A General Multi-Context Embedding Model for Mining Human Trajectory Data

Ningnan Zhou, *Student Member, IEEE*, Wayne Xin Zhao, *Member, IEEE*, Xiao Zhang, Ji-Rong Wen, *Senior Member, IEEE*, and Shan Wang, *Member, IEEE*

Abstract—The proliferation of location-based social networks, such as Foursquare and Facebook Places, offers a variety of ways to record human mobility, including user generated geo-tagged contents, check-in services, and mobile apps. Although trajectory data is of great value to many applications, it is challenging to analyze and mine trajectory data due to the complex characteristics reflected in human mobility, which is affected by multiple contextual information. In this paper, we propose a Multi-Context Trajectory Embedding Model, called MC-TEM, to explore contexts in a systematic way. MC-TEM is developed in the distributed representation learning framework, and it is flexible to characterize various kinds of useful contexts for different applications. To the best of our knowledge, it is the first time that the distributed representation learning methods apply to trajectory data. We formally incorporate multiple context information of trajectory data into the proposed model, including user-level, trajectory-level, location-level, and temporal contexts. All the context information is represented in the same embedding space. We apply MC-TEM to two challenging tasks, namely location recommendation and social link prediction. We conduct extensive experiments on three real-world datasets. Extensive experiment results have demonstrated the superiority of our MC-TEM model over several state-of-the-art methods.

Index Terms—Human trajectory, distributed representation, contextual information, location recommendation, social link prediction

1 Introduction

With the permeation of positioning-enabled mobile devices, the proliferation of location-based social networks, such as Foursquare and Facebook Places, offers a variety of ways to record human mobility, including user generated geo-tagged contents (e.g., tweets, photos on Instagram), check-in services (e.g., Foursquare's check-ins) and mobile Apps (e.g., Highlight, Glancee). Most importantly, location based social networks have been becoming the digital mirror to human mobility in physical world, which provides an opportunity to deeply understand people's behavior. For example, users tend to have different spatial and temporal activity preference as their lifestyles [50].

Since trajectory data reflects human mobility, it can be naturally utilized for location-based recommendation applications, including personalized location prediction [29], [49], group-based location (or *event*) recommendation [51] and user mobility modelling [53]. Further, previous studies [10], [39] have shown that publicly published spatiotemporal data, such as users' check-in records, strongly indicates social strength among users, which is the likelihood for users to be socially related and can be utilized for link prediction.

- *N. Zhou, X. Zhang, and S. Wang are with the School of Information, Renmin University of China, China. E-mail: zhouningnan@gmail.com, {zhangxiao, swang}@ruc.edu.cn.*
- *W.X. Zhao and J. Wen are with the School of Information, Renmin University of China, and the Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China. E-mail: {batmanfly, jirong.wen}@gmail.com.*

Manuscript received 18 Aug. 2015; revised 6 Feb. 2016; accepted 19 Mar. 2016. Date of publication 5 Apr. 2016; date of current version 5 July 2016. Recommended for acceptance by F. Bonchi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2550436

Although it is of great practical value, it is challenging to analyze and mine trajectory data due to the complex characteristics reflected in human mobility. Even for a single check-in record, the generative process is typically related to several important factors, including user's, regional and categorical preference [49], [53]. In specific, the most recent study [6] has proposed a multi-dimensional tensor factorization model, which exploits multiple kinds of contexts including who, what, when and where. As a tensor factorization model, it is more suitable to capture the direct interactions among multiple dimensions, while surrounding contexts in a trajectory cannot be directly modeled, e.g., the previous and successive locations of a check-in location in a trajectory. Trajectory data itself is a kind of sequential data, and such surrounding contexts are important to consider for trajectory modeling [9]. Another issue with previous studies is that the incorporation of more new contexts will subtantially increase model complexity, e.g., tensor decomposition. Hence, a more flexible way to characterize multiple kinds of contextual information is needed for modeling trajectory data.

In this paper, we propose a general Multi-Context Trajectory Embedding Model (MC-TEM), which provides a flexible way to characterize and leverage multiple contexts. Inspired by the recent progress in deep learning and neural networks [31], [40], we propose to use the distributed representation method for modeling trajectory data. In our model, each contextual feature corresponds to a unique distributed vector (a.k.a., embedding vector). Further, a *contextual vector* can be averaged or concatenated using the embedding vectors of contextual features. A location itself is associated with an embedding vector, which is further generated based on its contextual vector by using a softmax probability function. As a result, the model is general and flexible to characterize arbitrary contextual features in a

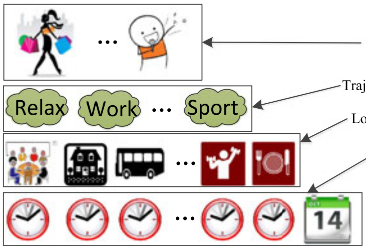
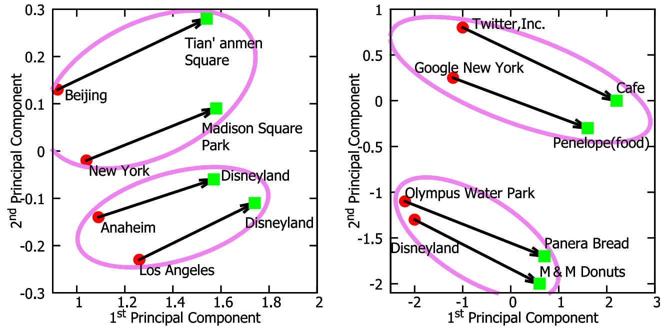


Fig. 1. An illustrative example of the contexts used in our approach. User-level contexts characterize user preference such as shopaholic or music fan, trajectory-level contexts capture trajectory intent such as relax or work, location-level contexts include surrounding location/category information, and temporal contexts include day and hour information.



(a) Type I: City *to* Location (b) Type II: Location *to* Location

Fig. 2. Identified examples for the pattern "A_1-A_2 $\approx$ B_1-B_2" using the embedding vectors learnt on the Foursquare$_S$ [49] dataset. Each arrow line is associated with a pair of two entities. Each two corresponding entities pairs is in a ellipsoidal circle, indicating they have the approximately equal distance. For ease of visualization, PCA is used for dimensionality reduction, i.e., the first two principle components are used. See details in Section 5.4.

unified manner. Different from traditional one-hot representation [39], the distributed representation model uses dense embedding representations for various contexts, which is more capable of uncovering hidden knowledge in trajectory data such as fine-grained location relatedness. As we will show, the proposed model is flexible and effective to apply to application tasks related to trajectory data. To instantiate the proposed model, we have presented a systematic selection of various useful contextual features, including user-level, trajectory-level, location-level and temporal contexts. Fig. 1 gives an illustrative example of the contextual information for trajectory generation.

In a general sense, embedding (a.k.a. distributed representation) aims to project an informational entity into a low-dimensional vector space. Distinguishing itself from traditional one-hot representation, embedding adopts a "dense" real vector for information represenation, which has been shown to yield better performance and more resistant to noise in many fields [40]. We take the initiative to apply distributed representation to model trajectory data. Tailored to trajectory data, our model is flexible to model any contextual features in the embedding space, and further characterize check-in sequences in a generative process. With the embedding representations, it is particularly convenient to analyze the association of multiple contexts. Similar to [31], as shown in Fig. 2, we have derived several interesting findings based on the learnt embedding vectors, e.g., `Beijing - Tian'anmen Square` $\approx$ `Los Angeles - Disneyland`. Extensive experiments on real applications using real-world datasets have demonstrated the effectiveness of MC-TEM, which indicates that the distributed representation approach is promising to deal with trajectory data.

2 Related Works

Our work is related to the following two research areas.

2.1 Mining Trajectory Data

Trajectory pattern mining. It aims to find interesting trajectory patterns from both visiting sequences and movement durations. Originally, these ideas have been borrowed from sequential pattern mining in transactional databases [1], [55], and many efficient algorithms are proposed, such as PrefixSpan [37] and CloseSpan [46]. Subsequently, researchers also propose to find the most frequent paths with the time period constraint [28]. Furthermore, spatial-temporal pattern mining is proposed to incorporate both regions of interest and travel time between movements [17]. Following this line, close locations frequently visited by the same group of persons in a time period are clustered to form regions [18], [20], [21]. Scalable algorithms based on pattern-growth and Aprior algorithm [2], [17] have also been proposed.

Link prediction using trajectory data. It aims to infer the likelihood for a pair of users to be friends. The correlation between mobile homophily and social ties has been investigated [14], [43]. Early methods first represented each trajectory in a hierarchial fashion, and then a graph-based sequence matching algorithm was used to measure the social tie of two users based on trajectory similarity [25]. For user pairs who do not share well-matched sequences, users' location category sequences are used to develop a sequence matching algorithm for inferring their social ties [45]. As a fundamental basis, general statistical characteristics of individual human trajectories have been investigated [36], [42]. Following this line, human mobility factors are further utilized for link prediction [10], [43]. They have proposed several high-quality mobile homophily measures, such as co-location rate and spatial cosine similarity, and yield practically predictive power. More recently, several studies alsoexplore other measures such as the diversity of co-occurrence and location entropy [12], [39].

Location recommendation. It aims to recommend locations that a user is likely to visit based on users' historical trajectory information. Existing methods usually exploit geographical influence, social influence, content information and temporal effect. Geographical and social influence indicate that people tend to explore nearby locations of a location that they or their friends have visited before [8], [24], [48]. Several studies jointly incorporate geographical and social influence, including CF based methods [48] and MF (matrix factorization) based methods [8], [26]. In addition, content information is explored to alleviate data sparsity. In [49], location category labels are leveraged to explore the personal interests and local preferences; tags and user comments also show good performance in aggregated LDA and MF models [15], [27] respectively. For time-aware location recommendation task, which recommends locations for a user at a given time, temporal effect is highly emphasized. Early work extends user-based CF to incorporate both the temporal and geographical effects with a linear combination framework [52]. Further, a graph based method is proposed to propagate preference on a graph constructed from the check-in data [54]. More recently, W^4 leverages tensor factorization to enable multi-dimensional collaborative recommendation for Who (user), What (location category), When (time) and Where (location) on sparse data [6], which is an multi-dimensional extension of the method in [56].

