Markov transition approaches[18]. Due to the sparsity of users' check-in records, it is hard to find similar users or to calculate transition probability. Although matrix factorization (MF) methods are effective in dealing with the sparsity in user-POI matrix[14,15,20], they do not consider the current location of the user.

Several recent studies[7,21–24] also exploited the power of distributed representation for alleviating data sparsity. The personalized ranking metric embedding (PRME) by Feng *et al.*[21] projects each POI and each user into a latent space, and then recommends a POI $v$ to a user $u$ at location $l$ based on the Euclidean distance between the POI and the user $\parallel \vec{\boldsymbol{u}} - \vec{\boldsymbol{v}} \parallel^2$ and that between the POI and the location $\parallel \vec{\boldsymbol{l}} - \vec{\boldsymbol{v}} \parallel^2$. Xie *et al.*[22] proposed a graph-based embedding model (GE) by embedding graphs into a shared low dimensional space, and then computed the similarity between a user $u$'s query $q$ at current time $t$ and location $l$ and a POI $v$ using an inner product, $S(q,v) = \vec{\boldsymbol{u}}^{\mathrm{T}} \cdot \vec{\boldsymbol{v}} + \vec{\boldsymbol{t}}^{\mathrm{T}} \cdot \vec{\boldsymbol{v}} + \vec{\boldsymbol{l}}^{\mathrm{T}} \cdot \vec{\boldsymbol{v}}$. GE is a unified model which integrates the spatial, temporal, and semantic effects by using POI-POI, POI-region, POI-time, and POI-word bi-partite graphs. Zhao *et al.*[23] proposed a spatial-temporal latent ranking (STELLAR) method to explicitly model the interactions among user, POI, and time. Zhao *et al.*[24] then proposed a Geo-Temporal sequential embedding rank (Geo-Teaser) model for POI recommendation. The Geo-Teaser model is a unified framework combining the temporal POI embedding model and the geographically hierarchical pairwise preference ranking model. While these approaches show significant improvements over many other baselines, they have the drawback that both users and POIs are embedded in a common latent space, and users' preference is inferred based on the distance/similarity between a user and a POI. Since users and POIs are inherently different objects, these embedding models are unnatural.

## 3 Problem Definition and Preliminary

**Definition 1** (POI). *A POI $v$ is defined as a unique identifier representing one specific position* (*e.g., a cafe or a hotel*), *and $V$ is a set of POIs, i.e.,* $V = \{v|v = (pid,\ position)\}$. *In paritcular, for POIs in Foursquare, there is additional tag information.*

**Definition 2** (Check-in Activity). *A check-in activity is a quadruple* $(u,\ t,\ l,\ v)$, *which means a user $u$ visits a POI $v$ in location $l$ at time $t$.*

**Definition 3** (Spatiotemporal Pattern). *A spatiotemporal pattern, denoted as $tl$, is a combination of a time slot $t$ and a location $l$ like* <11 a.m., Los Angeles>.

**Definition 4** (TL-Translation). *We define a TL-translation as the connection between user $u$ and POI $v$ corresponding to a spatiotemporal pattern. Specifically, a TL-translation means that in this situation* (*time $t$ and location $l$*) *$u$ tends to visit $v$.*

For ease of presentation, we summarize the notations in Table 1.

**Table 1.** Notations Used in This Paper

| Variable | Interpretation |
|---|---|
| $u$, $v$ | User $u$ and POI $v$ |
| $t$ | Time slot discretized from timestamp |
| $l$ | Location mapped from (longitude, latitude) |
| $tl$ | Spatiotemporal pattern $<t,\ l>$ |
| $\vec{\boldsymbol{u}}$, $\vec{\boldsymbol{tl}}$, $\vec{\boldsymbol{v}}$ | Embeddings of $u$, $(t,\ l)$, and $v$ |
| $u_q$, $t_q$, $l_q$ | Query user $u_q$, his/her current time $t_q$ and location $l_q$ |
| $v_q$ | Potential POI that query user $u_q$ is interested in |
| $D$ | Users' activity set $D = \{d|d = (u, t, l, v)\}$ |

The POI recommendation problem investigated in this paper has the same settings as that in [22]. Formally, the formal definition is as follows.

**Definition 5** (Location-Based Recommendation). *Given a dataset $D = \{d|d = (u, t, l, v)\}$ recording a set of users' activities, and a query $q = (u_q, t_q, l_q)$, we aim to recommend top-$k$ POIs in $V$ that the query user $u_q$ would be interested in.*

*Preliminary — KG Embedding.* The knowledge graph (KG) is a directed graph whose nodes and edges describe entities and their relations of the form (head, relation, tai), denoted as $(h,\ r,\ t)$. The goal of knowledge graph embedding is to learn a continuous vector space where the embeddings of entity and relation can preserve certain information of the graph. Bordes *et al.*[8] presented a simple yet effective approach TransE to learn vector embeddings for both entities and relations in KG. The basic idea is that the relationship between entities corresponds to a translation of the embeddings of entities, namely, $\vec{\boldsymbol{h}} + \vec{\boldsymbol{r}} \approx \vec{\boldsymbol{t}}$ when $(h\ ,r,\ t)$ exits in graph. Later, a model named TransH[25] was proposed to enable an entity to have distinct representations when it is involved in different relations.

Both TransE and TransH project all entities and relations into the same space. However, some entities may have multiple aspects and relations focusing on different aspects of the entities. Such entities are close in the entity space when they are similar, but they should be far away from each other in the relation space if they are strongly different in some specific aspects. To address this issue, Lin *et al.*[6] presented a TransR model to project two entities $h$ and $r$ of $(h,\ r,\ t)$ into a $r$-relation space as $h_r$ and $t_r$ with operation $\boldsymbol{M_r}$, such that $\vec{\boldsymbol{h_r}} + \vec{\boldsymbol{r}} \approx \vec{\boldsymbol{t_r}}$ holds in the relation space. A short description of TransR is as follows. For each triple $(h,\ r,\ t)$, two entities $h$ and $t$ are embedded into a $k$-dimensional entity space $R^k$ and the relation $r$ embedding is embedded into a $d$-dimensional relation space $R^d$. For each relation $r$, a projection matrix $\boldsymbol{M_r}$ is adopted to project the entities from entity space to relation space, and we get two projected vectors of entities, i.e., $h_r = h\boldsymbol{M_r}$, and $t_r = t\boldsymbol{M_r}$. The score function is then defined as $f_r(h,t) = ||\vec{\boldsymbol{h_r}} + \vec{\boldsymbol{r}} - \vec{\boldsymbol{t_r}}||_2$, and the objective is to minimize the score function.

