Modeling Spatial Trajectories With Attribute Representation Learning

Meng Chen, *Member, IEEE*, Yan Zhao, Yang Liu, *Member, IEEE*, Xiaohui Yu, *Member, IEEE*, and Kai Zheng, *Senior Member, IEEE*

Abstract—The widespread use of positioning devices has given rise to many trajectories, with each having three explicit attributes: *user ID*, *location ID*, and *time-stamp* and an implicit attribute: *activity type* (akin to "topic" in text mining). To model these trajectories, existing works learn different attribute representations by either introducing latent activity types based on topic models or transforming the location and time context into a low-dimensional space via embedding techniques. In this paper, we propose a holistic approach named Human Mobility Representation Model (HMRM) to simultaneously produce the vector representations of all four (explicit and implicit) attributes. The merits of HMRM lie in that: (1) it models the latent activity types and learns trajectory attribute embeddings in an integrated manner, and (2) it connects the activity-related distributions and these attributes embeddings by adding a newly designed collaborative learning component, and makes them mutually exchanged to take the best of both worlds. We apply HMRM to both unsupervised and supervised tasks including two activity evaluation tasks and two embedding evaluation tasks, on two real check-in datasets collected from Foursquare. Experimental results show that HMRM could not only improve the performance of capturing latent activity types, but also learn better trajectory embeddings.

Index Terms—Human mobility representation model, attribute representation learning, activity modeling, trajectory embedding, collaborative learning

1 Introduction

The increasing prevalence of location acquisition technologies (e.g., global positioning system enabled mobile devices and video capturing equipments) has made it possible to collect a deluge of users' spatio-temporal trajectories, where a trajectory is defined as the sequence of locations of a user as a function of time. For instance, check-in records collected by social network sites (e.g., Foursquare and Gowalla) over time form a trajectory of the locations visited by a user [22], [40]; the Vehicle Passage Records (VPRs) acquired via the surveillance cameras installed on city streets constitute vehicle trajectories [6]. Both types of trajectory data contain explicit attributes including *user ID*, *location ID*, and *time-stamp*. Besides, there exist hidden semantic structures underlying users' trajectories, and some studies [1], [2] define them as the latent activity types (akin to "topics" in text mining). The *activity types* are considered as the implicit attributes of trajectories. We tackle the task of *attribute representation learning*, which is to find representations of trajectory attributes. These learned attribute representations could capture the characteristics of trajectories, e.g., sequential patterns and semantic properties, and be used as the feature vectors for a wide spectrum of downstream applications, e.g., location categorization, user similarity computation and user classification [28], [32], [46].

Existing work on representation learning of trajectory attributes can be grouped into two categories. The first category of methods are concerned with learning the embeddings of the locations [7], [21], [29], [38], [41], [42], [43], [45]. They follow the distributional hypothesis that locations occurring in similar contexts tend to have similar semantic properties, and project them into closer embedding vectors in the latent space. The second category of methods model the joint distribution of users, activity types and locations based on users' trajectories [1], [2], [16], [26]. They treat trajectories of a user as mixtures of latent activity types, which are in turn formulated as multinomial distributions over locations. Apparently, these two categories of approaches are good at different aspects of the attribute representation learning task, but neither is able to capture the interplay of the four attributes in a principled way.

In this work, we consider a holistic approach to attribute representation learning that takes the best of both worlds. Our objective is to develop a single Human Mobility Representation Model (HMRM) that is able to simultaneously produce the vector representations of all four (explicit and implicit) attributes. A first attempt to achieve this is to build a model that integrates one model from each of the aforementioned two categories of approaches using linear combination. However, since these two individual models do not

- M. Chen is with the School of Software, Shandong University, Jinan, Shandong 250100, China. E-mail: mchen@sdu.edu.cn.
- Y. Zhao is with the School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China. E-mail: zhaoyan@suda.edu.cn.
- Y. Liu is with the Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON N2L 3C5, Canada. E-mail: yangliu@wlu.ca.
- X. Yu is with with the School of Information Technology, York University, Toronto, Toronto, ON M3J 1P3, Canada. E-mail: xhyu@yorku.ca.
- K. Zheng is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610051, China. E-mail: zhengkai@uestc.edu.cn.

Manuscript received 23 Mar. 2019; revised 2 May 2020; accepted 5 June 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding authors: Kai Zheng.)
Recommended for acceptance by H. Lee.
Digital Object Identifier no. 10.1109/TKDE.2020.3001025

share any common attribute representations, optimizing for the linear combination reduces to learning the two models separately, defeating the purpose of this combined model. Therefore, we propose a novel method that trains both models simultaneously by adding a newly designed collaborative component. Specifically, HMRM consists of three components: the activity modeling component for learning the latent activity types, the trajectory embedding component for generating embeddings of the explicit attributes, and the collaborative learning component for making the connection between the attribute embeddings and the activity types.

A major difficulty in learning the proposed HMRM lies in different formulations of the three components. These activity modeling methods are usually probabilistic generative models while the trajectory embedding methods are mainly based on artificial neural networks, making it challenging to find a coherent way to learn them at the same time. The basic idea of our solution to this problem is to identify a "lowest common denominator" representation of the individual components and transform all of them into this new representation. Fortunately, for two particular models, PLSA [13] (Probabilistic Latent Semantic Analysis, a generative model that can be used for capturing activity types) and Skip-Gram [27] (a method that can be used for learning the attribute embeddings), their equivalences to matrix factorization operations have already been established in the literature [9], [20]. Therefore, we propose to formulate the three components via matrix factorization. To be more specific, the activity modeling component factorizes the user-location and user-time matrices and learns multiple activity-related distributions (e.g., the user-activity distribution and the activity-location distribution); the trajectory embedding component factorizes the location co-occurrence and location-time matrices and learns multiple embeddings (e.g., location embeddings and time embeddings); the collaborative learning component factorizes the activity-location/time distribution into the inner product of the corresponding location/time embeddings and activity embeddings. By this means, we establish direct connections between attribute embeddings and activity types, and regulate distributional activity semantics accordingly. Finally, we perform parameter inference through Alternating Least Squares matrix factorization method.

To evaluate how well HMRM captures latent activity types and learns attribute embeddings, we perform both unsupervised and supervised quantitative tasks including two activity evaluation tasks and two embedding evaluation tasks, on two real check-in datasets collected from Foursquare. We show that by modeling the activity types and these attributes embeddings collaboratively, HMRM outperforms the baselines on these evaluation tasks. We also provide qualitative analysis on the activity embeddings and the time embeddings. Further, we make detailed analysis to explain how activity types and attribute embeddings can collaboratively enhance the quality of each other. Finally, we give efficiency analysis on the proposed HMRM.

The main contributions of this paper are as follows.

- We propose an integrated Human Mobility Representation Model (HMRM) to learn dense representations of all four (explicit and implicit) attributes (i.e., user, location, time, and activity type) from trajectory data. HMRM captures the latent activity types and learns location/time/activity embeddings simultaneously.
- HMRM establishes connections between the activity types and these attributes embeddings by adding a newly designed separate collaborative learning component. In this way, it regulates locations/time with similar activity distributions to be close in the embedding space and nearby locations/time to have similar activity distributions.
- Experimental results show that the collaborative learning component could help learn better attribute embeddings and assist in capturing more coherent activity types.

2 Related Work

There mainly exist two kinds of popular methods to learn attribute representations from trajectory data – activity modeling and trajectory embedding.

2.1 Activity Modeling

Activity modeling, which aims at learning the latent activity types in trajectories, has received much attention recently. Some recent studies [12], [14], [25], [33] consider both trajectories and the background semantic labels of locations such as restaurant, store and park, to indicate activity types. Xie *et al.* [33] study the trajectory semantic join problem to find user activity sequences from a set of trajectories, in which they determine the activity types of a trajectory based on the semantic labels of nearby locations and the duration. Gong *et al.* [12] calculate the visit probability of each nearby location given the destination and time, and use the semantic information of possible visited locations to infer the trip purposes of taxi passengers. Huang *et al.* [14] first model the spatio-temporal attractiveness of locations to discover the activity spot and duration from raw GPS trajectories. Further, they present a novel approach to estimate the potential possibilities for activities with the intersections of trajectories and spatio-temporal attractiveness prisms. Liu *et al.* [25] leverage the trip context information, i.e., the semantic information of locations around destination, to describe the activity types of a trip.