2.2 Distributed Representation Learning

Recently, distributed representation learning has been successfully applied to various fields, such as Natural Language Processing, Speech Recognition and Signal Processing, since its seminal work [40].

Particularly, in NLP field, the word embedding model `word2vec` developed in [31], [32] makes it possible to train the embedding vectors on large-scale datasets with a single machine. Stochastic Gradient Descent (SGD) is used to train the parameters with two alternative optimized algorithms for speedup, namely hierarchical softmax [34] and noise contrastive estimation [33]. Several variants based on `word2vec` have been proposed for different applications, including phrase mining [32], paragraph modelling [23], smart phone application recommendation [3], and next-basket recommendation [44]. To preserve the neighborhood structure in a network, a truncated random walk is proposed to determine the context vertices for each vertex [38].

To the best of our knowledge, distributed representation methods on trajectory data have seldom been studied. As we will see in experiments, direct application of existing models (e.g., `paragraph2vec`) may not work well on trajectory data due to complex data characteristics. We present a general formulation to incorporate multiple kinds of contextual information, and it is the first time that distributed representation models apply to the trajectory data. We carefully study how to extract and model various contexts. More importantly, we have shown how to apply the proposed model to two important applications. Our experimental results also give several important and interesting observations, which are useful to future studies in this line.

3 A General Multi-Context Trajectory Embedding Model

Before introducing the model, we first present the preliminary used throughout the paper.

Definition 1 (Check-in record). *When a user u checks in a location ℓ with a category label c at the timestamp s, the check-in record can be modeled as a quadruple $\langle u, \ell, c, s\rangle$.*

An example check-in record can be (`UID25821, Burger King@BH Point, Restaurant, 2015-01-13/1:30pm`), which tells that a user with the ID of $25,821$ has visited the restaurant "Burger King@BH Point" at 1:30 pm on January 13, 2015.

Definition 2 (Trajectory). *Given a user u, a trajectory t is a sequence of chronologically ordered check-in records related to u: $\langle u, \ell_1, c_1, s_1\rangle, \ldots, \langle u, \ell_i, c_i, s_i\rangle, \ldots, \langle u, \ell_N, c_N, s_N\rangle$, where N is the sequence length and $s_i < s_{i+1}$ for $i \leq N-1$.*

TABLE 1
Notations and Descriptions

Notations	Descriptions
u, t, r	index variable for user, trajectory and city (region)
ℓ, c, s	index variable for location, category and timestamp
d, h	index variable for day and hour
N_t	the length of trajectory t
$\mathcal{U}$	the set of users
$\mathcal{T}^{(u)}$	the set of trajectories generated by user u
$\mathcal{L}$	the set of locations
$\mathcal{C}$	the set of category labels
$\mathcal{C}^{(\ell)}$	the set of category labels attached to ℓ
$\boldsymbol{v}$	an embedding vector
$\boldsymbol{v}_f$	the embedding vector for a contextual feature f
$\bar{\boldsymbol{v}}_\ell$	the averaged contextual embedding vector for location ℓ
$\boldsymbol{v}_u$, $\boldsymbol{v}_t$, $\boldsymbol{v}_\ell$	embedding vectors for user, trajectory, location
$\boldsymbol{v}_c$, $\boldsymbol{v}_d$, $\boldsymbol{v}_h$, $\boldsymbol{v}_r$	embedding vectors for category, day, hour and city
CL	the context window length
VS	the embedding dimension size
U, M, T, C	numbers of users, locations, trajectories and categories

As a user can generate multiple trajectories, we use a trajectory set $\mathcal{T}^{(u)}$ to denote all the trajectories from u.[1] Table 1 presents the notations used in this paper.

3.1 Overview of the Proposed Model

Now we develop our model, a general Multi-Context Trajectory Embedding Model, denoted by ***MC-TEM***. We aim to provide a *general* and *flexible* manner to characterize multiple kinds of contextual information for trajectory data. We start with a single trajectory. Given a trajectory sequence $\langle u, \ell_1, c_1, s_1\rangle, \ldots, \langle u, \ell_i, c_i, s_i\rangle, \ldots, \langle u, \ell_N, c_N, s_N\rangle$, the overall objective function is to maximize the average log probability for each location given its corresponding contextual information:

$$\frac{1}{N}\sum_{j=1}^{N} \log Pr(\ell_j|\boldsymbol{x}^{(\ell_j)}), \tag{1}$$

where the $\boldsymbol{x}^{(\ell_j)}$ is a real-valued feature vector consisting of all the contextual information for the *target location* ℓ_j. Each dimension in $\boldsymbol{x}^{(\ell_j)}$ corresponds to a unique contextual feature and $x_f^{(\ell_j)}$ is the weight for the fth feature in $\boldsymbol{x}^{(\ell_j)}$. For simplicity, we further assume that $\boldsymbol{x}^{(\ell_j)}$ is a non-negative vector, each entry of which denotes the number of occurrences for a feature in the *context* correspondingly. Trajectory data itself is complex and related to multiple types of information. Hence, we need to seek a flexible way for modeling trajectory data. Inspired by the recent progress of deep learning and neural networks [31], [32], we propose to use the distributed representation for trajectory modeling.

Formally, we model each check-in location ℓ_j with an L-dimensional embedding vector $v_{\ell_j} \in \mathbb{R}^L$, consisting of L real values. The essence of distributed representation learning is to characterize objects as continuous variables in a vector space, which overcomes the representation sparsity

1. In our dataset, there is no explicit delimiters for trajectories. We follow previous studies [57] to split trajectories if the time interval of consecutive check-ins exceeds a threshold, say a day in our work.

in "one-hot" representation. We assume that the generation of a check-in location is associated with some contextual information. We denote the fth contextual feature by an L-dimensional embedding vector $\boldsymbol{v}_f \in \mathbb{R}^L$. A softmax multi-class classifier is employed to generate a check-in location conditioned on its contextual information

$$Pr(\ell_j|\boldsymbol{x}^{(\ell_j)}) = \frac{\exp(\bar{\boldsymbol{v}}_{\ell_j}^\top \cdot \boldsymbol{v}_{\ell_j})}{\sum_{\boldsymbol{v}'} \exp(\bar{\boldsymbol{v}}_{\ell_j}^\top \cdot \boldsymbol{v}')}, \tag{2}$$

where $\bar{\boldsymbol{v}}_{\ell_j}$ is the averaged vector representation of the embedding vectors corresponding to all relevant contexts (i.e., non-zero entries) in $\boldsymbol{x}^{(\ell_j)}$ for ℓ_j as follows

$$\bar{\boldsymbol{v}}_{\ell_j} = \frac{1}{\sum_f x_f^{(\ell_j)}} \sum_f x_f^{(\ell_j)} \times \boldsymbol{v}_f. \tag{3}$$

The proposed MC-TEM is general and flexible to model various kinds of contextual information for trajectory modeling. The associated contexts for each check-in location are characterized as embedding vectors in the same space. In this way, it will be relatively easy to measure the relatedness between different kinds of contextual features. As will be shown in Section 5.4, it is convenient to perform qualitative analysis on how each kind of contextual feature is useful with simple similarity measurements such as cosine similarity. Although we adopt a relatively simple average aggregation, a merit of MC-TEM is that the computational cost for Eq. (2) is not affected with varying of the context window length. We only need to pay slightly more costs to Eq. (3). Hence, it is useful to help control model complexity when using a long window or incorporating multiple kinds of contextual information.

3.2 Modeling the Contexts

With the above general model, we now study how to model the contextual features. It is easy to identify an interesting generation hierarchy in trajectory modeling: "`user` $\rightarrow$ `trajectory` $\rightarrow$ `location`". Hence, we adopt a three-level hierarchy to organize the contextual features in a top-down way, i.e., user level, trajectory level and location level. A fourth kind of features has also been considered, i.e., temporal contexts. Given a target location ℓ_j, we next construct the contextual feature vector $\boldsymbol{x}^{(\ell_j)}$. We assume that there are totally U users, M locations, T trajectories and C category labels.

3.2.1 User-Level Contexts

We first study user-level contexts, which would be used by all trajectories from the same user.

User preference. Intuitively, the personal preference is of priority to contribute to the location choices in trajectories, which reflects the overall user interests, habits and behavioural patterns. With the same intent, two users are likely to have different travel schedules. E.g., for the intent of "relax afterwork", a shopaholic is probable to go shopping while a music fan is probable to attend a music concert; while for the intent of "dinner", two guys might prefer "Burger King" and "McDonald's" respectively. We reserve U dimensions for user features, and the uth user dimension indicates that whether ℓ is generated by u or not.

3.2.2 Trajectory-Level Contexts

Contrast to user-level contexts shared by all trajectories from each user, trajectory-level contexts will be used by the locations in a specific trajectory.

Trajectory intent. The generation of the current location is influenced by the overall trajectory intent. For example, with different trajectory intents, the missing entry ℓ_j in an example trajectory sequence "`residential distinct` $\rightarrow$ `transportation` $\rightarrow \ell_j$ " can be assigned to different choices: ℓ_j is likely to be a *workplace* location for working intents while ℓ_j is likely to be a *movie* location for recreation intents. We reserve T dimensions for trajectory features, and the tth user dimension indicates that whether the current target location ℓ is in trajectory t.

3.2.3 Location-Level Contexts

We continue to study location-level contexts, which are the building blocks for trajectories. Since we would discuss the surrounding locations, we introduce *context window* to denote all the previous K and the successive K locations together with corresponding auxiliary information such as category labels. The length for context window is thus $2 \times K$.