## 4 Proposed Framework

We seek to learn the representations with the following characteristics.

• *Spatiotemporal Awareness.* Location and time together play a crucial role when a user selects a POI;

they should not be separated into individual ones.

• *Semantics Consistency*. All the POIs, either the query user's interested POI $v_q$ or all existing POIs $v \in V$, should come from a consistent semantic space.

In order to satisfy the first requirement, we combine each time slot and location as a spatiotemporal pattern $<t, l>$, and convert the quadruples $(u, t, l, v) \in D$ into triples $(u, <t, l>, v)$ in $D'$. We then learn representations for users, spatiotemporal patterns, and POIs from the converted set $D'$ to meet the second condition, using the translation technique originated from knowledge graph embedding.

### 4.1 TransTL Model

For the location-based recommendation problem, we focus on the connections between users and POIs corresponding to the spatiotemporal relations. Intuitively, if a POI $v$ is often visited by similar users in location $l$ at time $t$, the probability of a query user $u_q$ visiting $v$ with the same spatiotemporal relation will be high. On the other hand, users similar in the entity space may visit different POIs under distinct temporal and geographic conditions. In order to capture the strong correlations of users and POIs to the spatiotemporal patterns, we generalize the TransR technique[6] to fit the POI recommendation task. The basic idea is that a user $u$ will reach an interested POI $v_q$ via a translation edge $tl$, i.e., $\vec{\boldsymbol{u}} + \vec{\boldsymbol{tl}} \approx \vec{\boldsymbol{v}}_q$. Fig.1 illustrates the impacts of $tl$ patterns.

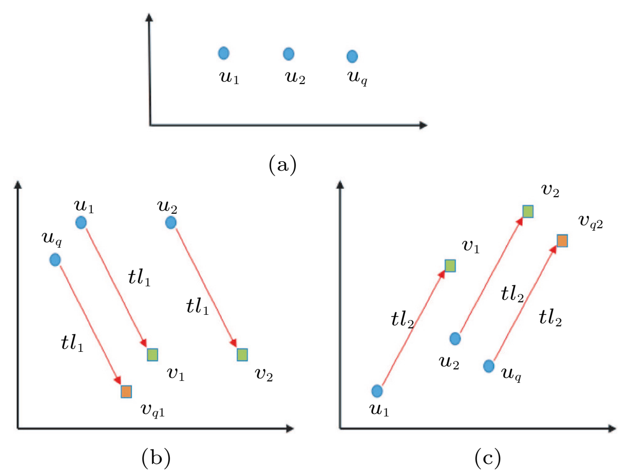



Fig.1. Impacts of spatiotemporal patterns. (a) Entity space. (b) Relation space of $tl_1$. (c) Relation space of $tl_2$.

In Fig.1, suppose $u_1$, $u_q$, and $u_2$ are three university students, $u_1$ and $u_q$ taking same courses, and $u_2$ and $u_q$ sharing the dormitory. Given two patterns $tl_1$ = <12 p.m., campus> and $tl_2$ = <8 p.m., dormitory>, the query user $u_q$ will be translated into two POIs $v_{q1}$ and $v_{q2}$; hence we should recommend for $u_q$ the POI $v_1$ in Fig.1(b) and the POI $v_2$ in Fig.1(c), which are the close neighbors of $v_{q1}$ and $v_{q2}$, respectively. The different recommending results $v_1$ and $v_2$ are caused by the effects of different spatiotemporal relations $tl_1$ and $tl_2$ respectively.

We now give the detail for TransTL. For each triple $(u, <t, l>, v)$ in $D'$, the user $u$, the spatiotemporal pair $<t, l>$ ($tl$ in short), and the POI $v$ correspond to the head entity $h$, the relationship edge $r$, and the tail entity $t$ in TransR, respectively. Their embeddings are set as $\vec{\boldsymbol{u}}, \vec{\boldsymbol{v}} \in R^d$, and $\vec{\boldsymbol{tl}} \in R^m$. For each spatiotemporal pair $tl$, we set a projection matrix $\boldsymbol{M}_{tl} \in R^{d\times m}$ to project a user embedding $\vec{\boldsymbol{u}}$ and a POI embedding $\vec{\boldsymbol{v}}$ in the original entity space to $\vec{\boldsymbol{u}}_{tl} = \vec{\boldsymbol{u}}\boldsymbol{M}_{tl}$ and $\vec{\boldsymbol{v}}_{tl} = \vec{\boldsymbol{v}}\boldsymbol{M}_{tl}$ in the relation space respectively, such that $\vec{\boldsymbol{u}}_{tl} + \vec{\boldsymbol{tl}} \approx \vec{\boldsymbol{v}}_{tl}$. This indicates that a POI embedding $\vec{\boldsymbol{v}}_{tl}$ should be the nearest neighbor of $\vec{\boldsymbol{u}}_{tl} + \vec{\boldsymbol{tl}}$. Temporal can be encoded as hourly, weekday hourly or weekend hourly slot depending on applications. Hence the score function can be defined as:

$$
\begin{aligned}
& s_{tl}(u, v) = \| \vec{\boldsymbol{u}}_{tl} + \vec{\boldsymbol{tl}} - \vec{\boldsymbol{v}}_{tl} \|_2^2, \\
& \text{s.t. } \| \vec{\boldsymbol{u}} \|_2 \leqslant 1, \| \vec{\boldsymbol{v}} \|_2 \leqslant 1, \| \vec{\boldsymbol{tl}} \|_2 \leqslant 1, \\
& \| \vec{\boldsymbol{u}}_{tl} \|_2 \leqslant 1, \| \vec{\boldsymbol{v}}_{tl} \|_2 \leqslant 1. \qquad (1)
\end{aligned}
$$

Given the score function defined in (1) for a triple $(u, tl, v)$, the entire objective function for training is as follows.

$$
\begin{aligned}
L = & \sum_{(u,\, tl,\, v)\in T} \sum_{(u',\, tl,\, v')\in T'} \max(0, s_{tl}(u, v) + \\
& \gamma - s_{tl}(u', v')), \qquad (2)
\end{aligned}
$$

where $\max(a,b)$ is used to get the maximum between $a$ and $b$, $\gamma$ is the margin, and $T$ and $T$' are the sets of correct and corrupted triples, respectively. The corrupted triples are generated by replacing the head and tail entities in correct triples using the same sampling method as that in [25]. More specifically, we assign different probabilities when replacing the user or POI entity. We also raise the chance of replacing the "one" side for the 1-to-$N$, $N$-to-1 and $N$-to-$N$ relations, such that the chance of generating false-negative instances will be reduced.