On the other hand, there are some studies [1], [2], [16], [18], [26], [44] that focus on learning the latent activity types from sequences of semantically unlabeled locations as opposed to semantically labeled trajectories. Joseph *et al.* [16] model users' check-in behaviors using LDA (Latent Dirichlet Allocation [3]), and assume that every user can be represented by multiple activity types, wherein each check-in by that user is motivated by one or more of these activity types. Long *et al.* [26] directly employ the LDA model to investigate the local geographic topics based on the users' check-ins in Foursquare. Alharbi and Zhang [2] propose a Social Trajectory Amplification and Representation learning model, which considers both extrinsic (social network) and intrinsic (user trajectory) factors and infers the activity types from unlabeled and incomplete location-user traces by leveraging the network of friends. Further, Alharbi *et al.* [1] propose a model named HuMoR based on LDA, which extracts the activity types from community-level sequences while making use of metadata (e.g., user social graph, visiting time) associated with location IDs. Zheng and Ni [44] propose a generative model that models both location and time to understand human behaviors from mass amount of mobile data. Specifically, they first draw a behavior pattern z, and then draw a latent state, a time point and a location id dependent on z. Kurashima *et al.* [18] propose a Geo Topic Model to estimate both the user's interest and the user's spatial area of activity, which models both the locations and the geotags (e.g., users' reviews) of these visited locations. However, all these methods are built upon LDA and only model the global user-location frequency matrix.

In addition, some methods [10], [31] organize trajectory data in the form of higher-order tensor and leverage tensor decomposition to model the relationships between multiple modes. For example, Fan *et al.* [10] leverage a non-negative tensor factorization approach to factorize the spatial distribution of POIs. They use a people flow tensor to model the relationship of POIs and human mobility, where the three-way tensor contains the number of regions, time-slices and sample days respectively. Takeuchi *et al.* [31] propose a novel tensor factorization technique called Non-negative Multiple Tensor Factorization, which naturally incorporates auxiliary data tensors into standard tensor factorization, to solve the data sparsity problem. However, users' check-in data are extremely sparse, and if we build the user-location-time tensor, only about 0.01 percent elements of the tensor will have values, which is difficult to learn meaningful attribute representations.

2.2 Trajectory Embedding

Inspired by the success of word2vec [27], many studies [11], [24], [41], [43], [45] adopt the framework of word2vec to learn trajectory embeddings with check-in data. Liu *et al.* [24] model the check-in sequences based on the Skip-gram model and learn the latent representation for a location to capture the influence of its context. They consider the confidences of observed user preferences for locations with a pair-wise ranking loss and leverage the latent representations for personalized location recommendations. Further, Zhou *et al.* [45] propose a general Multi-Context Trajectory Embedding Model (MC-TEM), which leverages multiple contexts, including users, trajectories, surrounding locations and their corresponding category labels, as well as the temporal factor. Note that, all the context information is represented in the same embedding space. Similarly, Zhao *et al.* [43] propose a Time-Aware Trajectory Embedding Model (TA-TEM) which considers surrounding locations, dynamic user preference and the temporal factor. Specially, they jointly model multiple kinds of temporal factors in a unified manner. However, all these models adopt the framework of word2vec, and only consider the local contexts.

Besides the contextual check-in information and the various temporal characteristics, some studies [4], [11], [39], [41] leverage external information (e.g., geographical information and text content) to learn trajectory embeddings. Zhao *et al.* [41] propose a temporal location embedding model (Geo-Teaser) which captures the geographical influence. Specifically, they discriminate the unvisited POIs according to geographical information and incorporate the geographical influence into the pairwise preference ranking method. Feng *et al.* [11] present a new latent representation model named POI2Vec which captures user preference, location sequential transition influence, and geographical influence for predicting potential visitors for a given location. Yao *et al.* [39] propose a method named Semantics-Enriched Recurrent Model (SERM) for the location prediction with semantic trajectory data. SERM jointly learns the embeddings of multiple factors (e.g., location, keyword) and the transition parameters of a recurrent neural network. Chang *et al.* [4] propose a content-aware POI embedding model to utilize the text content of a POI to boost the performance of prediction. In addition, except the check-in trajectory data, Chen *et al.* [7] focus on the traffic trajectory data and propose a Mobility Pattern Embedding (MPE) method. They consider the characteristics of urban road networks and embed the time slots, current locations and next locations together as points in a latent space.

In addition, there also exist some methods [17], [23], [35] using the Recurrent Neural Networks to model the sequential patterns of trajectories, in which the trajectory embeddings can be learned as by-products. For instance, Liu *et al.* [23] propose Spatial Temporal Recurrent Neural Networks (ST-RNN) to model the local temporal and spatial contexts for mining mobility patterns. Yang *et al.* [35] present a neural network by modelling both the social networks and mobile trajectories, in which they employ RNN to capture the sequential relatedness in mobile trajectories. Kong and Wu [17] propose a hierarchical spatio-temporal LSTM model, leveraging the historical visit information and spatio-temporal factors for the location prediction. However, these RNN-based (or LSTM-based) methods focus on storing statistical weights for long-term transitions in a trajectory, and use the side features (e.g., friendship network) that do not exist in our trajectory data.

To the best of our knowledge, all the methods that mine human mobility patterns from trajectory data either model the latent activity types or learn the trajectory embeddings. Overall, our HMRM could learn these dense attribute representations including the activity-related distributions and the trajectory embeddings simultaneously, and model the relations between the distributions and the corresponding embeddings collaboratively.

3 Problem Definition

We first introduce some preliminary concepts and then define the problem studied in this paper.

Definition 1 (Trajectory). *Given a user u, a* trajectory T_u *is defined as a time-ordered sequence of location-time pairs: $\langle (l_1, t_1), (l_2, t_2), \ldots, (l_n, t_n) \rangle$, where l and t are the ID of location and time-stamp respectively.*

Definition 2 (Trajectory Attributes). *Given a trajectory T_u, there exist three explicit* trajectory attributes, *i.e.,* user ID, location ID, *and* time. *We also include the latent activity type a as an implicit attribute reflecting the hidden semantic structures underlying users' trajectories.*

Given the trajectories of all the users, we build 1) the user-location frequency matrix $\mathbf{U^l}$ and the user-time frequency matrix $\mathbf{U^t}$ (detailed in Section 4.2.1), and 2) the shifted location co-occurrence PMI (Point-wise Mutual Information) matrix $\mathbf{L^l}$ and the shifted location-time PMI

TABLE 1
Notations and Descriptions

Notations	Descriptions
T, u, a	Trajectory, User, Activity type
l, c, t	Target Location, Context location, Time slots
K	Number of activity types
M	Dimensionality of embedding space
N_l, N_u, N_t	Number of locations, users, and time slot
$\boldsymbol{v}_l, \boldsymbol{v}_t$	embedding vectors for location, time
$\mathbf{U}^\mathbf{l} \in \mathbb{R}^{N_u \times N_l}$	User-location frequency matrix
$\mathbf{U}^\mathbf{t} \in \mathbb{R}^{N_u \times N_t}$	User-time frequency matrix
$\mathbf{L}^\mathbf{l} \in \mathbb{R}^{N_l \times N_l}$	Location co-occurrence PMI matrix
$\mathbf{L}^\mathbf{t} \in \mathbb{R}^{N_l \times N_t}$	Location-time PMI matrix
$\mathbf{A}^\mathbf{l} \in \mathbb{R}^{N_l \times K}$	Activity-location distribution matrix
$\mathbf{A}^\mathbf{t} \in \mathbb{R}^{N_t \times K}$	Activity-time distribution matrix
$\boldsymbol{\Theta} \in \mathbb{R}^{N_u \times K}$	User-activity distribution matrix
$\mathbf{E}^\mathbf{a} \in \mathbb{R}^{K \times M}$	Activity embedding matrix
$\mathbf{E}^\mathbf{l} \in \mathbb{R}^{N_l \times M}$	Target Location embedding matrix
$\mathbf{E}^\mathbf{c} \in \mathbb{R}^{N_l \times M}$	Context location embedding matrix
$\mathbf{E}^\mathbf{t} \in \mathbb{R}^{N_t \times M}$	Time slot embedding matrix

matrix $\mathbf{L}^\mathbf{t}$ (detailed in Section 4.2.2). With $\mathbf{U}^\mathbf{l}$, $\mathbf{U}^\mathbf{t}$, $\mathbf{L}^\mathbf{l}$, and $\mathbf{L}^\mathbf{t}$, we aim at learning the dense attribute representations including the activity-related distributions (i.e., the user-activity distribution $\boldsymbol{\Theta}$, the activity-location distribution $\mathbf{A}^\mathbf{l}$ and the activity-time distribution $\mathbf{A}^\mathbf{t}$) and the trajectory embeddings (i.e., the activity embeddings $\mathbf{E}^\mathbf{a}$, the target location embedddings $\mathbf{E}^\mathbf{l}$, the context location embeddings $\mathbf{E}^\mathbf{c}$ and the time embeddings $\mathbf{E}^\mathbf{t}$). These trajectory attribute representations could be used in many applications, e.g., location categorization, user similarity computation, and user persona. The major notations used in this paper are listed in Table 1.

4 Human Mobility Representation Model

In this section, we first give the overview of the proposed HMRM, and then describe the activity modeling component, the trajectory embedding component, and the collaborative learning component in detail respectively. Finally, we present the training algorithm for HMRM.