Location relatedness. It is a common observation that the current location is influenced by its surrounding locations in a trajectory sequence. For example, a person taking a subway may check-in according to the subway stop schedule. Let $\ell_{j-K} : \ell_{j+K}$ denote the location sequence ℓ_{j-K}, ℓ_{j-K+1}, \ldots, ℓ_{j+K} (excluding ℓ_j). This idea resembles to that in the word embedding model (e.g., word2vec [31]): the semantics of a word is related to those of the surrounding words. We reserve M dimensions for location features, and the mth location dimension indicates that whether m is in the list $\ell_{j-K} : \ell_{j+K}$.

Category label. Existing studies [29] have shown that the categories of surrounding locations influence the generation of the current location. Here a "category" can be understood as a label for a collection of locations with some similar service function. For example, a subway stop can be labelled as "transportation" while a "Burger King" restaurant can be labelled as "restaurant". Given an example trajectory for "going home afterwork", it usually starts with *working office* and ends with *home* via *public transportation*. With this kind of knowledge, for a sequence "`office` $\rightarrow \ell_j \rightarrow$ `residen-tial distinct`", we can infer that the missing entry ℓ_j is likely to be a traffic-based location. We reserve C dimensions for category features, and the cth category dimension indicates that whether c is in the list $c_{j-K} : c_{j+K}$.

3.2.4 Temporal Contexts

We finally study the temporal contexts.

Daily routine. Typically, a user will have regular daily activities. For example, weekday morning will be the peak period for transportation due to the fact that a large amount of users need to go to working office; At noon, food places will have their reception peak hours for lunch service. Thus, daily routine summarizes the overall regular behavior patterns for users. We characterize it at the hour scale, i.e., a number h_j ranging from 1 to 24 is used to denote the hour index. We reserve 24 dimensions for hour features, and the hth hour dimension indicates that whether ℓ is visited at the hth hour. Note that it is difficult to estimate the stay interval

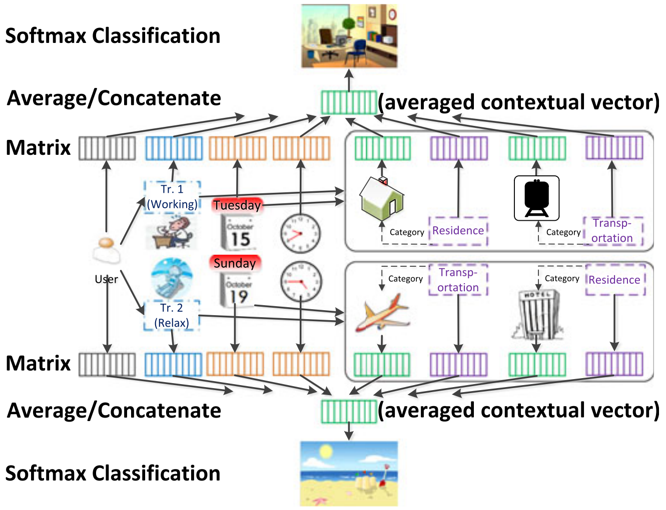


Fig. 3. An illustrative example for the proposed MC-TEM ($K = 1$). The user generates two trajectories with different intents (i.e., *working* versus *relax*), locations, categories, days, and hours. Each colored box denotes a kind of contextual information, denoted by an L-dimensional vector.

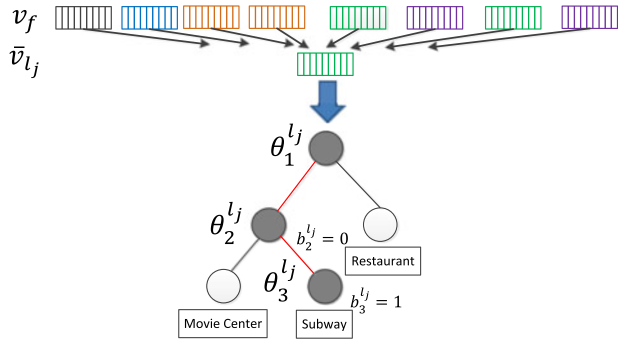


Fig. 4. An illustrative example for the binary tree in the hierarchical softmax ($\ell_j =$ "*Subway*" and $L(\ell_j) = 3$). Every node is represented in a circle associated with an embedding vector θ. v_f denotes the embedding vector of the fth context feature for location ℓ_j and these vectors are averaged to $\bar{v}_{\ell_j}$. The nodes on the path from the root to l_j are marked in gray and linked with red lines.

at a specific location, and we only use the hour index extracted from the check-in timestamp.

Weekly periodicity. Intuitively, a user will show some periodic activities by weeks. For example, Monday to Friday are typically reserved for *working*, while Saturday and Sunday are reserved for *rest*. Even for the five weekdays, a user is likely to have different working schedules, e.g., a weekly report meeting might be set on Mondays. In our dataset, a trajectory takes place within a single day, and it is associated with a unique day. We map it to the numbers from 1 to 7 by the order in a week. We reserve seven dimensions for week features. The target location ℓ is associated with a trajectory t, and the dth week dimension indicates that whether the trajectory t is generated on the dth day in a week.

3.3 The Final Model

By instantiating the context feature vector $\boldsymbol{x}^{(\ell_j)}$ in Eq. (1), we define the following objective function on the entire data collection

$$\sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{T}^{(u)}}\frac{1}{N_t}\sum_{j=1}^{N_t}\log\Pr\Big(\ell_j \,|\, \underbrace{u}_{\text{user-level context}}, \underbrace{t}_{\text{trajectory-level context}}, \underbrace{\ell_{j-K}:\ell_{j+K}, c_{j-K}:c_{j+K}}_{\text{location-level context}}, \underbrace{d,h}_{\text{temporal context}}\Big), \tag{4}$$

where $\mathcal{U}$ is the set of users, $\mathcal{T}^u$ is the set of trajectories generated by user u and N_t is the length of trajectory t. We model each kind of contextual feature with an L-dimensional embedding vector. Following Eq. (3), given a target location ℓ_j, we derive its contextual embedding vector as follows

$$\bar{\boldsymbol{v}}_{\ell_j} = \frac{1}{4K+4}\Bigg\{\sum_{-K\le k\le K, k\ne 0}\big(\boldsymbol{v}_{\ell_{j+k}} + \boldsymbol{v}_{c_{j+k}}\big) + \big(\boldsymbol{v}_u + \boldsymbol{v}_t + \boldsymbol{v}_d + \boldsymbol{v}_h\big)\Bigg\}, \tag{5}$$

where $\boldsymbol{v}_{\ell_{j+k}}$, $\boldsymbol{v}_{c_{j+k}}$, $\boldsymbol{v}_u$, $\boldsymbol{v}_t$, $\boldsymbol{v}_d$ and $\boldsymbol{v}_h$ correspond to the contexual embedding vectors for surrounding locations, surrounding category labels, user preference, trajectory intent, day and hour respectively. With $\bar{\boldsymbol{v}}_{\ell_j}$, we can plug it into Eq. (2) for computing the generative probability of ℓ_j. Finally we can derive the objective function in Eq. (4). In order to understand how these kinds of contexts are used in the generation process of trajectory data, we present an illustrative example for MC-TEM in Fig. 3. A user has generated two trajectory sequences corresponding to the trajectory intents of *work* and *relax* respectively. In the first trajectory, she goes from home to the working office via subway on weekday morning; while in the second trajectory, she takes the flight to the hotel for vocation. Each contextual feature is associated with an embedding vector in the same space for embedding target locations. Further, the target locations are generated with the aggregated contextual embedding vector using Eq. (2).

3.4 Parameter Learning

In MC-TEM, the parameters to learn are the embedding vectors for various context features $\{\boldsymbol{v}_f\}$ and the output embedding vectors for target locations $\{\boldsymbol{v}_\ell\}$. By following [31], we do not discriminate between the emitting vector and output vector for a location. For convenience, let $\boldsymbol{v}_f$ denote the embedding vector for any contextual feature.

For parameter learning, MC-TEM needs to maximize the log probability defined in Eq. (4). However, directly optimizing this objective function is impractical because the cost of computing the full softmax for the multi-classifier to predict the current location is extremely high. To solve this difficulty, the recently proposed *hierarchical softmax* [34] and *negative sampling* [19], [33] techniques have been shown to be both efficient and effective to approximate the full softmax. In what follows, we derive how to update the parameters in our model with hierarchical softmax, and the parameter update method is similar for negative sampling.

As shown in Fig. 4, the hierarchial softmax uses a binary tree representation for every location as its leaves, and each node is explicitly associated with an embedding vector $\boldsymbol{\theta}$ for computing the relative probability to take the branch. Each leaf can be reached by an appropriate path from the root of the tree. In this way, instead of evaluating all the $|\mathcal{L}|$ nodes to obtain the probability distribution, it is needed to evaluate only about $\log_2(|\mathcal{L}|)$ nodes. More precisely, given the aggregate vector $\bar{\boldsymbol{v}}_{\ell_j}$ for target location ℓ_j, let $L(\ell_j)$ be the length of its corresponding path, and let $b_n^{\ell_j} = 0$ when the path of ℓ_j takes the left branch at the nth level and $b_n^{\ell_j} = 1$ otherwise. Then, the hierarchical softmax defines $Pr(\ell_j|\boldsymbol{x}^{(\ell_j)})$ as follows:

$$Pr(\ell_j|\boldsymbol{x}^{(\ell_j)}) = \prod_{n=2}^{L(\ell_j)} ([\sigma(\bar{\boldsymbol{v}}_{\ell_j}^\top \boldsymbol{\theta}_{n-1}^l)]^{1-b_n^{\ell_j}} \cdot [1-\sigma(\bar{\boldsymbol{v}}_{\ell_j}^\top \boldsymbol{\theta}_{n-1}^l)]^{b_n^{\ell_j}}), \quad (6)$$

where $\sigma(z) = \frac{1}{1+\exp(-z)}$. All parameters are trained by using the Stochastic Gradient Descent method. During the training, the algorithm iterates over the locations of all trajectories, and at each time, a target location ℓ_j with its context window is used for update. After computing the hierarchical softmax according to Eq. (6), the error gradient is obtained via backpropagation and we use the gradient to update the parameters in our model. To derive how $\boldsymbol{\theta}$ is updated at each step, the gradient for $\boldsymbol{\theta}_{n-1}^{\ell_j}$ is computed as follows:

$$\frac{\partial \mathcal{L}(\ell_j, n)}{\partial \boldsymbol{\theta}_{n-1}^{\ell_j}} = [1 - b_j^{\ell_j} - \sigma(\bar{\boldsymbol{v}}_{\ell_j}^\top \boldsymbol{\theta}_{n-1}^{\ell_j})]\bar{\boldsymbol{v}}_{\ell_j}.$$