We adopt stochastic gradient descent (SGD) (in mini-batch mode) to minimize the objective function in (2). A small set of triplets is sampled from the training data. For each such triplet, we sample its corresponding incorrect triplets. All the correct and incorrect triples are put into a mini-batch. We compute the gradient and update the parameters after each mini-batch. When the iteration reaches a predefined number, we learn all the embeddings for users, POIs, and spatiotemporal patterns.

### 4.2 Recommendation Using TransTL

Once we have learned the embeddings, given a query user $u_q$ with the query time $t_q$ and location $l_q$, i.e., $q = (u_q, t_q, l_q)$, we first combine $t_q$ and $l_q$ as a spatiotemporal pattern $tl_q$, and then we can get the potential POI $v_q$ using (3).

$$\vec{\boldsymbol{v}}_q \boldsymbol{M}_{tl} = \vec{\boldsymbol{u}}_q \boldsymbol{M}_{tl} + \vec{\boldsymbol{tl}}_q. \tag{3}$$

The learned POI embedding $v_q$ naturally reflects the user's preference, because it encodes the user's past activities in $\vec{\boldsymbol{u}}_q$. It also captures the geographic and temporal influence in $\vec{\boldsymbol{tl}}_q$.

For each POI $v \in V$, we compute its distance to the POI $v_q$ in the normed linear space as defined in (4), and then select the $k$ POIs with the smallest ranking score as recommendations.

$$d(v, v_q) = \parallel \vec{\boldsymbol{v}} \boldsymbol{M}_{tl} - \vec{\boldsymbol{v}}_q \parallel_1. \tag{4}$$

We would like to emphasize our differences in computing $v_q$ and recommending POIs from those in [6, 22]. First, we can find an explicit POI $v_q$ directly from the latent space through the translation of the embedding of the spatiotemporal pattern on the user's embedding, while others compute an implicit $v_q$ by its distance/similarity to user $u_q$. Second, since the embeddings for POIs in $V$ are also from the same space, we can choose the ones which are the closest neighbors of $v_q$ in this space. This indicates that our recommended POIs are semantically consistent with the query user's interested POI $v_q$.

### 4.3 Dealing with Cold Start POIs

Considering the cold start POIs, which contain geographic and content information like tags but do not have any check-ins[22], we can simply extend our model to include the POI-POI relationship through the translation of content patterns. We call this model TransTL-C. The rationale is that if two POIs share a common tag or location, there will be a high degree of similarity between them, and their vector representations should be close to each other. Based on this observation, we define the score function as follows:

$$\begin{aligned} & s_{tlw}(u, v, s) \\ =\ & s_{tl}(u, v) + s_{wl}(v, s) \\ =\ & \parallel \vec{\boldsymbol{u}}_{tl} + \vec{\boldsymbol{tl}} - \vec{\boldsymbol{v}}_{tl} \parallel_2^2 + \parallel \vec{\boldsymbol{v}}_{wl} + \vec{\boldsymbol{wl}} - \vec{\boldsymbol{s_{wl}}} \parallel_2^2, \end{aligned} \tag{5}$$

where $s$ is a POI sharing at least one <word, location> pair with POI $v$. By combining (2) and (5) together, the objective function for cold start POIs is defined as:

$$\begin{aligned} LC = & \sum_{(u,tl,v)\in T} \sum_{(u',tl,v')\in T'} \max(0, s_{tl}(u, v) + \\ & \gamma - s_{tl}(u', v')) + \\ & \sum_{(v,wl,s)\in W} \sum_{(v',wl,s')\in W'} \max(0, s_{wl}(v, s) + \\ & \gamma - s_{wl}(v', s')). \end{aligned} \tag{6}$$

We once again use stochastic gradient descent to minimize the objective function $LC$ in (6). The only difference is the sampling procedure. For TransTL-C, since we have two types of edges, we sample the triplets ($u$, $tl$, $v$) and ($v$, $wl$, $s$) and their corresponding incorrect triples alternatively to update the model.

Our TransTL-C model proposed for dealing with cold start POIs can also be applied to the normal POI recommendation problem. However, it requires that those POIs should contain content information. For the recommendation on datasets like Gowalla, TransTL-C is not valid. Hence we only treat it as an extended model. Please also note that it is TransTL-C that uses the same information as GE does. Our standard TransTL model, on the other hand, uses less information than GE because it does not include the contents of POIs.

## 5 Experimental Evaluation

In this section, we first introduce the experimental setup and then compare our experimental results with those of baselines. Finally we show the performance of our method for addressing the data sparsity and cold start problem.

### 5.1 Experimental Setup

*Datasets.* We evaluate our method on two real-life LBSN datasets: Foursquare and Gowalla. A number of researchers have conducted experiments on data collected from these two social networks[3,4,17,22,26]. However, many of them are collected from various regions or in different time spans. For a fair comparison with

GE, we use the publicly available version④ provided by the authors of [22].

The two datasets have different scales such as geographic ranges, the number of users, POIs, and check-ins. Hence they are good for examining the performance of algorithms on various data types. Their statistics are listed in Table 2.