4.1 Overview of HMRM

The trajectory data contains both the explicit attributes (i.e., user, location and time) and the implicit attribute (i.e., the latent activity type), and the interplay of the four attributes forms the mobility patterns of users. Therefore, we need to simultaneously learn the vector representations of these attributes with a holistic model. The trajectory embedding methods [38], [42], [45] could learn fine-grained attribute embeddings, and the activity modeling methods [1], [16], [26] are able to capture valuable activity types. It is natural to integrate an activity modeling method and a trajectory embedding model with a linear function. As there do not exist any common attribute representation in both individual models, optimizing for the linear integration is the same as learning attribute embeddings or activtiy-related distributions separately. Hence it is essential to establish direct connections between these embeddings and distributions and model them collaboratively.

To learn representations of the four attributes and collaboratively model attribute embeddings and activity structures,

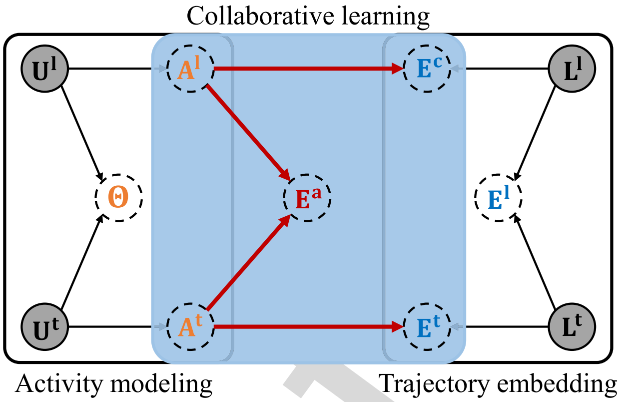


Fig. 1. The framework of HMRM. The gray circle denotes a variable that is (assumed as) observed, and dashed circles denote parameters in HMRM. $\mathbf{U}^\mathbf{l}$ is the user-location frequency matrix, $\mathbf{U}^\mathbf{t}$ is the user-time frequency matrix, $\mathbf{L}^\mathbf{l}$ is the shifted location co-occurrence PMI matrix, and $\mathbf{L}^\mathbf{t}$ is the shifted location-time PMI matrix. The learned attribute representations include the activity-related distributions (i.e., the user-activity distribution $\boldsymbol{\Theta}$, the activity-location distribution $\mathbf{A}^\mathbf{l}$ and the activity-time distribution $\mathbf{A}^\mathbf{t}$) and the trajectory embeddings (i.e., the activity embeddings $\mathbf{E}^\mathbf{a}$, the target location embedddings $\mathbf{E}^\mathbf{l}$, the context location embeddings $\mathbf{E}^\mathbf{c}$ and the time embeddings $\mathbf{E}^\mathbf{t}$).

we propose an integrated Human Mobility Representation Model (HMRM), which consists of the activity modeling component, the trajectory embedding component, and the collaborative learning component. HMRM is built based on the following three basic assumptions:

- *activity modeling component*: each user can be represented as a mixture of activity types, where each activity type assigns high probabilities to only a small number of locations and time slots;
- *trajectory embedding component*: locations appearing with similar context locations or in the same time slots tend to have similar semantic labels, hence should be mapped closer in the embedding space;
- *collaborative learning component*: locations (or time slots) close to each other in the embedding space tend to be associated with similar activity types and vice versa.

HMRM takes advantages of matrix factorization to formulate the three components, as PLSA [13] (a generative model that can be used for learning activity types) and Skip-Gram [27] (a method that can be used for learning the attribute embeddings) have been proven to be equivalent to optimizing objective functions through matrix factorization [9], [20].

The framework of HMRM is shown in Fig. 1, where the activity-location/time distributions and location/time embeddings are shared in two components. Let us take a running example to illustrate the proposed HMRM. Suppose a user usually visits restaurants (e.g., *sushi restaurant* and *ramen restaurant*) at noon. Given trajectories of this user, we first construct the user-location frequency matrix $\mathbf{U}^\mathbf{l}$ and the user-time frequency matrix $\mathbf{U}^\mathbf{t}$, and learn the latent activity types (i.e., activity-related distributions) via decomposing $\mathbf{U}^\mathbf{l}$ and $\mathbf{U}^\mathbf{t}$ simultaneously, as shown in the activity modeling component of Fig. 1. In this part, we expect that it has a high probability that *sushi restaurant* and *ramen restaurant* belong to the same activity type (e.g., *dining*). On the other hand, as shown in the trajectory embedding component of Fig. 1, we factorize the shifted location co-occurrence PMI matrix $\mathbf{L^l}$ and the shifted location-time PMI matrix $\mathbf{L^t}$, to learn location embeddings and time embeddings accordingly. In this part, we expect that *sushi restaurant* and *ramen restaurant* are close to each other in the embedding space. Further, we assume that the distances between location (or time) embeddings correlate with their activity similarities, and realize this assumption by factorizing the activity-location distribution matrix $\mathbf{A^l}$ and the activity-time distribution matrix $\mathbf{A^t}$. Via the collaborative learning component, we further regulate *sushi restaurant* and *ramen restaurant* to be close in the embedding space and to have similar activity distributions.

4.2 Model Description

4.2.1 Activity Modeling Component

Intuitively, users usually visit different locations according to their preferences. For example, the visited locations of students are mainly relevant to "studying" on weekdays, while the retirees are more likely to visit "shopping" and "relaxing" regarded places. Hence we first model the activity types concerning human mobility patterns from users' trajectories. Formally, trajectories of a user are modeled as mixtures of latent activity types, which are in turn formulated as multinomial distributions over locations. As Ding *et al.* [9] have proven the equivalence between PLSA [13] and NMF (Non-negative Matrix Factorization) [19] for optimizing the same objective, we model the activity types through NMF. More specifically, we decompose the user-location frequency matrix $\mathbf{U^l}$ into the inner product of the user-activity distribution matrix $\mathbf{\Theta}$ and the activity-location distribution matrix $\mathbf{A^l}$.

Furthermore, users may exhibit different temporal behaviors, and the temporal distributions vary from one activity to another. For instance, the activity regarding transportation usually occurs during the morning and evening rush hours. Therefore, we also need to consider the temporal factor in this component. Similarly, the user-time frequency matrix $\mathbf{U^t}$ is decomposed into the inner product of the user-activity distribution matrix $\mathbf{\Theta}$ and the activity-time distribution matrix $\mathbf{A^t}$. We discretize a day into hourly unit and distinguish weekdays from weekends, i.e., a number ranging from 0 to 47 is used to denote the hour index.

Finally, we factorize both $\mathbf{U^l}$ and $\mathbf{U^t}$ (as shown in the activity modeling component of Fig. 1), and define the objective as

$$\begin{aligned}
&\min_{\mathbf{\Theta},\mathbf{A^l},\mathbf{A^t}} \lambda_l \parallel \mathbf{U^l} - \mathbf{\Theta}{\mathbf{A^l}}^T \parallel_2^2 + (1-\lambda_l) \parallel \mathbf{U^t} - \mathbf{\Theta}{\mathbf{A^t}}^T \parallel_2^2,\\
&subject\ to:\\
&\mathbf{\Theta} \geq 0, \mathbf{A^l} \geq 0, and\ \mathbf{A^t} \geq 0,
\end{aligned} \tag{1}$$

where λ_l balances the two parts, and $\parallel \cdot \parallel_2$ is the euclidean norm. The non-negativity of NMF ensures the explainability of the user-activity distribution $\mathbf{\Theta}$, the activity-location distribution $\mathbf{A^l}$ and the activity-time distribution $\mathbf{A^t}$. In our model, the value of ${\mathbf{U^l}}_{ij}$ is the raw frequency that a user u_i visits a location l_j, and the value of ${\mathbf{U^t}}_{ij}$ is the raw frequency that a user u_i checks in within time slot t_j. In this way, the users' preference information encoded in $\mathbf{U^l}$ and $\mathbf{U^t}$ is fully considered in Eq. (1), and the attribute representations (i.e., $\mathbf{\Theta}$, $\mathbf{A^l}$, $\mathbf{A^t}$) could be learned accordingly.