In this way, $\boldsymbol{\theta}_{n-1}^{\ell_j}$ can be updated as follows:

$$\boldsymbol{\theta}_{n-1}^{\ell_j} \leftarrow \boldsymbol{\theta}_{n-1}^{\ell_j} + \eta[1 - b_j^{\ell_j} - \sigma(\bar{\boldsymbol{v}}_{\ell_j}^\top \boldsymbol{\theta}_{n-1}^{\ell_j})]\bar{\boldsymbol{v}}_{\ell_j},$$

where η denotes the learning rate. To derive how the context embedding vectors are updated, the gradient for $\bar{\boldsymbol{v}}_{\ell_j}$ is computed as follows:

$$\frac{\partial \mathcal{L}(\ell_j, n)}{\partial \bar{\boldsymbol{v}}_{\ell_j}} = [1 - b_j^{\ell_j} - \sigma(\bar{\boldsymbol{v}}_{\ell_j}^\top \boldsymbol{\theta}_{n-1}^{\ell_j})]\boldsymbol{\theta}_{n-1}^{\ell_j}.$$

With this derivative, an embedding vector $\boldsymbol{v}_f$ in the context of location ℓ_j can be updated as follows:

$$\boldsymbol{v}_f \leftarrow \boldsymbol{v}_f + \eta \sum_{h=2}^{L(\ell_j)} \frac{\partial \mathcal{L}(\ell_j, h)}{\partial \bar{\boldsymbol{v}}_{\ell_j}}.$$

The learning algorithm. We summarize the learning algorithm using hierarchical softmax for the proposed MC-TEM in Algorithm 1. The algorithm iterates through all the trajectories and updates the embedding vectors until the procedure converges. In each iteration, given a current location in a trajectory, the algorithm first obtains its embedding vectors and computes its context embedding vector. Based on the derivation above, the binary tree in hierarchical sampling is updated followed by the embedding vectors. Given the vector size of VS, the time complexity for an iteration is $\mathcal{O}(N \cdot VS \cdot CL \cdot \log|\mathcal{L}|)$, where N is the total number of check-ins considered in an iteration and the log term comes from the binary tree to determine the target location from the $|\mathcal{L}|$ leave nodes in the binary tree. As will be shown in Fig. 8, CL is usually smaller than 10, and VS is around 300.

Algorithm 1. The Hierarchical Softmax Algorithm for Learning the Parameters of MC-TEM

Input: $\mathcal{T}, K, VS, N$
Output: $\{\boldsymbol{v}_f\}$
Initialize the parameters randomly;
Shuffle the dataset;
repeat
  for $i = 1$ to N **do**
    Sample a check-in record $\langle u, \ell, c, d, h\rangle$ from $\mathcal{T}$;
    Set $e \leftarrow 0$;
    Compute $\bar{\boldsymbol{v}}_\ell$ by using Eq. (3);
    for $n = 2$ to $L(\ell)$ **do**
      $q \leftarrow \sigma(\bar{\boldsymbol{v}}_\ell^\top \cdot \theta_{n-1}^\ell)$;
      $g \leftarrow \eta \cdot (b_j^\ell - 1 - q)$;
      $e \leftarrow e + g \cdot \theta_{n-1}^\ell$;
      Update $\theta_{n-1}^\ell = \theta_{n-1}^\ell + g \cdot \bar{\boldsymbol{v}}_\ell$;
    end
    for *each embedding vector* $\boldsymbol{v}_f$ **do**
      Update $\boldsymbol{v}_f \leftarrow \boldsymbol{v}_f + e$;
    end
  end
until convergence

4 APPLICATIONS

In the above, we have presented the general formulation of MC-TEM. Now, we continue to study how it applies to real applications tasks. We select two important tasks in location-based services, namely *location recommendation* and *social link prediction*.

4.1 Location Recommendation

Location recommendation aims to recommend locations for users to visit in a personalized way, which delivers business venues to persons of potential interests [4]. It is particularly useful when a user travels in a new city or region.

General location recommendation. Formally, given a user u together with a candidate list of locations $\mathcal{L}$, a *location recommender* system will generate a ranked list of the candidate locations by using a recommendation score function $S(u, \ell)$ indicating the preference level at which ℓ is recommended to u. We assume that a location point ℓ is associated with a small set of category labels $\mathcal{C}^{(\ell)}$. We can define the objective function for this task as follows

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

where u, t, r, $\ell_{j-K} : \ell_{j+K}$ and $c_{j-K} : c_{j+K}$ correspond to the user context, trajectory context, region (city) context, location contexts and category contexts respectively. Then we can train the model to derive the embedding vectors $\boldsymbol{v}_u$, $\boldsymbol{v}_r$, $\boldsymbol{v}_\ell$ and $\boldsymbol{v}_c$ respectively for users, cities (region), locations and category labels. Our MC-TEM method defines the ranking score function as follows

$$S(u, \ell) \propto (\boldsymbol{v}_r + \boldsymbol{v}_u)^\top \cdot \left(\frac{\sum_{c\in\mathcal{C}^{(\ell)}} \boldsymbol{v}_c}{|\mathcal{C}^{(\ell)}|} + \boldsymbol{v}_\ell\right), \quad (7)$$

where the item $\frac{\sum_{c\in\mathcal{C}^{(\ell)}} \boldsymbol{v}_c}{|\mathcal{C}^{(\ell)}|}$ characterizes the category contexts. Our ranking criteria follows the idea presented in [49], where city (region) preference is shown to be effective in location recommendation.

Time-aware location recommendation. With the incorporation of temporal factors, given a user u together with a candidate list of locations $\mathcal{L}$, a *location recommender* system will generate a ranked list of the candidate locations in a specific time interval (accurate to the hour scale of a day in this paper). We can extract the day and hour information from the given time interval and characterize the contextual information used for this task as follows

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

With this model, we can define the score function $S(u,\ell)$ as follows

$$S(u,\ell)\propto(\boldsymbol{v}_r+\boldsymbol{v}_{d_t}+\boldsymbol{v}_{h_j}+\boldsymbol{v}_u)^\top\cdot\left(\frac{\sum_{c\in\mathcal{C}^{(\ell)}}\boldsymbol{v}_c}{|\mathcal{C}^{(\ell)}|}+\boldsymbol{v}_\ell\right), \quad (8)$$

where d_t and h_j is the day index in a week and the hour index in a day respectively.

4.2 Social Link Prediction

Besides the direct applications to improve geo-oriented tasks, recently human trajectory similarity has been shown to be of important evidence for predicting social links[25]. Especially, the geo-social networks provide us with both trajectory data and link information. Several pioneering studies have tried to validate "homophily principle" of "like to associate with alike" [22], [30] based on human mobility.

Link prediction. Given the trajectory data from users u and v, link prediction aims to predict whether there is a (reciprocal) link between u and v. A typical solution [39] to this task is described as follows. We first extract useful features from a pair of users (u,v) and then construct a feature vector $\mathbf{x}^{(u,v)}\in\mathbb{R}^N$. Furthermore, we train a learning function $f(\mathbf{x}^{(u,v)})$ for predicting the linking label $y^{(u,v)}$, which is a binary value in the set *{LINK, NON-LINK}*, given the input $\mathbf{x}^{(u,v)}$.

The key point of this approach becomes how to derive effective feature representation for link prediction. Next we show how to generate an effective feature representation by using MC-TEM. We first extract features from a single user and then construct the feature vector for the user pair. We represent each user u with a user vector $\mathbf{x}^{(u)}$. Given a pair of users u and v, we can obtain the feature vector $\mathbf{x}^{(u,v)}=\mathbf{x}^{(u)}\circ\mathbf{x}^{(v)}$, where "$\circ$" is the Hadamard product and the ith dimension $x_i^{(u,v)}$ is set to the product between $x_i^{(u)}$ and $x_i^{(v)}$. We build the user feature vector $\mathbf{x}^{(u)}$ by using the contexts based on user preference and trajectory intent:

User based features. For each user u, the user embedding vector $\boldsymbol{v}_u$ represents the user preference. As such, we set $x_i^{(u)}=v_{u,i}$, where $v_{u,i}$ denotes the ith entry in $\boldsymbol{v}_u$.

Trajectory based features. Since different users have a varying number of trajectories, we adopt the *max pooling* technique to derive a single vector with fixed length. Hence, we set $x_i^{(u)}=\max_{t\in\mathcal{T}^u}\{v_{t,i}^{(u)}\}$, where $v_{t,i}^{(u)}$ denotes the ith entry of the embedding vector $\boldsymbol{v}_t^{(u)}$ of trajectory t from user u.

The first kind of features can be directly obtained from MC-TEM while the second kind of features is obtained by using the max-pooling technique. *Max-pooling* is widely used in previous studies [7], [11] to combine multiple embedding vectors into a single fixed-length feature vector. We concatenate these two kinds of features as a user vector, and then perform the Hadamard product between user vectors to derive the final feature vector for classification. We also tried other aggregation methods, e.g., average pooling, but no one is better than that obtained with max-pooling.

TABLE 2
Statistics of Our Datasets

Dataset	# Users	# Check-ins	# Links	# Locations
Foursquare$_S$	4,163	483,814	32,512	121,142
Foursquare$_L$	266,909	33,278,683	—	3,680,126
Gowalla	216,734	12,846,151	736,778	1,421,262

5 Experiments

In this section, we conduct the evaluation experiments on the two application tasks introduced in Section 4.