**Table 2.** Statistics of Foursquare and Gowalla

| | Foursquare | Gowalla |
|---|---|---|
| Number of users | 114 508 | 107 092 |
| Number of POIs | 62 462 | 1 280 969 |
| Number of check-ins | 1 434 668 | 6 442 892 |
| Number of standard time slots | 24 | 24 |
| Number of locations | 5 846 | 200 |
| Number of $<t, l>$ patterns | 28 868 | 3 636 |

Each check-in is stored as user-ID, POI-ID, POI-location in the form of latitude and longitude, check-in timestamp, and POI-content (only for Foursquare). In order to get the spatiotemporal patterns $<t, l>$ in Table 2, we use the same discretized method as that in [22], i.e., dividing time into 24 time slots which correspond to 24 hours, and the whole geographical space into a set of regions according to 5 846 administrative divisions (for Foursquare) and 200 regions clustered by a standard $k$-means method (for Gowalla). We finally get 28 868 and 3 636 $<t, l>$ pairs on Foursquare and Gowalla, respectively.

*Baselines* — {*GE, TransRec, TransTL-E, TransTL-H*}. We use GE[22] and TransRec[7], two state-of-the-art location-based recommendation approaches, as our baselines. GE adopts a graph-based embedding framework. It learns the embeddings based on POI-POI, POI-Time, POI-Location, and POI-Words graphs. By integrating the sequential, geographical, temporal cyclic, and semantic effect into a shared space, GE effectively overcomes the data sparsity problem and reaches the best performance so far. TransRec[7] represents the user as a relation vector to capture the transition from the previous item to the next item, and makes recommendation via the nearest neighbor search between the recommended item and the candidates.

We do not compare our method with other existing approaches because GE and TransRec have already significantly outperformed a number of baselines including JIM[27], PRME[21], and Geo-SAGE[28]. We thus only show our improvements over GE and TransRec.

Also note that although we choose the TransR[6] technique in knowledge graph embedding to materialize our TransTL model, the essence of our proposed framework is the translation of <time, location> pairs in the embedding space. This indicates that we do not rely on a specific translation model. Hence we can use TransE[8] and TransH[25] to translate <time, location> pairs in the embedding space. We denote the resulting methods as TransTL-E and TransTL-H, respectively.

*Settings.* We first organize the quadruples ($u$, $v$, $t$, $l$) in each dataset by users to get each user's profile $D_u$. We then rank the records in $D_u$ according to the check-in timestamps, and finally divide these ordered records into two parts: the first 80% as the training data, and the rest 20% as the test data. Moreover, the last 10% check-in records in the training data are used as a validation set for tuning the hyper-parameters.

We use the default settings in the original TransR[6] as the parameter settings for our TransTL model. Specifically, we set the learning rate $\lambda = 0.000\,1$, the margin $\gamma = 2$, the mini-batch size $B = 4\,800$, and the embedding dimensions $m = d = 100$, and we traverse over all the training data for 1 000 rounds. The same settings are also used for TransTL-E and TransTL-H models. The parameters for other baselines are listed as follows. For GE, the dimensionality is set to 100, and the number of negative sampling is 150 million. For TransRec, the learning rate $\epsilon$ is 0.05, the regularization hyperparameter $\lambda_\Theta$ is 0.1, the balance hyperparameter $\alpha$ is 0.2, and the dimensionality $K$ is 100. The values of these parameters are the same with those in their original papers.

We use the Accuracy@$k$ and Recall@$k$ ($k = \{1, 5, 10, 15, 20\}$) as our evaluation metrics. Accuracy@$k$ is based on hit@$k$. For a single test case, hit@$k$ is either the value 1, if the ground truth POI appears in the top-$k$ results, or the value 0, otherwise. The Accuracy@$k$ for each user is defined by averaging over all test cases of this user:

$$Acc@k = \frac{|hit@k|}{|D_{\text{test}}|}, \tag{7}$$

where $|hit@k|$ denotes the number of hits in the whole test set, and $|D_{\text{test}}|$ is the number of test cases.

Like other recommender systems, we sort the predicted scores of the candidate POIs and recommend the top-$k$ POIs to the target user. The Recall@$k$ for each user is defined as:

$$Rec@k = \frac{tp}{tp + tn}, \tag{8}$$

where $tp$ is the number of recommended POIs visited by a user $u$, and $tn$ is the number of recommended POIs visited by $u$ but not in the top-$k$ recommendations.

④https:/sites.google.com/site/dbhongzhi, Aug. 2018.

The average of recall and accuracy values for all test users is reported as the final Recall@$k$ and Accuracy@$k$ respectively. These two metrics are both in the range from 0 to 1 and a higher value means a better result.

### 5.2 Comparison with Baselines

For a fair comparison, we implement GE using the same LINE software provided by the authors of [29]. TransRec is implemented using the code provided by the authors of the original paper[7]. All the parameters for GE and TransRec are the same with those in [22] and [7], respectively. Both methods are run on our data division. We find a slightly difference (less than 1% in accuracy) between the original results and the results by our implemented GE. This is understandable and acceptable considering the randomness when we sample negative edges in LINE and initiate the centers of clusters of regions. All parameters for TransTL-E and TransTL-H use the default settings in [8] and [25] respectively. We present the comparison results in terms of accuracy and recall in Fig.2 and Fig.3, respectively.

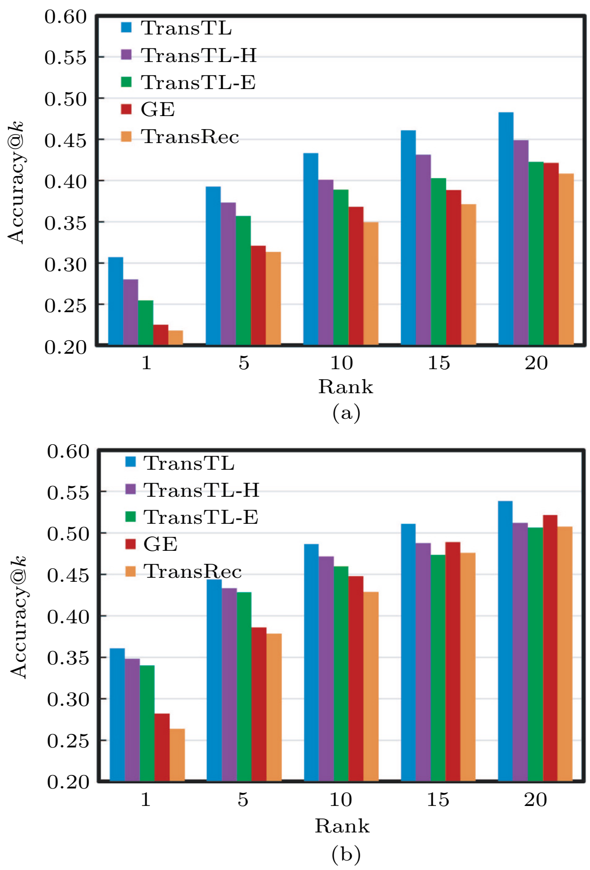



Fig.2. Comparisons with baselines in terms of accuracy. (a) Comparison on Foursquare(*Acc*@$k$). (b) Comparison on Gowalla(*Acc*@$k$).