4.2.2 Trajectory Embedding Component

As a kind of sequential data, it is important to consider the local context when learning semantic relatedness for attributes in trajectories [45]. We learn a low-dimensional representation for each location under the assumption that locations appearing with similar context locations tend to have similar semantics. Levy *et al.* [20] have found that the objective of Skip-Gram [27] (a word2vec model) is implicitly factorizing a shifted positive word co-occurrence PMI matrix. Therefore, we could learn the location embeddings by decomposing the shifted positive location co-occurrence PMI matrix $\mathbf{L^l}$ into the inner product of the target location embedding matrix $\mathbf{E^l}$ and the context location embedding matrix $\mathbf{E^c}$. The matrix $\mathbf{L^l} \in \mathbb{R}^{N_l \times N_l}$ is constructed as:

$$\begin{aligned}
&{\mathbf{L^l}}_{i,j} = \max(PMI(l_i, c_j), 0),\\
&PMI(l_i, c_j) = \log \frac{\sharp(l_i, c_j) \times \mid \mathcal{D}^l \mid}{\sharp(l_i) \times \sharp(c_j)},
\end{aligned} \tag{2}$$

where l_i is a target location, c_j is a context location, and N_l is the number of locations. We denote the collection of observed target locations and context pairs as $\mathcal{D}^l$. We use $\sharp(l_i, c_j)$ to denote the number of times the pair (l_i, c_j) appears in $\mathcal{D}^l$. Similarly, $\sharp(l_i)$ and $\sharp(c_j)$ are the number of times l_i and c_j occur in $\mathcal{D}^l$, respectively. $PMI(l_i, c_j)$ measures the association between a target location l_i and a context location c_j by calculating the logarithm of the ratio between their joint probability and their marginal probabilities.

Analogously, the temporal factor is also essential for trajectory embedding, as locations visited in the same time slot are more likely to have similar semantic labels. For example, people may go to different locations (e.g., *pizza* and *sushi restaurant*) to have lunch at noon. Hence we need to learn low-dimensional representations for both the time slots and the target locations. Similarly, we build a new shifted positive location-time PMI matrix $\mathbf{L^t} \in \mathbb{R}^{N_l \times N_t}$, and decompose it into the inner product of the target location embedding matrix $\mathbf{E^l}$ and the time embedding matrix $\mathbf{E^t}$. Finally, we exploit the local location and time contexts to learn trajectory embeddings by factorizing both $\mathbf{L^l}$ and $\mathbf{L^t}$ (as shown in the trajectory embedding component of Fig. 1):

$$\min_{\mathbf{E^l},\mathbf{E^c},\mathbf{E^t}} \lambda_l \parallel \mathbf{L^l} - \mathbf{E^l}{\mathbf{E^c}}^T \parallel_2^2 + (1-\lambda_l) \parallel \mathbf{L^t} - \mathbf{E^l}{\mathbf{E^t}}^T \parallel_2^2. \tag{3}$$

4.2.3 Collaborative Learning Component

We have discussed how to uncover the structures of activity types (e.g., activity-location distribution $\mathbf{A^l}$ and activity-time distribution $\mathbf{A^t}$) and learn the trajectory embeddings (e.g., context location embeddings $\mathbf{E^c}$ and time embeddings $\mathbf{E^t}$) respectively. However, these two steps should not be isolated from each other, as (1) locations with similar activity types (e.g., *dinning*) tend to be close in the embedding space and (2) users are more likely to have similar activities in those time slots which are in the nearby areas in the space. Hence it is reasonable to assume that the distances between location (or time) embeddings correlate with their activity type similarities.

By introducing the activity embedding matrix $\mathbf{E^a}$, we connect the activity-related distributions and the trajectory embeddings directly, as shown in the collaborative learning component in Fig. 1. Specifically, we factorize the activity-location matrix $\mathbf{A^l}$ into the inner product of the context location embedding matrix $\mathbf{E^c}$ and $\mathbf{E^a}$, and decompose the activity-time matrix $\mathbf{A^t}$ into the inner product of the time embedding matrix $\mathbf{E^t}$ and $\mathbf{E^a}$. That is,

$$\min_{\substack{\mathbf{A^l},\mathbf{A^t},\\ \mathbf{E^c},\mathbf{E^a},\mathbf{E^t}}} \lambda_l \parallel \mathbf{A^l} - \mathbf{E^c}\mathbf{E^a}^T \parallel_2^2 \; + (1-\lambda_l) \parallel \mathbf{A^t} - \mathbf{E^t}\mathbf{E^a}^T \parallel_2^2 . \tag{4}$$

The probability that a location l_i (or time t_i) being grouped into an activity type a_j can be computed by the inner product of the corresponding context location embedding (or time embedding) and activity embedding: $p(a_j|l_i) \propto \mathbf{E^c}_i \cdot \mathbf{E^a}_j^T$, $p(a_j|t_i) \propto \mathbf{E^t}_i \cdot \mathbf{E^a}_j^T$. Hence the objective is able to not only regulate locations and time slots with similar activity types to be close in the embedding space, but also make nearby locations and time slots in the embedding space to have similar activity-location distributions and activity-time distributions.

4.2.4 Unifying the Three Components

We integrate the above three components and propose the Human Mobility Representation Model. The overall objective is,

$$\begin{aligned}
\min_{\mathbf{\Phi}} & \underbrace{\lambda_l \parallel \mathbf{U^l} - \mathbf{\Theta}\mathbf{A^l}^T \parallel_2^2 + (1-\lambda_l) \parallel \mathbf{U^t} - \mathbf{\Theta}\mathbf{A^t}^T \parallel_2^2}_{\text{activity modeling component}} \\
& + \underbrace{\lambda_l \parallel \mathbf{L^l} - \mathbf{E^l}\mathbf{E^c}^T \parallel_2^2 + (1-\lambda_l) \parallel \mathbf{L^t} - \mathbf{E^l}\mathbf{E^t}^T \parallel_2^2}_{\text{trajectory embedding component}} \\
& + \underbrace{\lambda_l \parallel \mathbf{A^l} - \mathbf{E^c}\mathbf{E^a}^T \parallel_2^2 + (1-\lambda_l) \parallel \mathbf{A^t} - \mathbf{E^t}\mathbf{E^a}^T \parallel_2^2}_{\text{collaborative learning component}} \\
& + \lambda \parallel \mathbf{\Phi} \parallel_2^2 \\
& \textit{subject to}: \\
& \mathbf{\Theta} \geq 0, \mathbf{A^l} \geq 0, \textit{and } \mathbf{A^t} \geq 0,
\end{aligned} \tag{5}$$

where $\mathbf{\Phi}$ is the set of all the variables that need to estimate and λ is the parameter to prevent over-fitting.

Through the objective function (Eq. (5)) of HMRM, we know that the activity-location distribution matrix $\mathbf{A^l}$ and the activity-time distribution matrix $\mathbf{A^t}$ are shared by both the activity modeling component and the collaborative learning component; the context location embedding matrix $\mathbf{E^c}$ and the time embedding matrix $\mathbf{E^t}$ are shared by both the trajectory embedding component and the collaborative learning component. Therefore, the activity types and the trajectory embeddings we could obtain in HMRM can be mutually exchanged to take the best of the two worlds.

4.3 Parameter Inference

We now discuss how to perform the parameter inference for HMRM via collective matrix factorization. We first compute the gradient of our objective function (Eq. (5)) with respect to each variable, and then obtain the following closed-form updates by iteratively setting the gradient to zero, similar to Alternating Least Squares (ALS) matrix factorization method.

$$\begin{aligned}
\mathbf{\Theta} &= \left[\lambda_l \mathbf{U^l}\mathbf{A^l} + (1-\lambda_l)\mathbf{U^t}\mathbf{A^t}\right] \cdot \\
&\qquad \left[\lambda_l \mathbf{A^l}^T\mathbf{A^l} + (1-\lambda_l)\mathbf{A^t}^T\mathbf{A^t} + \lambda\mathbf{I}\right]^{-1} \\
\mathbf{A^l} &= \left[\lambda_l(\mathbf{U^l}^T\mathbf{\Theta} + \mathbf{E^c}\mathbf{E^a}^T)\right] \cdot \left[\lambda_l \mathbf{\Theta}^T\mathbf{\Theta} + (\lambda_l + \lambda)\mathbf{I}\right]^{-1} \\
\mathbf{A^t} &= \left[(1-\lambda_l)(\mathbf{U^t}^T\mathbf{\Theta} + \mathbf{E^t}\mathbf{E^a}^T)\right] \cdot \\
&\qquad \left[(1-\lambda_l)\mathbf{\Theta}^T\mathbf{\Theta} + (1-\lambda_l+\lambda)\mathbf{I}\right]^{-1} \\
\mathbf{E^a} &= \left[\lambda_l \mathbf{A^l}^T\mathbf{E^c} + (1-\lambda_l)\mathbf{A^t}^T\mathbf{E^t}\right] \cdot \\
&\qquad \left[\lambda_l \mathbf{E^c}^T\mathbf{E^c} + (1-\lambda_l)\mathbf{E^t}^T\mathbf{E^t} + \lambda\mathbf{I}\right]^{-1} \\
\mathbf{E^l} &= \left[\lambda_l \mathbf{L^l}\mathbf{E^c} + (1-\lambda_l)\mathbf{L^t}\mathbf{E^t}\right] \cdot \\
&\qquad \left[\lambda_l \mathbf{E^c}^T\mathbf{E^c} + (1-\lambda_l)\mathbf{E^t}^T\mathbf{E^t} + \lambda\mathbf{I}\right]^{-1} \\
\mathbf{E^c} &= \left[\lambda_l(\mathbf{L^l}^T\mathbf{E^l} + \mathbf{A^l}\mathbf{E^a})\right] \cdot \left[\lambda_l(\mathbf{E^l}^T\mathbf{E^l} + \mathbf{E^a}^T\mathbf{E^a}) + \lambda\mathbf{I}\right]^{-1} \\
\mathbf{E^t} &= \left[(1-\lambda_l)(\mathbf{L^t}^T\mathbf{E^l} + \mathbf{A^t}\mathbf{E^a})\right] \cdot \\
&\qquad \left[(1-\lambda_l)(\mathbf{E^l}^T\mathbf{E^l} + \mathbf{E^a}^T\mathbf{E^a}) + \lambda\mathbf{I}\right]^{-1},
\end{aligned} \tag{6}$$