5.1 Datasets

Three public geo-social networking datasets are used in this work, namely *Foursquare$_S$* [49], *Foursquare$_L$* [47] and *Gowalla* [35]. All datasets contain check-in records in the form of `(User ID, Location ID, Location Category, Timestamp, City)`. Only *Foursquare$_S$* and *Gowalla* contain the social connection links among users. Foursquare$_L$ and Gowalla are used for location recommendation, while Foursquare$_S$ and Gowalla are used for link prediction.

The detailed statistics of the three datasets are summarized in Table 2.

5.2 Evaluation on Location Recommendation

We first evaluate the effectiveness on location recommendation, including both *general location recommendation* and *time-aware location recommendation*. Based on the previous studies [16], [49], we further consider two different recommendation settings for both recommendation tasks: 1) home-city recommendation; 2) new-city recommendation. The former is a typical task for a traditional location-based recommender system; while the latter is recently proposed to help users organize travel routines outside their own home cities [49]. Next we will use the same following evaluation setup for both general and time-aware location recommendation.

5.2.1 Construction of Test Collection

Given a user, the corresponding home city is identified as the city with the most number of occurrences in her check-in records [41], [49]. To construct the training and test sets, we split the data by trajectories. The major reason is that splitting by check-in records is likely to yield data correlation between training and test sets, i.e., part of a trajectory is in training set and the rest of the trajectory is in test set. For home-city recommendation, we first select 20 percent trajectories with only home-city locations into the test set, while all the rest trajectories are considered as the training data; For new-city recommendation, we follow the setting in [49], where all the trajectories related to a non-home city are selected into the test set and the rest trajectories are considered as the training data.

5.2.2 Evaluation Setup

For each check-in record in test set, we use the following way in [49], [52] for evaluation. First, we randomly select (at most) $1{,}000$ locations not visited by the current user. Together with the target location (i.e., the one was visited), we can obtain a candidate list of $1{,}001$ locations for recommendation; Then, a recommender system will rank the locations in the candidate list. We form a ranked list by ordering all the $1{,}001$ locations according to their ranking scores. Let n denote the rank of the target location within this list, and the optimal result corresponds to the case where it precedes all the additional locations, i.e., $n = 1$; Finally, we form a top-k recommendation list by picking up the top k ranked locations from the list. We define hit@k for a single test case as either the value 1 if the target location is identified in the top k results, or else the value 0. The overall Recall@k is the ratio of hits in all the test check-in records:

$$Recall@k = \frac{\#hit@k}{\#all\_cases}. \tag{9}$$

5.2.3 Methods to Compare

For *general location recommendation*, we consider the following methods for comparison.

- *User, social and geography based model (USG)* [48]: It fuses the user based collaborative filtering, friendship based collaborative filtering and distance based likelihood to compute the ranking probability for all candidate locations.
- *W*^{4} [6]: W^4 is a tensor based multi-dimensional collaborative recommendation system for Who (User), What (Location Category), When (Time) and Where (Location). For fairness, we modify the original W^4 by using a new *City* dimension and removing the *When* dimension here.
- *Location and content aware LDA (LCA-LDA)* [49]: LCA-LDA is a location-content-aware recommender model which is developed to support location recommendation for users traveling in both new cities and home cities. This model takes into account both personal interests and local preferences of each city by exploiting both location co-visiting patterns and category labels.
- *Paragraph vector (PV)* [23]: PV combines all trajectories of the same user into a single pseudo "paragraph" and then runs the PV algorithm. For each user u, we can learn his/her paragraph embedding vector $\boldsymbol{x}^{(u)}$ as the user representation and the location embedding vectors as the location representation. Then cosine similarity is used to rank locations given a user. We implement it using the toolkit *Gensim*.[2]
- *MC-TEM*: It is our proposed method, which uses the ranking function $S(u, \ell)$ in Eq. (7) for recommendation.

For *time-aware location recommendation*, we consider three additional methods with temporal factors for comparison.

- W^5 [6]: W^5 further recovers the *Time* dimension for time-aware location recommendation based on the W^4 model for general location recommendation.
- *User based CF with time function (UT)* [13]: UT is a user-based CF algorithm that estimates the similarity between users as what traditional user-based CF does, but weights the check-ins of a similar user according to the gaps between their time slots and the recommendation time slot.
- *Time and geography based approach (TGA)* [52]: TGA is a unified framework based on user-based collaborative filtering which incorporates both temporal and spatial influence for location recommendation.
- *MC-TEM*$_{++}$: It is our proposed method, which uses the time-aware ranking function $S(u, \ell)$ in Eq. (8) for recommendation. We use the sub-suffix "++" to distinguish it from the variant using Eq. (7).

We take 10 percent training data as the development set to tune the parameters in various methods. Following [48], the weights in USG for user, social and geography are set to 0.2, 0.1 and 0.7 respectively; Following [49], the topic number of LCA-LDA is set to 100; Following W^4 [6], the optimal number of latent factors for tensor decomposition is set to 100; Following TGA [52], the decaying factor and time slot length is set to 1 and 1 hour respectively, and the weights for temporal and geographical influence are set to 0.5; Following [13], UT adopts the same time slot length of 1 hour. For PV, MC-TEM and MC-TEM$_{++}$, we adopt the hierarchial softmax and the vector size and context window length are set to 300 and 8 respectively.

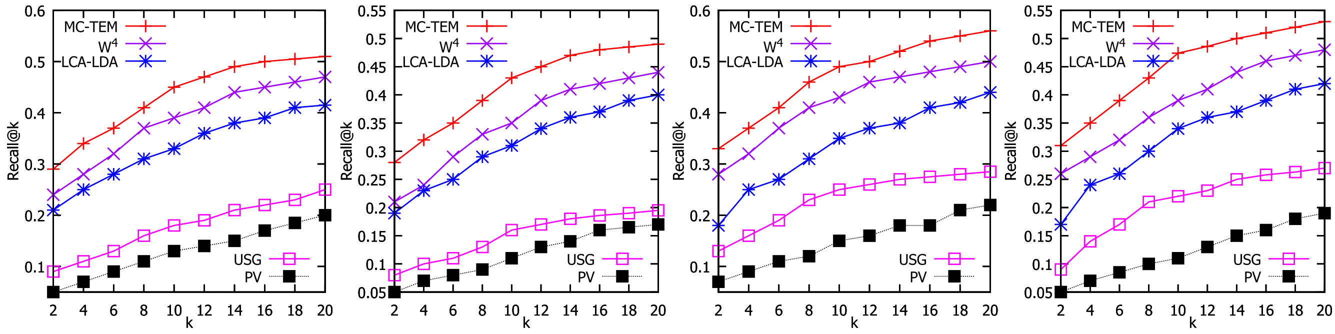
5.2.4 Performance Comparison

We present the results for *general* and *time-aware* location recommendation in Figs. 5 and 6 respectively, each of which reports the performance in the two settings, i.e., new-city and home-city settings, on the Foursquare$_L$ and Gowalla datasets. We show the results of Recall@k by varying k in the range $[2, \ldots, 20]$ with a step of 2.

General location recommendation. In Fig. 5, we can have the following observations: 1) MC-TEM, W^4 and LCA-LDA take up the top three ranks and perform relatively stable in both settings, while MC-TEM further yields about $10\% \sim 57\%$ improvement over W^4 and LCA-LDA; 2) The performance of USG and PV have shown a substantial performance drop in new-city recommendation compared to home-city recommendation. This is because only MC-TEM, W^4 and LCA-LDA can deal with the new-city recommendation, in which the information of local city preference and category labels is incorporated.

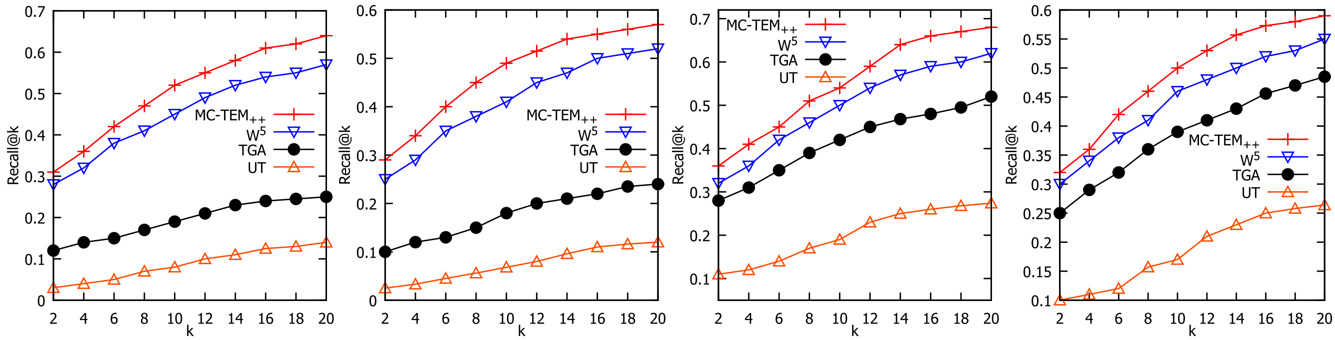
Since LCA-LDA, W^4 and MC-TEM use similar contextual information, it is worthwhile to analyze why MC-TEM performs better than W^4 and LCA-LDA. An important factor has been ignored in both LCA-LDA and W^4, i.e., surrounding contexts. In our formulation, the previous K and successive K locations with their labels are used as contextual

2. https://radimrehurek.com/gensim/index.html



(a) *Foursquare$_L$* (b) *Gowalla*
I. New-city setting.
(c) *Foursquare$_L$* (d) *Gowalla*
II. Home-city setting.

Fig. 5. Performance comparison on general location recommendation.



(a) *Foursquare$_L$* (b) *Gowalla*
I. New-city setting.
(c) *Foursquare$_L$* (d) *Gowalla*
II. Home-city setting.