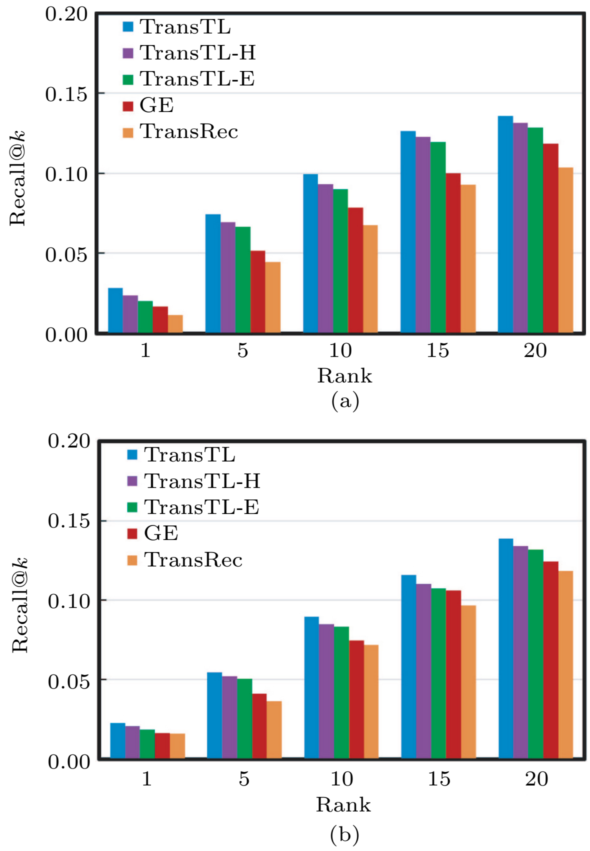



Fig.3. Comparisons with baselines in terms of recall. (a) Comparison on Foursquare(*Rec*@$k$). (b) Comparison on Gowalla(*Rec*@$k$).

From Fig.2 and Fig.3, it is clear that all our proposed TransTL-style models significantly outperform GE and TransRec in terms of accuracy and recall. For instance, the Accuracy@1 for TransTL, TransTL-H, and TransTL-E is 0.307, 0.280, 0.255, respectively, much better than 0.225 and 0.218 for GE and TransRec respectively. Similarly, we observe the Recall@1 for TransTL is 0.028 while that for GE and TransRec is 0.017 and 0.011, respectively. The superiority of our model over GE shows the effectiveness of our translation-based framework. More importantly, the comparison results between our TransTL style models and TransRec clearly demonstrate our modeling of time and location as the relation is much more reasonable than the modeling of TransRec.

While TransTL shows drastic improvement over GE and TransRec for all $k$s on Foursquare in Fig.2(a) and Fig.3(a), the trends on Gowalla in Fig.2(b) and Fig.3(b) are not that obvious. This is because there is a much smaller number of relations in Gowalla than in Foursquare. As shown in Table 2, Gowalla only has 3 636 relation patterns ($< t, l >$ pairs) while Foursquare has 28 868 pairs. Hence the learnt embeddings for entities and relations are worse than those on Foursquare, and incur the less accurate results when $k$ is large.

Besides the significant improvement over GE and TransRec (statistically significant at the 0.01 level for all $k$ settings), TransTL outperforms TransTL-H and TransTL-E as well. The reason is that TransR can differentiate the entities in the transformed relation space. Nevertheless, we see a less significant enhancement of TransTL over TransTL-H on Gowalla. This also conforms to the characteristics of the data: the graph of Gowalla is much larger but has less $tl$ relation edges than that of Foursquare, and the advantage of TransR over TransE is not obvious on such a dataset.

### 5.3 Effects of Model Parameters

This subsection shows the effects of model parameters. For clarity, we only show the results in terms of accuracy and omit those for recall which exhibit similar patterns. The effects of embedding dimension $d$ on Foursquare and Gowalla are shown in Table 3 and Table 4, respectively.

**Table 3.** Effects of Dimensionality on Foursquare

| $d$ | Accuracy | | | | |
|---|---|---|---|---|---|
| | $k$=1 | $k$=5 | $k$=10 | $k$=15 | $k$=20 |
| 70 | 0.281 | 0.376 | 0.409 | 0.433 | 0.451 |
| 80 | 0.294 | 0.384 | 0.417 | 0.445 | 0.462 |
| 90 | 0.300 | 0.390 | 0.425 | 0.459 | 0.476 |
| 100 | 0.307 | 0.393 | 0.434 | 0.461 | 0.483 |
| 110 | 0.311 | 0.407 | 0.439 | 0.463 | 0.486 |
| 120 | 0.312 | 0.407 | 0.439 | 0.464 | 0.486 |

**Table 4.** Effects of Dimensionality on Gowalla

| $d$ | Accuracy | | | | |
|---|---|---|---|---|---|
| | $k$=1 | $k$=5 | $k$=10 | $k$=15 | $k$=20 |
| 70 | 0.355 | 0.432 | 0.474 | 0.503 | 0.527 |
| 80 | 0.358 | 0.436 | 0.478 | 0.508 | 0.530 |
| 90 | 0.359 | 0.439 | 0.482 | 0.509 | 0.535 |
| 100 | 0.361 | 0.445 | 0.486 | 0.511 | 0.539 |
| 110 | 0.361 | 0.445 | 0.488 | 0.513 | 0.540 |
| 120 | 0.361 | 0.445 | 0.488 | 0.513 | 0.540 |

We can see that the experimental results are not very sensitive to the dimension $d$. With an increasing number of dimension, the accuracy on Gowalla is almost unchanged, i.e., the improvement is less than 1% in nearly all cases. The accuracy on Foursquare is slightly enhanced with a large dimension $d$, and finally it becomes stable.