where $\mathbf{I}$ is an identity matrix. This update does not guarantee the non-negativity of $\mathbf{\Theta}$, $\mathbf{A^l}$, and $\mathbf{A^t}$. Since our objective function is continuous, the minimum should be either at the point where the gradient is zero or on the boundary. Hence, if Eq. (6) assigns $\mathbf{\Theta}$, $\mathbf{A^l}$, and $\mathbf{A^t}$ with negative values, we can just set the negative values at zeros following [8], [34].

The learning algorithm of HMRM is depicted in Algorithm 1. Given the trajectories of all the users, we first build 1) the user-location frequency matrix $\mathbf{U^l}$ and the user-time frequency matrix $\mathbf{U^t}$ (detailed in Section 4.2.1) and 2) the location co-occurrence PMI matrix $\mathbf{L^l}$, and the location-time PMI matrix $\mathbf{L^t}$ (detailed in Section 4.2.2). Then we initialize the parameters ($\mathbf{\Theta}, \mathbf{A^l}, \mathbf{A^t}, \mathbf{E^a}, \mathbf{E^l}, \mathbf{E^c}, \mathbf{E^t}$) with the standard normal distribution. Finally, we iteratively update the parameters according to Eq. (6) until the objective value remains stable. We will evaluate whether our learning algorithm converges to a local minimum and report the running time of one iteration in the experiments (detailed in Section 6.9).

Algorithm 1. Learning Algorithm

Require: training trajectories, number of activity types K, dimensionality of embedding space M;
Ensure: ($\mathbf{\Theta}, \mathbf{A^l}, \mathbf{A^t}, \mathbf{E^a}, \mathbf{E^l}, \mathbf{E^c}, \mathbf{E^t}$);

// *construct training matrices*
1: build the user-location matrix $\mathbf{U^l}$ based on raw frequency;
2: build the user-time matrix $\mathbf{U^t}$ based on raw frequency;
3: build the location co-occurrence PMI matrix $\mathbf{L^l}$;
4: build the location-time PMI matrix $\mathbf{L^t}$;
// *train the model*
5: initialize the parameters ($\mathbf{\Theta}, \mathbf{A^l}, \mathbf{A^t}, \mathbf{E^a}, \mathbf{E^l}, \mathbf{E^c}, \mathbf{E^t}$);
6: **repeat**
7: update the parameters according to Eq. (6);
8: **until** stopping criteria is met;

5 Applications

To evaluate how well HMRM captures latent activity types and learns location embeddings, we perform both unsupervised and supervised quantitative tasks, including two location embedding evaluation tasks and two activity structure evaluation tasks.

5.1 Evaluation on Location Embeddings

Location Categorization. Besides the direct locations, recently their categories have been shown to be of important evidence for location recommendation [45]. Foursquare organizes the categories of locations with a hierarchical structure, and the top-level categories include *Arts & Entertainment*, *College & University*, *Event*, *Food*, *Nightlife Spot*, *Outdoors & Recreation*, *Professional & Other Places*, *Residence*, *Shop & Service*, and *Travel & Transport*. In our HMRM, we could learn the semantic relations among locations, and semantically related locations tend to be close in the embedding space. Therefore, we expect that locations with the same category are projected into closer vectors.

To measure the semantics of locations, we make location categorization following [46]. We define the similarity S of two locations (l_i and l_j) using the cosine similarity of their vector representations ($\mathbf{E^l}_i$ and $\mathbf{E^l}_j$),

$$S(l_i, l_j) = \frac{\mathbf{E^c}_i \cdot \mathbf{E^c}_j}{\parallel \mathbf{E^c}_i \parallel \cdot \parallel \mathbf{E^c}_j \parallel}. \tag{7}$$

For each location l_i from the test set, we use cosine similarity to find the most similar location l_j. We check the category of l_j, and if location l_j has the same category as location l_i, there is a match and location l_i is a matched location. The *match rate* of a test set is the ratio of the matched locations in the set over the size of the test set,

$$match\ rate = \frac{number\ of\ matched\ locations}{number\ of\ locations\ in\ test\ set}. \tag{8}$$

Larger the match rate is, better semantics the location representations retain.

Location Category Prediction. We also make supervised location category classification with these location embeddings. Given a location, we fetch the target location vector from $\mathbf{E^l}$ and the context location vector from $\mathbf{E^c}$, and concatenate them to build a feature vector. Then we use a classifier (e.g., SVM [5]) to predict the location's semantic category.

5.2 Evaluation on Activity Types

User Similarity. As we know, many users publish their meta information such as gender in the location-based social networks. Users with the same gender usually have similar preference on the visited locations. For example, a female usually prefers shopping, while a male is more likely to visit locations related to sport. Via HMRM, we could learn the user-activity distribution, i.e., each user can be represented as a numerical vector on the K latent activity types, reflecting user's implicit preference. Therefore, we expect that users with the same gender tend to have similar activity distributions. To validate it, given two sets of users, we measure the *average mutual similarity* of the activity distributions of those users following [30].

TABLE 2
Data Statistics

Dataset	♯Users	♯Locations	♯Check-ins
New York	7,704	40,895	988,955
Tokyo	6,233	29,585	1,362,782

We first measure the similarity between two user-activity distributions ($\mathbf{\Theta}_i$ and $\mathbf{\Theta}_j$) based on the cosine similarity,

$$S(u_i, u_j) = \frac{\mathbf{\Theta}_i \cdot \mathbf{\Theta}_j}{\parallel \mathbf{\Theta}_i \parallel \cdot \parallel \mathbf{\Theta}_j \parallel}. \tag{9}$$

Then we use the average mutual similarity to measure the similarity between activity distributions of two sets of users. The average mutual similarity between user sets $\mathcal{I}$ and $\mathcal{J}$ is defined by the following,

$$S(\mathcal{I}, \mathcal{J}) = \frac{1}{Z(\mathcal{I}, \mathcal{J})} \sum_{u_i \in \mathcal{I}} \sum_{u_j \in \mathcal{J}; u_j \neq u_i} S(u_i, u_j), \tag{10}$$

where $Z(\mathcal{I}, \mathcal{J})$ is the normalization term.

User Gender Classification. We learn the user-activity distribution $\mathbf{\Theta}$ in HMRM, and predict the gender label (male/female) of a user based on it. Specifically, we take the K-dimensional vector as basis features, the users' genders as the labels, and choose a classifier (e.g., SVM [5]) to predict the users' genders.

6 Experiments

With the proposed HMRM, we could learn the dense attribute representations, which could be used in many applications. In this section, we conduct the evaluation experiments on both unsupervised and supervised tasks introduced in Section 5 as well as the visualization of attribute representations.

6.1 Datasets and Settings

We carry out experiments on two publicly available check-in datasets collected from Foursquare from April 2012 to September 2013: one is from New York and the other is from Tokyo [36], [37]. Each check-in record contains three main properties: *user ID*, *location ID*, and *timestamp*. To make the model robust, we filter those users whose number of check-ins are fewer than 100, and those locations whose number of check-ins are fewer than 10. The statistical properties of the two datasets are shown in Table 2, where ♯Users, ♯Locations, ♯Check-ins are the number of users, locations, and check-ins, respectively.

When constructing the location co-occurrence PMI matrix $\mathbf{L^l}$, we set the size of context window b at 5, i.e., 5 preceding locations and 5 following locations are considered as context locations for a given target location. The parameters we use for the experiments are shown in Table 3. Grid search is employed to select the optimal parameters with a small but adaptive step size. For the regularization parameter, we set the default values at $\lambda = 0.001$. All the experiments run on a 3.4GHz Intel Core i5 PC with 16GB main memory.

TABLE 3
Parameters of HMRM

Parameters	Tested settings
number of activity types (K)	5, 10, 15, 20, 25, 30, 35, 40, 45, 50
dimension of embedding space (M)	10, 20, 30, 40, 50, 100, 150, 200, 250, 300
weight (λ_l)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

6.2 Baselines

As the proposed HMRM learns the activity-related distributions and the attribute embeddings, we compare it with the topic-related models and the embedding-related models.