Fig. 6. Performance comparison on time-aware location recommendation.

information, hence surrounding contexts are well characterized.[3] W^4 cannot directly model surrounding contexts in a long window. As a simplification, it incorporates pairwise location-to-location correlation as regularized factors, and further performs better than LCA-LDA.In addition, the recommendation task (Eq. (7) and (8)) can be essentially casted as a relatedness measurement problem. LCA-LDA adopts a probabilistic way, and characterizes the relatedness via probabilistic generative process. As a comparison, W^4 adopts a tensor decomposition way and MC-TEM adopts a distributed representation way. Both W^4 and MC-TEM represent each kind of contextual information in the same latent space, and such a way is more direct to capture the relatedness among multiple kinds of contexts.

Finally, we compare MC-TEM with PV. Although PV also adopts a similar distributed representation way, it performs much worse than MC-TEM, which indicates the simple application of distributed representation methods may not work well for trajectory data.

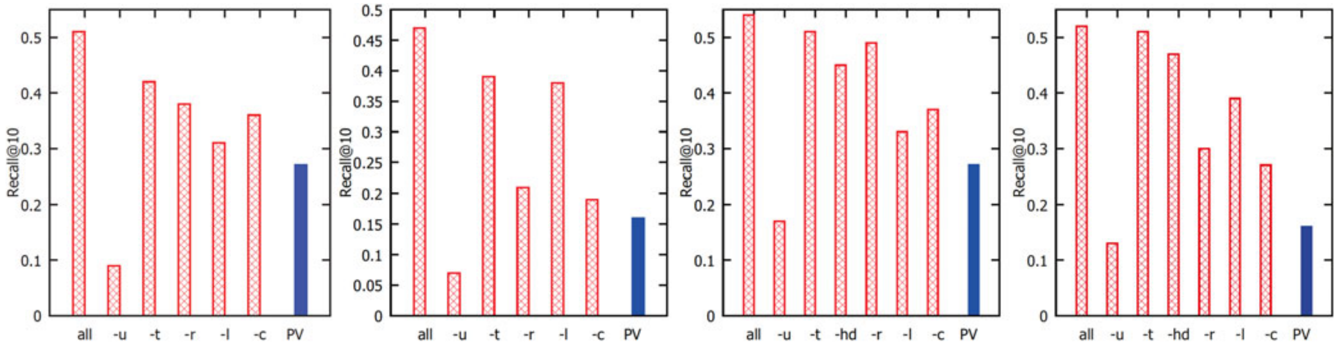
Time-aware location recommendation. Note that we do not incorporate USG, PV and LCA-LDA as baselines here, for they do not consider temporal information. As introduced in the setup (Section 5.2.3), we use the variant of MC-TEM with the incorporation of temporal contexts, denoted as MC-TEM$_{++}$. In Fig. 6, we can have the following observations: 1) MC-TEM still performs best among the three comparison methods, especially in the new-city setting; 2) The performance of baselines TGA and UT has shown a substantial drop in new-city recommendation compared to home-city recommendation. This is because TGA and UT cannot deal with the new-city setting well, when a user travels in a new city or region, both methods will suffer from the data sparsity problem; 3) W^5 is a very competitive method and it performs the best among all the baselines.

Summary. In these two sub-tasks for location recommendation, our proposed methods MC-TEM and MC-TEM$_{++}$ perform consistently better than several strong baselines recently proposed. More importantly, our model MC-TEM is flexible to incorporate various context information in a general way, and it further projects the contexts into the same embedding space.

5.2.5 Impact of Different Contextual Factors

Fig. 7 presents the effect of different contexts respectively for general and time-aware location recommendation on the Foursquare$_L$ dataset. We incorporate the method *PV* as a baseline. It can be observed as follows: 1) All the considered contexts are helpful to the two recommendation sub-tasks; 2) User context is the most important factor to consider. This is because the current task is essentially a personalized recommendation problem where user preference plays the key role in system performance; 3) Compared to home-city setting, city and category contexts have more significant effects in the new-city setting. It is mainly because in the new-city setting, user preference will not be well learnt due to data sparsity, while local city preference and category preference become more important. 4) Temporal context

3. We will specially examine the effect of the context window length in Figs. 8b and 8d.



(a) *Home-city setting.* (b) *New-city setting.*
I. General location recommendation.
(c) *Home-city setting.* (d) *New-city setting.*
II. Time-aware location recommendation.

Fig. 7. Impact of contextual factors for location recommendation on the Foursquare$_L$ dataset (u, t, d, h, r, l and c denote user context, trajectory context, day context, hour context, city context, location context, and category context, respectively. "-" indicates that the corresponding kind of context is excluded).

(see the performance with the legend label "-hd" in Figs. 7c and 7d) is important to consider in time-aware location recommendation. In fact, when day context and hour context are removed, MC-TEM$_{++}$ become MC-TEM with the score function in Eq. (7).

The only difference between the two evaluation sets for general and time-aware location recommendation lies in that we build the evaluation set for time-aware location recommendation by providing the timestamps for each location, too. Combining the results in Figs. 5 and 6, we can observe that MC-TEM$_{++}$ further improves over MC-TEM: $0.36 \sim 0.68$ versus $0.33 \sim 0.56$ in home-city setting and $0.31 \sim 0.64$ versus $0.29 \sim 0.51$ in new-city setting on the Foursquare$_L$ dataset.

The observations on the Gowalla dataset are similar to what we have obtained on the Foursquare$_L$ dataset, and we omit it here.

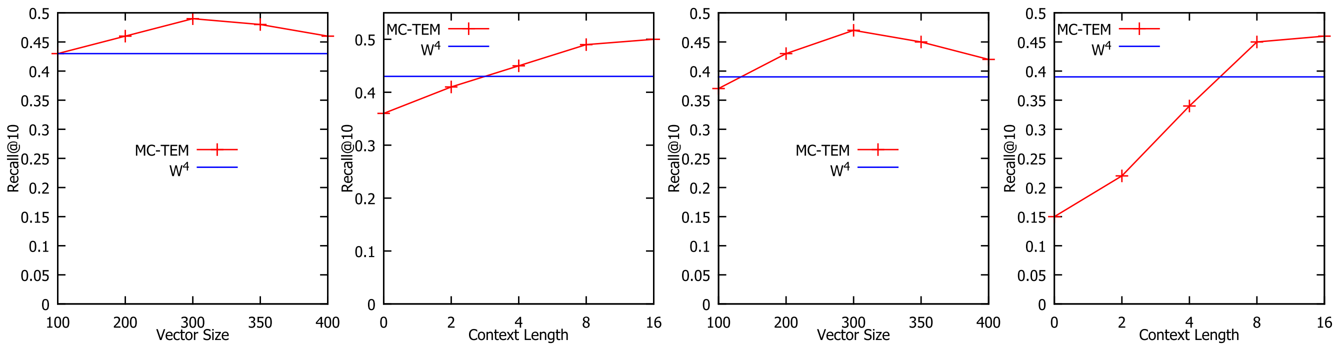
5.2.6 *Parameter Tuning*

We have two important parameters to set in MC-TEM, i.e., the vector size (VS) and context window length (CL). Previously, we have empirically found the following setting leads to a good performance for MC-TEM by using a development set: $VS = 300$ and $CL = 8$. In addition, as we have mentioned in Section 3.4, there are two alternative learning algorithms for MC-TEM, namely *hierarchical softmax* and *negative sampling*. In our experiments, we have found that hierarchical softmax leads to a better performance than *negative sampling*, and the performance of negative sampling slightly improves with the increasing of negative samples.

Hence, we next adopt the hierarchical softmax method to tune the two important parameters in MC-TEM, namely VS and CL. We select the W^4 (tuned to its optimal performance) as comparisons. By reusing the previous settings for VS and CL, we tune each with the other fixed. In Fig. 8, we can see that (1) MC-TEM is relatively stable when the vector size VS falls in the range $[200, 350]$ and achieves the optimal when $VS = 300$, and (2) the performance improvement becomes less significant when CL is larger than 8, which indicates that $CL = 8$ is sufficient to give good performance. Especially, our model is better than W^4 when $CL \geq 8$. This further explains why MC-TEM is better than the other baselines: it can model the surrounding contexts consisting of both locations and category labels in a relatively long window. Although W^4 can incorporate location-to-location correlation, it is modeled as pairwise factors and cannot directly characterize surrounding contexts. For MC-TEM, the performance when $CL \leq 2$ is significantly worse than that when $CL \geq 4$. These observations indicate the importance of surrounding contexts in trajectory modeling.

5.3 Evaluation on Social Link Prediction

In this section, we show how our MC-TEM model performs on the task of social link prediction.



(a) Varing VS ($CL = 8$). (b) Varying CL ($VS = 300$).
I. Home-city setting.
(c) Varing VS ($CL = 8$). (d) Varying CL ($VS = 300$).
II. New-city setting.

Fig. 8. Parameter tuning for general location recommendation on the Foursquare$_L$ dataset, where $k = 10$.

5.3.1 Construction of the Test Collection

The task aims to predict whether a social link between a pair of users exists or not *only based on their trajectory data*. The *Foursquare*$_S$ and *Gowalla* datasets in Table 2 contain both the trajectory data and social connection links among users. We assume that all the trajectory information for a user is available for unsupervised feature extraction. We split the positive instances (i.e., the social connection links) into training set and test set. To generate negative instances (i.e., user pairs without links) for both training and test sets, we randomly select non-linked user pairs with a ratio of 1:1 compared to positive instances. To prevent data overlap, the users in training set will not appear in test set. Furthermore, five-fold cross validation test is used for the evaluation of the final performance.

5.3.2 Evaluation Metrics

To measure the effectiveness on social link prediction, we follow the previous work [39] and adopt the commonly used precision, recall and F1-measure as the metrics. Let $\mathcal{P}_T$ be the set of all gold user pairs with real friend links, $\mathcal{P}_R$ be the number of all user pairs identified by a candidate method as friends. The precision, recall and F1 are defined as

$$P = \frac{|\mathcal{P}_T \cap \mathcal{P}_R|}{|\mathcal{P}_R|}, R = \frac{|\mathcal{P}_T \cap \mathcal{P}_R|}{|\mathcal{P}_T|}, F1 = \frac{2PR}{P + R}.$$

5.3.3 Methods to Compare

In this task, our major aim is to test the capability of the proposed method for learning useful feature representation. We adopt the commonly used binary classifier *Support Vector Machine* (SVM) for the comparison methods.[4] A prediction method first generates a feature vector for a candidate user pair, which will be subsequently used as the input of the SVM classifier. Thus, a better method will yield a more effective feature representation leading to a higher classification performance.