To investigate the effects of time interval, we divide timestamps by three methods, i.e., splitting time into 24, 7, and 2 time slots, corresponding to the hourly, daily, and weekday/weekend patterns, respectively. Fig.4 shows the effects of various time intervals. We observe that the impact of the hourly patterns is the most significant on both datasets. In addition, the results for different patterns vary widely, suggesting a good strategy for dividing the time slot is important.

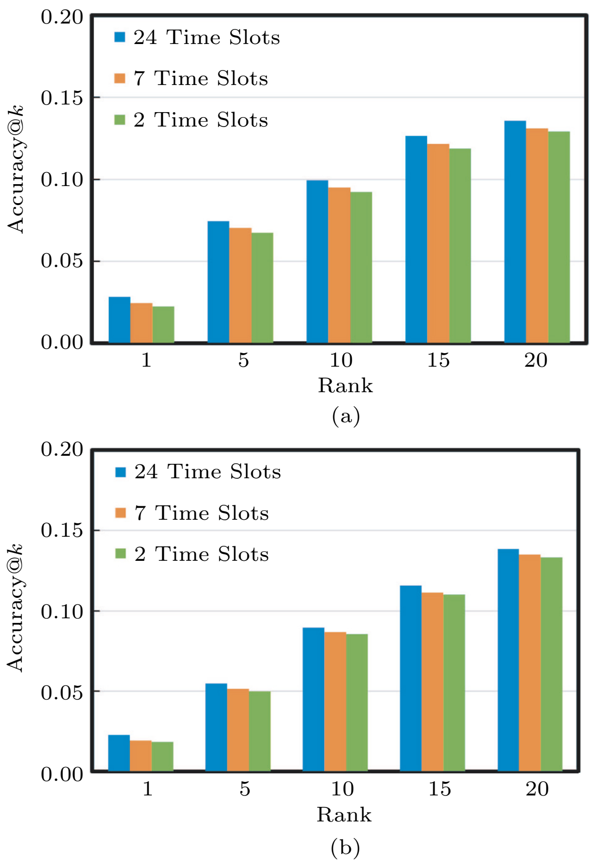



Fig.4. Effects of time interval. (a) Time interval effects on Foursquare ($Acc@k$). (b) Time interval effects on Gowalla($Acc@k$).

### 5.4 Sensitivity to Data Sparsity

In order to compare the sensitivity to the data sparsity of our proposed TransTL model with that of GE and TransRec, we conduct extensive experiments to evaluate the performance on two datasets by reducing training data. More precisely, we keep the testing dataset unchanged and reduce the training data randomly by a ratio of 5% to 20% stepped by 5. In this paper, we only present the results by reducing 20% training data on Foursquare and Gowalla in terms of accuracy and recall in Table 5–Table 6, and Table 7–Table 8, respectively. The trends with other ratios are

**Table 5**. Sensitivity to Sparsity on Foursquare in Terms of Accuracy

| $k$ | GE | | | TransRec | | | TransTL | | |
|---|---|---|---|---|---|---|---|---|---|
| | + | − | ↓ (%) | + | − | ↓ (%) | + | − | ↓ (%) |
| 1 | 0.225 | 0.154 | −31.69 | 0.218 | 0.102 | −53.41 | **0.307** | **0.246** | **−19.99** |
| 5 | 0.321 | 0.228 | −28.84 | 0.313 | 0.193 | −38.39 | **0.393** | **0.320** | **−18.46** |
| 10 | 0.369 | 0.270 | −26.82 | 0.350 | 0.229 | −34.59 | **0.434** | **0.365** | **−15.86** |
| 15 | 0.388 | 0.295 | −23.95 | 0.372 | 0.247 | −33.62 | **0.461** | **0.382** | **−17.04** |
| 20 | 0.422 | 0.318 | −24.66 | 0.409 | 0.271 | −33.74 | **0.483** | **0.407** | **−15.74** |

**Table 6**. Sensitivity to Sparsity on Foursquare in Terms of Recall

| $k$ | GE | | | TransRec | | | TransTL | | |
|---|---|---|---|---|---|---|---|---|---|
| | + | − | ↓ (%) | + | − | ↓ (%) | + | − | ↓ (%) |
| 1 | 0.017 | 0.014 | −18.93 | 0.011 | 0.010 | −15.93 | **0.028** | **0.027** | **−6.69** |
| 5 | 0.052 | 0.048 | −6.21 | 0.045 | 0.039 | −13.45 | **0.074** | **0.072** | **−3.09** |
| 10 | 0.078 | 0.064 | −18.01 | 0.068 | 0.055 | −18.37 | **0.100** | **0.095** | **−4.92** |
| 15 | 0.100 | 0.096 | −4.30 | 0.093 | 0.087 | −6.58 | **0.126** | **0.124** | **−1.82** |
| 20 | 0.119 | 0.115 | −3.20 | 0.104 | 0.094 | −9.36 | **0.136** | **0.133** | **−2.43** |

**Table 7**. Sensitivity to Sparsity on Gowalla in Terms of Accuracy

| $k$ | GE | | | TransRec | | | TransTL | | |
|---|---|---|---|---|---|---|---|---|---|
| | + | − | ↓ (%) | + | − | ↓ (%) | + | − | ↓ (%) |
| 1 | 0.282 | 0.209 | −25.77 | 0.264 | 0.175 | −33.78 | **0.361** | **0.291** | **−19.36** |
| 5 | 0.386 | 0.303 | −21.67 | 0.379 | 0.252 | −33.38 | **0.445** | **0.384** | **−13.72** |
| 10 | 0.448 | 0.354 | −20.98 | 0.429 | 0.313 | −27.18 | **0.486** | **0.415** | **−14.64** |
| 15 | 0.489 | 0.396 | −19.13 | 0.476 | 0.348 | −26.93 | **0.511** | **0.445** | **−13.01** |
| 20 | 0.521 | 0.423 | −18.91 | 0.508 | 0.385 | −24.15 | **0.539** | **0.468** | **−13.15** |