- *CBOW:* the word2vec model which considers the local context in learning word embeddings [27].
- *Geo-Teaser:* an embedding model which considers the sequential context and the temporal factor to model the check-in sequences based on word2vec [41].
- *MC-TEM:* an embedding model which adopts the framework of CBOW and leverages multiple contexts including users, trajectories, surrounding locations and time to learn trajectory embeddings [45].
- *GTM:* we introduce four methods [1], [2], [16], [26] in Section 2.1 which discover the activity types from semantically unlabeled trajectories. The methods in [16] and [26] apply LDA to the trajectory data directly, and the methods in [2] and [1] incorporate the side features (e.g., user social graph) of trajectories into LDA to learn the activity types. As these side features do not exist in our trajectory data, we choose GTM [26] as the representative baseline for fairness.
- *UBM:* a generative user behavior method which models both location and time to understand users' latent activity types [44].
- *CLM:* a language model which learns topic structures considering both global and local contexts from the text corpus [34].

Among these baselines, CBOW, Geo-Teaser, and MC-TEM are embedding-related methods, which model the local context (e.g., surrounding location, time) and learn location embeddings, without modeling the latent activity types; HuMoR and UBM are able to learn the activity-related distributions, without learning fine-grained location embeddings; CLM do not exploit the temporal factor, as it is deigned for modeling text data.

6.3 Evaluation on Location Categorization

We randomly sample 1000 locations as the test set, and compute its match rate. To make the results more accurate, we perform the sampling process 5 times, and report the mean of 5 match rates.

6.3.1 Performance Comparison

We compare the proposed HMRM with those baselines (including CBOW [27], Geo-Teaser [41], MC-TEM [45], and CLM [34]) which could learn location embeddings, and report the results in terms of match rate on the New York and Tokyo datasets in Fig. 2a. We make the following observations:

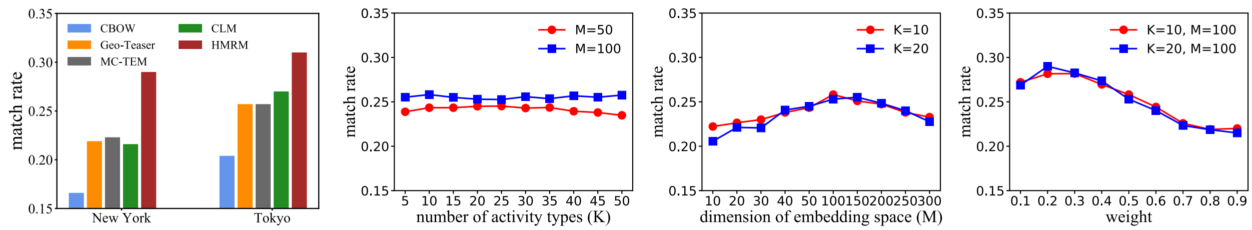
(i) HMRM is better than all the baseline on both datasets. For example, the proposed HMRM achieves 34.3 and 14.8 percent improvements over CLM, and improves by 30 and 20.6 percent over MC-TEM on the New York and Tokyo datasets, respectively.
(ii) Geo-Teaser and MC-TEM, which are based on the framework of word2vec, take the user and temporal information into account and perform better than the CBOW model on both datasets.
(iii) CLM does not exploit the temporal factor in learning location embeddings, and performs worse than the proposed HMRM, indicating the importance of temporal factor in modeling users' trajectories.

6.3.2 Parameter Sensitivity

We have three parameters to tune in HMRM: the number of latent activity types (K), the dimension of embedding space (M), and the weight λ_l. The tuning results on the New York dataset are reported in Fig. 2b. We first select the number of activity types (K) ranging from 5 to 50 with a step interval of 5 to determine the optimal, with default $M \in \{50, 100\}$ and $\lambda_l = 0.5$. The performance varies little when K increases from 5 to 50. Next, we vary the dimension of embedding space (M) from 10 to 300 with default $K \in \{10, 20\}$ and $\lambda_l = 0.5$. The performance has an obvious improvement when M increases from 10 to 100, and then starts to decline when we increase it further. Finally, we set $(K, M) \in \{(10, 100), (20, 100)\}$ and vary λ_l from 0.1 to 0.9, to validate whether it is essential to model the temporal factor in HMRM. We observe that the values of match rate reach the best when λ_l is equal to 0.2, and then drop gradually with λ_l increasing from 0.2 to 0.9, indicting that the temporal factor plays an important role in HMRM. The tuning results on the Tokyo dataset are similar, and we do not report them due to page limit.

6.4 Evaluation on Location Category Prediction

With these location embeddings, we then make the supervised location category prediction task. During the training



(a) Performance comparison. (b) Parameter tuning of HMRM for location categorization.

Fig. 2. Performance on location categorization.

TABLE 4
Performance Comparison on Location Category Prediction

Methods	New York				Tokyo			
	Recall	Precision	F1	Accuracy	Precision	Recall	F1	Accuracy
CBOW	0.21	0.20	0.20	0.29	0.22	0.18	0.20	0.32
Geo-Teaser	0.20	0.27	0.23	0.41	0.23	0.32	0.27	0.42
MC-TEM	0.21	0.27	0.24	0.42	0.24	0.31	0.27	0.43
CLM	0.44	0.48	0.46	0.53	0.46	0.51	0.48	0.52
HMRM	**0.52**[1]	**0.57**[1]	**0.54**[1]	**0.68**[1]	**0.55**[1]	**0.61**[1]	**0.58**[1]	**0.69**[1]

[1]*The improvements over baselines are statistically significant in terms of paired t-test [15] with* $p < 0.01$.

phase, we observe a certain fraction of locations and all their category labels. The task is to predict the labels for the remaining locations. Here we choose SVM [5] as the classifier, and report the average results of 10-fold cross-validation. To evaluate the classification performance, we adopt four well-known metrics, i.e., accuracy, recall, precision and average F1-measure values.

6.4.1 *Performance Comparison*

We report the results in terms of recall, precision, F1, and accuracy on the New York and Tokyo datasets in Table 4, and highlight the best results in boldface. We observe that

(i) CBOW only considers the sequential patterns in the check-in sequences to learn location embeddings, and performs the worst. Geo-Teaser and MC-TEM consider user, time, and contextual locations as the local context in learning embeddings ,and perform better than CBOW.

(ii) CLM considers the global and local surrounding locations in learning location embedding and yields decent results.

(iii) HMRM models the temporal factor and leverages the latent activity types to assist in learning location embeddings, and performs the best. For example, it achieves 28.3 percent and 32.7 percent improvements on average over CLM in terms of accuracy on the New York and Tokyo datasets.

6.4.2 *Parameter Sensitivity*

We measure the performance of HMRM with different K and M on location category prediction and report the results in terms of accuracy on the New York dataset. The experimental results on the Tokyo dataset are similar. We first set $M \in \{20, 50\}$, and report the accuracy with K from 5 to 50 in Fig. 3. With the increase of K, the values of accuracy improve gradually and reach the best when K is equal to 40. By setting $K \in \{10, 20\}$ and varying M from 10 to 300, we see that the accuracy improves when we increase M

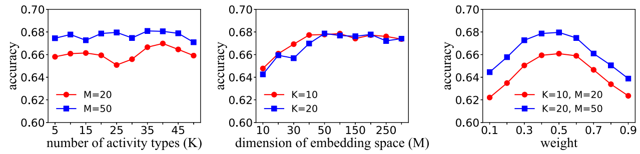


Fig. 3. Parameter tuning of HMRM for location category prediction.

from 10 to 50, and then starts to drop slightly when increasing M further. Finally, we set $(K, M) \in \{(10, 20), (20, 50)\}$ and vary λ_l from 0.1 to 0.9. The optimal performance is achieved when λ_l is equal to 0.5.

6.5 Evaluation on User Similarity

To validate whether the user representations are correctly generated (i.e., the activity distributions of users with the same gender are similar), we measure the average mutual similarity between users with the same gender labels.

6.5.1 *Performance Comparison*

We compare the proposed HMRM with those baselines (including GTM [26], UBM [44], and CLM [34]) which are able to learn user representations. The results in terms of average mutual similarity on the New York and Tokyo datasets are summarized in Fig. 5a.

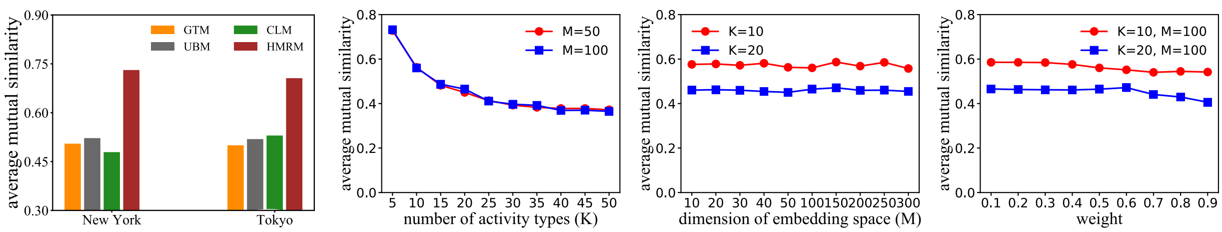
(i) GTM learns the user-activity distribution via modeling the location co-occurrences in users' trajectories, and gets decent performance; UBM additionally models the temporal factor, and performs better than GTM.