We consider the following methods for comparison.

- *Entropy based model (EBM)* [39]: It calculates the diversity of co-occurrences for each user pair and the weighted location entropy for each co-visited location. It represents the state-of-the-art of social link prediction by using trajectory data.
- *MH* [43]: It mainly characterizes the mobile homophily, including distance between users' most likely visited locations, Spatial Co-Location Rate, Weighted Spatial Cosine Similarity, Weighted Co-Location Rate and Extra-role Co-Location Rate, to predict the social connection.
- *Trajectory similarity based approach (TSA)* [25]: TSA first represents each user as a trajectory in a hierarchical fashion, and then uses the similarity between the trajectories of two users to predict their social link.
- *HMM based approach (HMM)* [5]: HMM can be used to discover the underlying patterns in trajectory data in terms of transition probabilities between hidden states. A user u can be represented as a vector $\mathbf{x}^{(u)}$ of $C \times C$ entries, where C is the total number of states and x_i^u ($i = c_1 \times C + c_2$) denotes the number of state transitions from c_1 to c_2 for user u.
- *Paragraph vector (PV)* [23]: Similar to the PV approach for location recommendation, in this task, we combine all trajectories of the same user into a single pseudo "paragraph" and then runs the PV algorithm. For each user u, we learn his/her paragraph vector $\mathbf{x}^{(u)}$ as the feature vector.
- *MC-TEM*: It is our proposed method which is presented in Section 3.3 as Eq. (4) defines.

TABLE 3
Performance Comparison on Link Prediction

Methods	*Foursquare*$_S$			*Gowalla*		
	P	R	F1	P	R	F1
MH	0.71	0.64	0.67	0.69	0.62	0.65
EBM	0.78	0.50	0.61	0.77	0.49	0.60
TSA	0.61	0.55	0.58	0.6	0.53	0.56
HMM	0.22	0.73	0.34	0.21	0.71	0.33
PV	0.56	0.31	0.40	0.54	0.29	0.37
MC-TEM	**0.81**	**0.75**	**0.77**	**0.79**	**0.74**	**0.76**

We take 10 percent training data as development data to set the parameters in various methods. As suggested in [39], in EBM, the order of diversity parameter q is set to 0.1. Following [25], in TSA, check-ins within 300 meters and half an hour are considered to be the same stay point. Resulting in a good performance, the number of states for HMM is set to 20. Similar to the setting in location recommendation, both PV and MC-TEM are trained with the hierarchical softmax algorithm and the embedding vector size and context length are set to 300 and 8 respectively.

5.3.4 Performance Comparison

Table 3 presents the prediction performance of different methods on the Foursquare$_S$ and Gowalla datasets.

In Table 3, we can have the following observations: 1) EBM yields a higher precision score but a lower recall score while HMM has a higher recall but a lower precision. The EBM only considers the similarity based on location information and ignores the latent similarity between users with few common locations. In contrast, HMM gives high recall due to the fact that it characterizes the similarities based on high-level units (i.e., states) but has low precision since the learnt states are too general and not discriminative; 2) EBM achieves 27 and 39 percent improvement on precision than TSA and PV respectively. Although TSA also considers the co-occurrence, compared to EBM, it does not distinguish popular or rarely visited places; 3) Using several high-quality mobile homophily features, MH achieves decent precision and recall performance. Consistent with the findings in [43], human mobility could serve as a good predictor for the formation of new links; 4) MC-TEM performs best in terms of both precision and recall. Compared to HMM which clusters users based on states, MC-TEM derives the user similarity using multi-grained contextual information (i.e., user-level and trajectory-level) and thus obtains a higher recall score. Compared to MH and EBM, MC-TEM can further capture the order of check-ins together with their category labels, and thus obtains a higher precision score.

4. https://www.csie.ntu.edu.tw/~cjlin/libsvm/#download

TABLE 4
Effect of Contexts for Social Link Prediction on the Foursquare$_S$ Dataset

Metric	All	-u	-t	-l	-c	PV
Precision	0.81	0.35	0.18	0.25	0.72	0.56
Recall	0.75	0.51	0.41	0.66	0.29	0.21

5.3.5 Impact of Different Contextual Factors

Table 4 shows the comparisons for precision and recall scores of MC-TEM with different context information on the Foursquare$_S$ dataset. As a comparison, PV is incorporated with only the context information from users and surrounding locations. We can have the following observations: 1) Similar to the findings in Fig. 7, all the considered contexts contribute to the current task, while user context is not the most important factor to consider, in current task the aggregated trajectory context of a user yields the most improvement to the performance; 2) Interestingly, user, trajectory and location contexts have more significant effect on precision than recall, while category context has more significant effect on recall than precision. It conforms to the intuition that category labels capture a more general similarity evidence which tends to improve the recall score.

5.3.6 Parameter Tuning

For social link prediction, we have found that using the hierarchical softmax with the settings of $VS = 300$ and $CL = 8$ gives good performance. We examine the effect of kernel selection and kernel parameters on the performance. To tune the parameters for SVM, we take 10 percent training data as the developing data. We consider two commonly used kernels, namely linear kernel and Radius Basis Function (RBF) kernel. Both kernels need to specify a *cost* parameter, and the RBF kernel involves a second parameter γ. The default value for *cost* is set to 1 and default γ parameter is set to the reciprocal of the feature vector length, which is $\frac{1}{2VS} \approx 0.0016$ in our case. First, we have found that the performance improves with the increase of *cost*, and it becomes relatively stable when *cost* is set to 1,000 or larger. Thus, we set *cost* to 1,000. Then, we tune the parameter γ with a grid search method: when the γ is 2-10 times larger than the default value, the RBF kernel outperforms the performance obtained by the linear kernel.

5.4 Qualitative Analysis with Embedding Vectors

Previously, we have shown the effectiveness of our proposed method with the quantitative experiment results. As mentioned before, an important merit of MC-TEM is that it is convenient to measure the association between different context features. Now, we present qualitative analysis why it works well via the cosine similarities between embedding vectors of context information.

Table 5 presents several examples on context association. In each example, a *query context* is given and then the top five associated locations are returned. We can observe that (1) location and category contexts are useful to retrieve locations with similar services; (2) day and hour contexts can identify locations with significant temporal characteristics (e.g., traffic-based locations); city context is typically related to popularly visited places. More interestingly, the combination of multiple context features can lead to meaningful recommendation results. For example, the user "USER3584" will be recommended to different sets of locations with two different category contexts, namely *Food* and *Shop*.

Similar to the finding "`King-Man` $\approx$ `Quene-Woman`" in text processing [31], we identify interesting patterns existed

TABLE 5
Illustrative Examples for Query Context with the Corresponding Top Five Related Locations on the Foursquare$_S$ Dataset, Where ✓ Indicates the Location was Actually Visited by the User and ✘ Otherwise

Type	Query Context	Top five related locations
Location	HK Star Seafood Resturant	Bo Ling's Chinese Restaurant$_{\text{food}}$, Soho Restaurant$_{\text{food}}$, Guadalajara Grill$_{\text{food}}$, Flavor Del Mar$_{\text{food}}$, Perch$_{\text{food}}$
Service Category	Food	Brown Owl Coffee$_{\text{food}}$, Teriyaki Maki$_{\text{food}}$, New Ca Mau Restaurant$_{\text{food}}$, Espresso Roma Cafe$_{\text{food}}$, Ajisen Ramen$_{\text{food}}$
	Museum	San Francisco Museum of Modern Art$_{\text{museum}}$, San Jose Museum of Art, MOCA$_{\text{museum}}$, The Metropolitan Museum of Art$_{\text{museum}}$, Griffith Observatory$_{\text{museum}}$
	7 o' clock	MTA Subway$_{\text{travel}}$, California Highway Patrol$_{\text{travel}}$, MUNI Bus Stop$_{\text{travel}}$, Metro Bus$_{\text{travel}}$, USA Gas Station$_{\text{travel}}$
Time	12 o'clock	Starbucks$_{\text{food}}$, McDonald's$_{\text{food}}$, Denny's$_{\text{food}}$, Subway$_{\text{food}}$, Burger King$_{\text{food}}$
	Tuesday	Twitter, Inc.$_{\text{office}}$, US Post Office$_{\text{office}}$, YouTube HQ$_{\text{office}}$, Bank of America$_{\text{office}}$, First Team SnS Real Estate$_{\text{office}}$, Chegg HQ$_{\text{office}}$
	Sunday	Times Square$_{\text{recreation}}$, Landmark Theatres$_{\text{recreation}}$, Runyon Canyon Park$_{\text{recreation}}$, Alcatraz Island$_{\text{recreation}}$, Macy's$_{\text{recreation}}$
City	Anaheim	Disneyland$_{\text{recreation}}$, Metrolink Anaheim Station$_{\text{travel}}$, Anaheim Hills Medical Center$_{\text{office}}$, Disney California Adventure Park$_{\text{recreation}}$, City National Grove of Anaheim$_{\text{recreation}}$
	San Diego	Balboa Park$_{\text{recreation}}$, San Diego International Airport$_{\text{travel}}$, San Diego Zoo$_{\text{recreation}}$, Port Of San Diego$_{\text{travel}}$, Fashion Valley Trolley Station$_{\text{travel}}$ and Transit Center$_{\text{travel}}$
User + Service Category	USER3584 + Food	KFC$_{\text{food}}$(✓), Vallejo's Restaurant$_{\text{food}}$(✓), Starbucks$_{\text{food}}$(✘), Roli Roti Gourmet Rotisserie$_{\text{food}}$ (✓), Bistro Burger$_{\text{food}}$(✓)
	USER 3584 + Shop	Walgreens$_{\text{shop}}$(✓), Free People$_{\text{shop}}$(✓), Wells Fargo - Ocean$_{\text{shop}}$(✓), Old Navy$_{\text{shop}}$ (✘), Apple Store$_{\text{shop}}$(✓)

in trajectory data via embedding vectors, i.e., $\texttt{context}_{A_1}$-$\texttt{context}_{A_2} \approx \texttt{context}_{B_1}$-$\texttt{context}_{B_2}$. In Fig. 2, we present four interesting example pairs for the above pattern. The first two are city-location pairs, while the last two are location-location pairs. By picking up one example `Twitter, Inc.-Cafe des Amis` $\approx$ `Google - Penelope`, it is interesting to find that Twitter employees are likely to drink in *Cafe des Amis* while Google employees are likely to drink in *Penelope*, which might be caused by the geographical factor.