**Table 8**. Sensitivity to Sparsity on Gowalla in Terms of Recall

| $k$ | GE | | | TransRec | | | TransTL | | |
|---|---|---|---|---|---|---|---|---|---|
| | + | − | ↓ (%) | + | − | ↓ (%) | + | − | ↓ (%) |
| 1 | 0.017 | 0.010 | −37.35 | 0.016 | 0.007 | −55.90 | **0.023** | **0.022** | **−4.82** |
| 5 | 0.041 | 0.038 | −8.74 | 0.036 | 0.031 | −16.21 | **0.055** | **0.053** | **−4.01** |
| 10 | 0.075 | 0.070 | −6.71 | 0.072 | 0.066 | −7.95 | **0.090** | **0.087** | **−2.46** |
| 15 | 0.106 | 0.102 | −4.33 | 0.097 | 0.092 | −5.17 | **0.116** | **0.114** | **−1.99** |
| 20 | 0.124 | 0.120 | −3.78 | 0.118 | 0.113 | −4.73 | **0.139** | **0.137** | **−1.37** |

all alike. Note that the higher accuracy (recall) and the smaller change, the better (shown in bold). Also note that +, −, ↓ denote the original, 20% less training data, and change ratio, respectively.

We have the following important notes for Tables 5–8. With the reduction of training data, the accuracy values for GE, TransRec, and TransTL all decrease. However, TransTL always achieves the best results at different $k$ values on two datasets in terms of both metrics.

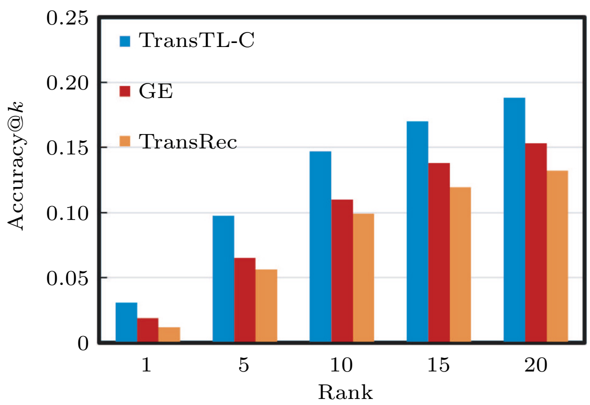
The reduction of accuracy of our TransTL model is much smaller than that of GE and TransRec. For instance, the Accuracy@1 of GE and TransRec shows a 31.69% and 53.41% drop, respectively. In contrast, our TransTL model only has a 19.99% change. This strongly suggests that our model is more robust to the data sparsity.

TransRec is the most sensitive approach. The reasons may be two-fold. Firstly, compared with GE, it lacks the integration of various kinds of social, geographical, and semantic information. Secondly, compared with TransTL, its dependency on previous item makes it more sensitive to the reduced POI embeddings.

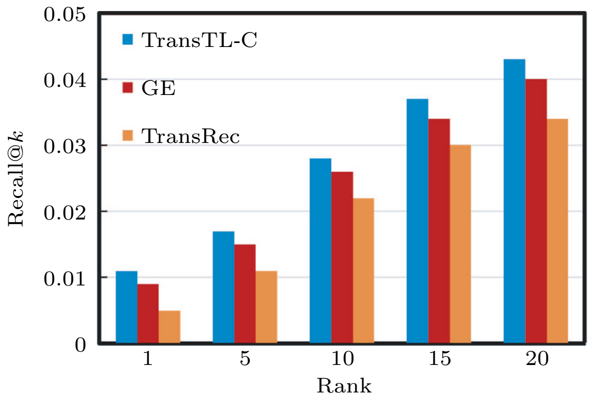
The declination of accuracy on Foursquare is more obvious than that on Gowalla. The reason may be that Foursquare is much sparser in users' check-ins than Gowalla, hence reducing the training data has a greater impact on Foursquare.

### 5.5 Test for Cold Start Problem

In this experiment, we further compare the effectiveness of our extended TransTL-C model with that of GE and TransRec when addressing the cold-start problem. The cold-start POIs are defined as those visited by less than five users[26]. To test the performance of cold-start POI recommendations, we select users who have at least one cold-start check-in as test users. For each test user, we choose his/her check-in records associated with cold-start POIs as test data and the remains as training data. Since there is no content information for POIs in Gowalla, we conduct experiments, just as GE does, only on Foursquare. The results are shown in Fig.5.




(a)




(b)

Fig.5. Test for cold start problem on Foursquare. (a) Test for cold start problem (*Acc*@$k$). (b) Test for cold start problem (*Rec*@$k$).

From Fig.5, it is clear that our proposed TransTL-C model consistently beats GE and TransRec in terms of both metrics when recommending cold-start POIs. The superior performance of TransTL-C model is due to the translation of content and geography information $wl$ from an ordinary POI $v$ to a cold-start POI $v_c$. As long as there is an existing $v$ sharing one <word, location> pair with $v_c$, our TransTL-C model can get a translation for $v_c$. In contrast, GE utilizes the bipartite graphs of POI-Word and POI-Location. The weight of an edge in the graph is calculated by a TF-IDF value of the word or the frequency of a location. The edge weight is proportional to the probability of edge sampling. Since there are few check-in records for cold-start POIs, a $v_c$-word and $v_c$-location edge has an extremely rare chance to be selected and updated. Similarly, TransRec requires the embedding of a previous check-in, which is scare in the case of the cold-start scenario. Consequently, the learnt embedding for $v_c$ in GE and TransRec will be poor and further deteriorates the recommendation accuracy.

## 6 Conclusions

We presented a novel translation-based, spatiotemporal aware model TransTL for learning representations of users, spatiotemporal patterns, and POIs. The basic idea is to capture the geographic and temporal effects using a <time, location> pair, and then model it as a translation connecting users and POIs. We realized TransTL using the knowledge graph embedding technique. Our method has two distinguished advantages. 1) We learned a joint representation for spatiotemporal patterns whose components contribute together to a user's choice in POIs. 2) The translation mechanism enables the learnt POI embeddings to be in the same semantic space with that of the query POI.

We conducted extensive experiments on two real-life datasets. Our results showed that TransTL achieves the state-of-the-art performance in recommendation accuracy. It also significantly outperforms the baselines in terms of the effectiveness in addressing both the data sparsity and cold-start problems.