(ii) CLM models the local trajectory sequence in learning the user representations, without considering the temporal factor, and it performs better than GTM and UBM on the Tokyo dataset.

(iii) HMRM unifies the process of modeling activity types and learning trajectory embeddings, and outperforms all the baseline. For example, HMRM achieves 40 and 36 percent improvements over UBM, and improves by 52.6 and 33.2 percent over CLM on the New York and Tokyo datasets, respectively.

6.5.2 *Parameter Sensitivity*

We tune each of these parameters (i.e., the number of latent activity types (K), the dimension of embedding space (M), and the weight λ_l) with the others fixed. Fig. 5 shows the tuning results on measuring user similarity on the New York dataset. The results show that 1) the performance drops when we increase the number of activity types from 5 to 50, 2) HMRM is relatively stable when the dimension M is from 10 to 300, and 3) the average mutual similarity



(a) Performance comparison.
(b) Parameter tuning of HMRM for user similarity.

Fig. 4. Performance on measuring user similarity.

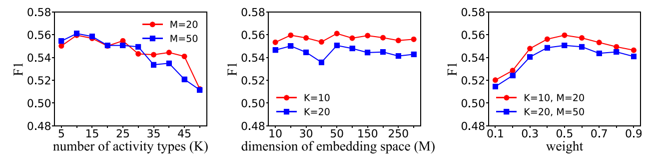


Fig. 5. Parameter tuning of HMRM for user gender classification.

declines gradually with the weight increasing from 0.1 to 0.9, indicating that the temporal factor is pretty important in the proposed HMRM, which is consistent with the conclusion derived from location categorization.

6.6 Evaluation on User Gender Classification

With the latent activity types, we choose SVM [5] to make supervised user gender classification. To evaluate the classification performance, we adopt four well-known metrics, i.e., accuracy, recall, precision and average F1-measure values. We use 10-fold cross-validation and report the average results.

To evaluate the quality of user representations, we also compare with the following methods for user gender classification.

- *BoW:* the raw BoW (Bag-of-Word) model, which assigns a vector to a user as $\vec{u} = (x_1, x_2, \ldots, x_m)$, where x_i denotes the normalized number of occurrence of the ith location, and m is the size of the collection of locations. Here, the top 1000 high-frequency locations are used as basis features.
- *BoW-F:* it considers all the locations as features.
- *BoW-T:* it uses both the locations and the check-in time distributions to build features.

6.6.1 Performance Comparison

The results on the New York and Tokyo datasets are summarized in Table 5. The best results are highlighted in boldface.

(i) BoW takes the top 1000 high-frequency locations as features, and achieves the best precisions on both New York and Tokyo datasets. However, it has the worst recalls. BoW-F takes all the locations as features, and it gets the best recalls. Except the check-in location distributions, Bow-T also considers the check-in time distributions as features, and obtains similar performances with BoW-F.

(ii) GTM learns the user-activity distribution and gets decent performance; UBM models both location and time to learn latent activities, and performs better than GTM; CLM considers the local context to help learn the latent activity types, and it performs better than GTM and UBM in some metrics.

(iii) Our HMRM models latent activity types and learns trajectory embeddings collaboratively, and considers the temporal factor in the three components. It performs the best in terms of accuracy and F1 values.

TABLE 5
Performance Comparison on User Gender Classification

Methods	New York				Tokyo			
	Recall	Precision	F1	Accuracy	Recall	Precision	F1	Accuracy
BoW	0.25	**0.61**	0.35	0.54	0.47	**0.71**	0.56	0.64
BoW-F	**0.59**	0.51	0.55	0.53	**0.66**	0.59	0.62	0.60
BoW-T	0.56	0.53	0.54	0.54	0.64	0.59	0.62	0.60
GTM	0.51	0.56	0.53	0.55	0.48	0.51	0.49	0.50
UBM	0.52	0.56	0.54	0.56	0.57	0.56	0.56	0.58
CLM	0.58	0.51	0.54	0.52	0.61	0.59	0.60	0.59
HMRM	0.57	0.56	**0.56**[1]	**0.56**[1]	0.62	0.65	**0.63**[1]	**0.65**[1]

[1]*The improvements over baselines are statistically significant in terms of paired t-test [15] with* $p < 0.01$.

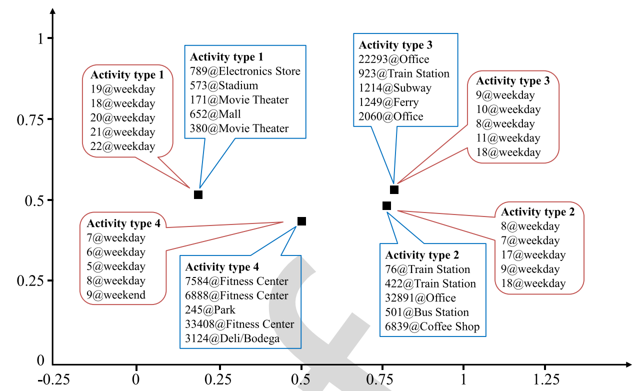


Fig. 6. Two-dimensional PCA projection of activity representations.

6.6.2 Parameter Sensitivity

We set the dimension of embedding space $M \in \{20, 50\}$, vary the number of activity types K from 5 to 50, and demonstrate the performance on user gender classification on the New York dataset. As shown in Fig. 5, the values of $F1$ improve when we increase K from 5 to 10, and then decline gradually when increasing K further. Then, we set $K \in \{10, 20\}$ and vary M from 10 to 300. HMRM remains stable with the increase of M, and performs relatively better when M is equal to 50. Finally, we evaluate the effect of λ_l, which balances the two parts in each component. By varying λ_l from 0.1 to 0.9 and setting $(K, M) \in \{(10, 20), (20, 50)\}$, we find that F1 values improve when we increase λ_l to 0.5, and then decline when we increase it further. The experimental results on the Tokyo dataset are similar, and we do not show them here.

6.7 Qualitative Analysis of Representations

6.7.1 Activity Representations

A major merit of HMRM is that it could learn the activity-location distribution, the activity-time distribution and the activity embeddings simultaneously. Those activity embeddings are of the same dimensionality as location/time embeddings. The relationships between activity embeddings and location/time embeddings are modeled in Eq. (4): the larger inner product value a location/time embedding and an activity embedding get, the more important that location/time is in the activity type. After convergence, the similarities and correlations among activity types are also captured in the embedding space. Fig. 6 shows the two-dimensional PCA projection of representations of four activity types. Each activity is annotated with its top 5 locations and time slots. Since we cannot understand the activity types according to the location IDs, we label these top 5 locations with the crawled category labels. For the time, we characterize a day at the hour scale (i.e., the numbers from 0 to 23 represent the 24 hours in a day) and distinguish

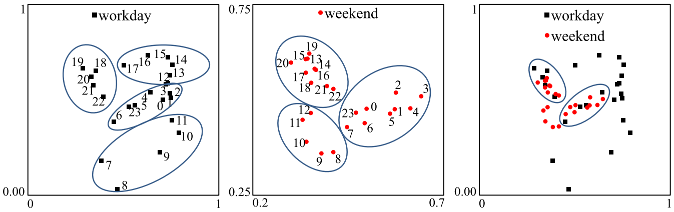
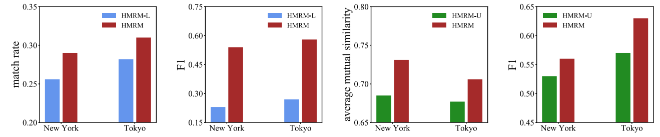


Fig. 7. Two-dimensional PCA projection of time embeddings.



(a) Location categorization. (b) Location category prediction. (c) User similarity. (d) User gender classification.

Fig. 8. Comparison results of HMRM-U, HMRM-L and our method.

weekdays from weekends. We can observe that the semantic similarities between activity types correlate with the euclidean distances between the corresponding activity embeddings. For example, activity types 2 and 3 are both about "going to work/coming off work", appearing in the morning and evening rush hours, and they are pretty close in the two-dimensional space; while activity types 1 and 2 have different semantics, and they are far from each other.