6 Conclusions

In this paper, we have proposed a Multi-Context Trajectory Embedding Model to explore contexts in a systematic way, including user-level, trajectory-level, location-level and temporal contexts. Our model is flexible to characterize various kinds of contexts for different applications using trajectory data. All the contextual information is represented in the same embedding space, which makes it convenient to analyze the association among different contexts. We apply the proposed general model to two challenging tasks, namely location recommendation and social link prediction. We conduct extensive experiments on three real-world datasets. Extensive experimental results demonstrated the superiority of our MC-TEM model over several state-of-the-art methods. Our experimental results indicate that the distributed learning based approach is very promising for mining trajectory data.

Currently, only a simple distributed representation architecture has been employed for modelling trajectory data. In the future, more advanced deep learning models such as Convolutional Neural Networks[5] can be explored for feature learning. We will also consider applying the current model to solve more real geo-based applications.

Acknowledgments

Xin Zhao is the corresponding author and co-first author. The work was partially supported by National Natural Science Foundation of China under the grant number 61502502, the National Key Basic Research Program (973 Program) of China under the grant number 2014CB340403 and the support by Beijing Natural Science Foundation under the grant number 4162032. Ningnan Zhou is partially funded by China Scholarship Council (File No. 201506360199).

References

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, 1995, pp. 3–14.

[2] H. A. Salas, S. Bringay, F. Flouvat, N. S.-Folcher, and M. Teisseire, "The pattern next door: Towards spatio-sequential pattern discovery," in *Proc. Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, 2012, pp. 157–168.

[3] R. B.-Yates, D. Jiang, F. Silvestri, and B. Harrison, "Predicting the next app that you are going to use," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 285–294.

[4] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: A survey," *Geoinformatica*, vol. 19, no. 3, pp. 525–565, 2015.

[5] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, pp. 1554–1563, 1966.

5. http://deeplearning.net/tutorial/lenet.html

[6] P. Bhargava, T. Phan, J. Zhou, and J. Lee, "Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 130–140.

[7] Y-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 111–118.

[8] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. Nat. Conf. Artif. Intell.*, 2012, pp.17–23.

[9] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2605–2611.

[10] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1082–1090.

[11] M. Cho, J. Sun, O. Duchenne, and J. Ponce, "Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2091–2098.

[12] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the gap between physical location and online social networks," in *Proc. Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2010, pp. 119–128.

[13] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2005, pp. 485–492.

[14] N. Eagle, A. (Sandy) Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci.*, vol. 106, no. 36, pp. 15274–15278, 2009.

[15] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *Proc. Nat. Conf. Artif. Intell.*, 2015, pp. 1721–1727.

[16] H. Gao, J. Tang, and H. Liu, "Gscorr: Modeling geo-social correlations for new check-ins on location-based social networks," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2012, pp. 1582–1586.

[17] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 330–339.

[18] J. Gudmundsson and M. V. Kreveld, "Computing longest duration flocks in trajectory data," in *Proc. 14th Annu. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2006, pp. 35–42.

[19] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 307–361, 2012.

[20] C. S. Jensen, D. Lin, and B. C. Ooi, "Continuous clustering of moving objects," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 9, pp. 1161–1174, Sep. 2007.

[21] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 1068–1080, Aug. 2008.

[22] P. F. Lazarsfeld and R. K. Merton, "Friendship as a social process: A substantive and methodological analysis," in *Freedom and Control in Modern Society*, Van Nostrand, pp. 18–66, 1954.

[23] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[24] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "Lars: A location-aware recommender system," in *Proc. Int. Conf. Data Eng.*, 2012, pp. 450–461.

[25] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proc. Annu. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2008, pp. 34:1–34:10.

[26] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 831–840.

[27] B. Liu and H. Xiong, "Point-of-interest recommendation in location based social networks with topic and location awareness," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 396–404.

[28] W. Luo, H. Tan, L. Chen, and L. M. Ni, " Finding time period-based most frequent path in big trajectory data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 713–724.

[29] W. Mathew, R. Raposo, and B. Martins, " Predicting future locations with hidden markov models," in *Proc. Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2012, pp. 911–918.

[30] M. McPherson, L. S.-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, pp. 415–444, 2001.

[31] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, 2013.

[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[33] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1751–1758.

[34] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2005, pp. 246–252.

[35] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *Proc. ASE/IEEE Int. Conf. Social Comput. 2012 ASE/IEEE Int. Conf. Privacy, Security, Risk Trust*, 2012, pp. 144–153.

[36] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabasi, "Returners and explorers dichotomy in human mobility," *Nature Commun.*, vol. 6, no. 8166, pp. 30:1–30:23, 2015.

[37] J. Pei, J. Han, B. M.-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, " Prefixspan,: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. Int. Conf. Data Eng.*, 2001, pp. 215–224.

[38] B. Perozzi, R. A.-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[39] H. Pham, C. Shahabi, and Y. Liu, "Ebm: An entropy-based model to infer social strength from spatiotemporal data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 265–276.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[41] S. Scellato, R. Lambiotte, A. Noulas, and C. Mascolo, "Socio-spatial properties of online location-based social networks," in *Proc. Int. Conf. Weblogs Social Media*, 2011, pp. 329–336.

[42] C. Song, T. Koren, P. Wang, and A.-L. Barabasi, "Modelling the scaling properties of human mobility," *Nature Phys.*, vol. 6, pp. 818–823, 2010.

[43] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1100–1108.

[44] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 403–412.

[45] X. Xiao, Y. Zheng, Q. Luo, and X. Xie, "Finding similar users using category-based location history," in *Proc. Annu. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2010, pp. 442–445.

[46] X. Yan, J. Han, and R. Afshar, "Clospan: Mining closed sequential patterns in large datasets," in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 166–177.

[47] D. Yang, D. Zhang, and B. Qu, "Participatory cultural mapping based on collective behavior data in location-based social networks," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, pp. 30:1–30:23, 2016.

[48] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 325–334.

[49] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: A location-content-aware recommender system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 221–229.

[50] N. J. Yuan, F. Zhang, D. Lian, K. Zheng, S. Yu, and X. Xie, "We know how you live: Exploring the spectrum of urban lifestyles," in *Proc. 1st ACM Conf. Online Soc. Netw.*, 2013, pp. 3–14.

[51] Q. Yuan, G. Cong, and C.-Y. Lin, "Com: A generative model for group recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 163–172.

[52] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 363–372.

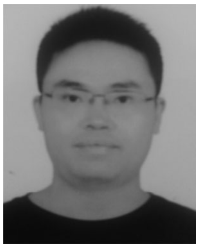
[53] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, 'Who, where, when and what: Discover spatio-temporal topics for twitter users," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 605–613.

[54] Q. Yuan, G. Cong, and A. Sun, "Graph-based point-of-interest recommendation with geographical and temporal influences," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2014, pp. 659–668.

[55] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, no. (1-2), pp. 31–60, 2001.

[56] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with GPS history data," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 1029–1038.

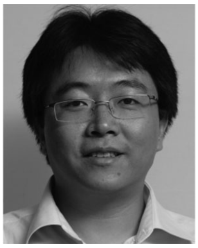
[57] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Trans. Web*, vol. 5, no. 1, pp. 1–44, 2011.



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

Wayne Xin Zhao received the PhD degree from Peking University in 2014. He is currently an assistant professor in the School of Information, Renmin University of China. His research interests are web text mining and natural language processing. He has published several referred papers in international conferences and journals such as ACL, EMNLP, COLING, ECIR, CIKM, SIGIR, SIGKDD, AAAI, IJCAI, *ACM Transactions on Information Systems*, *ACM Transactions on Knowledge Discovery from Data*, *ACM Transactions on Intelligent Systems and Technology*, *IEEE Transactions on Knowledge and Data Engineering*, *Knowledge and Information Systems*, and *World Wide Web Journal*. He is a member of the IEEE.

Xiao Zhang received the master's degree from Renmin University and the PhD degree from the Institute of Computing Technology, Chinese Academy of Science, in 1998 and 2001, respectively, both in computer science and technology. He is an associate professor at the School of Information, Renmin University of China. His research interests include database architecture and big data management. He specializes in design and implementation of database management system, and is familiar with the open source database system PostgreSQL.



Ji-Rong Wen is a professor in the School of Information, Renmin University of China. He is also the director of the Beijing Key Laboratory of Big Data Management and Analysis Methods. Before that, he was a senior researcher and group manager of the Web Search and Mining Group at MSRA. His main research interests include big data management & analytics, information retrieval, data mining, and machine learning. He is currently the associate editor of the *ACM Transactions on Information Systems* (*TOIS*). He is a senior member of the IEEE.

Shan Wang received the bachelor's degree in physics from Peking University from 1962 to 1968. She is a professor in the School of Information, Renmin University of China. Her research interests include RDBMS (XDB, KingbaseES), parallel and distributed DBMS (PBase) and main memory DBMS, data warehouse, DB-IR, and big data management. She was the cochair of VLDB'02 and honorary chair of SIGMOD'07. She is a member of the IEEE.