## References

[1] Cheng C, Yang H Q, King I, Lyu M R. A unified point-of-interest recommendation framework in location-based social networks. *ACM Transactions on Intelligent Systems and Technology*, 2016, 8(1): Article No. 10.

[2] Zhu W Y, Peng W C, Chen L J, Zheng K, Zhou X F. Modeling user mobility for location promotion in location-based social networks. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.1573-1582.

[3] Chen X F, Zeng Y F, Cong G, Qin S C, Xiang Y P, Dai Y S. On information coverage for location category based point-of-interest recommendation. In *Proc. the 29th AAAI*

*Conference on Artificial Intelligence*, January 2015, pp.37-43.

[4] Gao H J, Tang J L, Hu X, Liu H. Content-aware point of interest recommendation on location-based social networks. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, January 2015, pp.1721-1727.

[5] Qiao S J, Han N, Zhou J L, Li R H, Jin C Q, Gutierrez L A. SocialMix: A familiarity-based and preference-aware location suggestion approach. *Engineering Applications of Artificial Intelligence*, 2018, 68: 192-204.

[6] Lin Y K, Liu Z Y, Sun M S, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, January 2015, pp.2181-2187.

[7] He R N, Kang W C, McAuley J. Translation-based recommendation. In *Proc. the 11th ACM Conference on Recommender Systems*, August 2017, pp.161-169.

[8] Bordes A, Glorot X, Weston J, Bengio Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 2014, 94(2): 233-259.

[9] Zheng Y, Zhang L Z, Xie X, Ma W Y. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. the 18th International Conference on World Wide Web*, April 2009, pp.791-800.

[10] Ye M, Yin P F, Lee W C. Location recommendation for location-based social networks. In *Proc. the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 2010, pp.458-461.

[11] Cho E, Myers S A, Leskovec J. Friendship and mobility: User movement in location-based social networks. In *Proc. the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2011, pp.1082-1090.

[12] Yan B, Janowicz K, Mai G, Gao S. From ITDL to place2Vec — Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proc. the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 2017, Article No. 35.

[13] Ye M, Yin P F, Lee W C, Lee D L. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc. the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2011, pp.325-334.

[14] Cheng C, Yang H Q, King I, Lyu M R. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proc. the 26th AAAI Conference on Artificial Intelligence*, July 2012, pp.17-23.

[15] Lian D F, Zhao C, Xie X, Sun G Z, Chen E H, Rui Y. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2014, pp.831-840.

[16] Ye M, Janowicz K, Mülligann C, Lee W C. What you are is when you are: The temporal dimension of feature types in location-based social networks. In *Proc. the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 2011, pp.102-111.

[17] Yuan Q, Cong G, Ma Z Y, Sun A X, Thalmann N M. Time-aware point-of-interest recommendation. In *Proc. the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2013, pp.363-372.

[18] Cheng C, Yang H Q, Lyu M R, King I. Where you like to go next: Successive point-of-interest recommendation. In *Proc. the 23rd International Joint Conference on Artificial Intelligence*, August 2013, pp.2605-2611.

[19] Zhao S L, Lyu M R, King I. Aggregated temporal tensor factorization model for point-of-interest recommendation. In *Proc. the 23rd International Conference on Neural Information Processing*, October 2016, pp.450-458.

[20] Zhao S L, King I, Lyu M R. Geo-pairwise ranking matrix factorization model for point-of-interest recommendation. In *Proc. the 24th International Conference on Neural Information Processing*, November 2017, pp.368-377.

[21] Feng S S, Li X T, Zeng Y F, Cong G, Chee Y M, Yuan Q. Personalized ranking metric embedding for next new POI recommendation. In *Proc. the 24th International Conference on Artificial Intelligence*, July 2015, pp.2069-2075.

[22] Xie M, Yin H Z, Wang H, Xu F J, Chen W T, Wang S. Learning graph-based POI embedding for location-based recommendation. In *Proc. the 25th ACM International on Conference on Information and Knowledge Management*, October 2016, pp.15-24.

[23] Zhao S L, Zhao T, Yang H Q, Lyu M R, King I. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Proc. the 30th AAAI Conference on Artificial Intelligence*, February 2016, pp.315-322.

[24] Zhao S L, Zhao T, King I, Lyu M R. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proc. the 26th International Conference on WWW*, April 2017, pp.153-162.

[25] Wang Z, Zhang J W, Feng J L, Chen Z. Knowledge graph embedding by translating on hyperplanes. In *Proc. the 28th AAAI Conference on Artificial Intelligence*, July 2014, pp.1112-1119.

[26] Yin H Z, Zhou X F, Cui B, Wang H, Zheng K, Hung N Q. Adapting to user interest drift for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(10): 2566-2581.

[27] Yin H, Zhou X F, Shao Y X, Wang H, Sadiq S. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proc. CIKM*, October 2015, pp.1631-1640.

[28] Wang W Q, Yin H Z, Chen L, Sun Y Z, Sadiq S, Zhou X F. Geo-SAGE: A geographical sparse additive generative model for spatial item recommendation. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.1255-1264.

[29] Tang J, Qu M, Wang M Z, Zhang M, Yan J, Mei Q Z. LINE: Large-scale information network embedding. In *Proc. the 24th International Conference on WWW*, May 2015, pp.1067-1077.

**Tie-Yun Qian** received her Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, in 2006. She is a professor and Ph.D. supervisor at Wuhan University, Wuhan. Her research interests include Web mining, data management, etc. She has published over 40 papers on leading journals and conferences like ACL, EMNLP, COLING, SIGIR, CIKM, and INS. She is a member of CCF, ACM, and IEEE.

**Bei Liu** received her M.S. degree in computer technology from Wuhan University, Wuhan, in 2018. Her research interests include web mining, data management, etc.

**Liang Hong** received his B.S. and Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, in 2003 and 2009, respectively. Now, he is an associate professor in the School of Information Management, Wuhan University, Wuhan. His research interests include knowledge graph, spatio-temporal data management, and social networks. He is a senior member of CCF, and a member of ACM and IEEE.

**Zhen-Ni You** received her M.S. degree in computer technology from Wuhan University, Wuhan, in 2018. Her research interests include web mining, data management, etc.