6.7.2 Time Representations

The relationships between time embeddings and location embeddings are modeled in Eq. (3) and those between time embeddings and activity embeddings are modeled in Eq. (4). Therefore, the time slots with semantically similar activity types and locations tend to be close in the embedding space. Fig. 7 shows the two-dimensional PCA projection of time embeddings. Specifically, the left two figures introduce the details of embedding on weekdays and on weekends respectively, and the right figure indicates the relationships of time embeddings between weekdays and weekends. We can observe that the hours on weekdays can be split into four groups (namely *night*: from 23:00 to 7:00, *morning*: from 7:00 to 12:00, *afternoon*: from 12:00 to 18:00, and *evening*: from 18:00 to 23:00), which are consistent with human behaviors in one day. For example, on weekdays, people usually go to work in the morning, have some activities related to "entertainment", "shopping" and "nightlife" in the evening, and sleep at home at night. Different from those of weekdays, the hours on weekends can be split into three groups: 1) The time from 7:00 to 8:00 is in the *night* group, as people usually get up later on weekends. 2) The *afternoon* and *evening* are in one group, which may be due to the fact that people's activities in the afternoon and evening are similar on weekends. Further, we see that the time embeddings in the two groups are relatively close between weekdays and weekends. On one hand, the time embeddings in the *night* are close, as the main activity type of people is "sleeping at home" no matter on weekdays and on weekends. On the other hand, the embeddings in the *evening* of weekdays are close to those in the *afternoon* and *evening* of weekends, because people usually have similar activity types in those time periods.

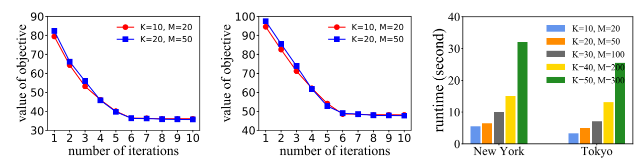
6.8 Model Analysis

To verify the effectiveness of our method, we also design several variants. 1) HMRM-U: it factorizes $\mathbf{U}^\mathbf{l}$ and $\mathbf{U}^\mathbf{t}$ with NMF and learns the latent activity types. This variant is to evaluate how trajectory embeddings assist in capturing the latent activity structures. 2) HMRM-L: it simply factorizes the two SPPMI matrices $\mathbf{L}^\mathbf{l}$ and $\mathbf{L}^\mathbf{t}$, which is equivalent to our HMRM without modeling the latent activity types. This variant is to evaluate how the latent activity types assist in learning location embeddings. We record the comparison performance with the aforementioned unsupervised and supervised tasks on the New York and Tokyo datasets in Fig. 8. From the results, we can find that the performance of HMRM is obviously higher than that of both variants. On one hand, if two locations have similar activity distributions, HMRM would adjust the two corresponding location embeddings closer to each other accordingly; therefore, these location embeddings could better retain semantics, and performs better in location categorization and location category prediction. On the other hand, HMRM considers the spatial information of location/time embeddings, and groups semantically related locations/time (which are geographically close in the embedding space) into the same activity types; hence, it captures more coherent latent activity types and generates better user-activity distributions, and outperforms HMRM-U in the tasks of user similarity and user gender classification.

6.9 Efficiency Analysis

Our learning method with Alternating Least Squares matrix factorization is an iterative algorithm. We want to know whether our model's objective achieves a stationary point fast when iteratively performing these updates. We respectively vary the number of activity types and the dimensionality of embedding space $(K, M) \in \{(10, 20), (20, 50)\}$. The values of objective function (Eq. (5)) with the number of iterations varying from 1 to 10 on both datasets are shown in Figs. 9a and 9b. Clearly, with the increase of the number of iterations, the values of objective decline gradually, and remain stable after about 10 iterations. Overall, our learning algorithm has fast convergence speed in practice.

At each iteration, our model needs to update all the parameters, including $\mathbf{\Theta}$, $\mathbf{A}^\mathbf{l}$, $\mathbf{A}^\mathbf{t}$, $\mathbf{E}^\mathbf{a}$, $\mathbf{E}^\mathbf{l}$, $\mathbf{E}^\mathbf{c}$, $\mathbf{E}^\mathbf{t}$. The sizes of these matrices determine the runtime of each iteration. Fig. 9c shows the runtime of one iteration for both datasets with different K and M. On one hand, as the New York dataset has more users and locations, its runtime is longer than that on the Tokyo dataset for the same K and M; on the other hand, the runtime increases gradually when we increase K and M. We could train HMRM offline in advance, and use the learned activity distributions and the trajectory embeddings to support real-time applications.



(a) New York. (b) Tokyo. (c) Runtime.

Fig. 9. Efficiency Performance of HMRM.

7 Conclusion and Future Work

We have proposed a Human Mobility Representation Model (HMRM) to learn dense representations of attributes including user, location, time and activity type from the semantically unlabeled trajectory data. The proposed HMRM contains the activity modeling component, the trajectory embedding component, and the collaborative learning component. The activity modeling component formulates users as a mixture of latent activity types, the trajectory embedding component projects locations and time slots into an embedding space, and the collaborative learning component establishes direct connections between attribute embeddings and activity types, and regulates distributional activity semantics accordingly. Via HMRM, the locations (or time slots) close to each other in the embedding space tend to have similar activity distributions. We evaluate the performance of HMRM on two real check-in datasets with quantitative and qualitative tasks, and experimental results show that HMRM outperforms the baselines.

Several interesting research problems exist for further exploration. First, though about 30 percent of check-in locations do not possess meaningful semantic labels, we can still try to exploit the incomplete semantic information in our model. Second, since users' activity types change over time, we can consider how to incorporate the dynamism of user activities into the model.

Acknowledgments

This work was supported in part by the Fundamental Research Funds of Shandong University, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2019BF010, the National Natural Science Foundation of China under Grant No. 61906107, 61572289, 61972069, 61836007, 61832017 and 61532018, the Open Fund of Key Laboratory of Urban Natural Resources Monitoring and Simulation, Ministry of Natural Resources, and the NSERC Discovery under Grant RGPIN-2017-05723 and Grant RGPIN-2018-06641.

References

[1] B. Alharbi, A. A. Qahtan, and X. Zhang, "Minimizing user involvement for learning human mobility patterns from location traces," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 865–871.

[2] B. Alharbi and X. Zhang, "Learning from your network of friends: A trajectory representation learning model based on online social ties," in *Proc. IEEE 16th Int. Conf. Data Mining*, 2016, pp. 781–786.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[4] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, "Content-aware hierarchical point-of-interest embedding model for successive poi recommendation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3301–3307.

[5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[6] M. Chen, X. Yu, and Y. Liu, "Mining moving patterns for predicting next location," *Inf. Syst.*, vol. 54, pp. 156–168, 2015.

[7] M. Chen, X. Yu, and Y. Liu, "MPE: A mobility pattern embedding model for predicting next locations," *World Wide Web J.*, vol. 22, no. 6, pp. 2901–2920, 2019.

[8] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. S. Kankanhalli, "MMALFM: Explainable recommendation by leveraging reviews and images," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 16:1–16:28, 2019.

[9] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 137–143.

[10] Z. Fan, X. Song, and R. Shibasaki, "CitySpectrum: A non-negative tensor factorization approach," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 213–223.

[11] S. Feng, G. Cong, B. An, and Y. M. Chee, "POI2Vec: Geographical latent representation for predicting future visitors," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 102–108.

[12] L. Gong, X. Liu, L. Wu, and Y. Liu, "Inferring trip purposes and uncovering travel patterns from taxi trajectory data," *Cartography Geographic Inf. Sci.*, vol. 43, no. 2, pp. 103–114, 2016.

[13] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, 1999, pp. 289–296.

[14] L. Huang, Q. Li, and Y. Yue, "Activity identification from GPS trajectories using spatial temporal pois' attractiveness," in *Proc. 2nd ACM SIGSPATIAL Int. Workshop Location Based Social Netw.*, 2010, pp. 27–30.

[15] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1993, pp. 329–338.

[16] K. Joseph, C. H. Tan, and K. M. Carley, "Beyond local, categories and friends: Clustering foursquare users with latent topics," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 919–926.

[17] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2341–2347.

[18] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: Joint modeling of user's activity area and interests for location recommendation," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 375–384.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[20] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.

[21] Y. Li, H. Chen, L. Wang, and Q. Xiao, "POI representation learning by a hybrid model," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage.*, 2019, pp. 485–490.

[22] D. Lian *et al.*, "Scalable content-aware collaborative filtering for location recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1122–1135, Jun. 2018.

[23] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.

[24] X. Liu, Y. Liu, and X. Li, "Exploring the context of locations for personalized location recommendations," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1188–1194.

[25] Y. Liu, C. Liu, X. Lu, M. Teng, H. Zhu, and H. Xiong, "Point-of-interest demand modeling with human mobility patterns," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 947–955.

[26] X. Long, L. Jin, and J. Joshi, "Exploring trajectory-driven local geographic topics in foursquare," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 927–934.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[28] J. Pang and Y. Zhang, "DeepCity: A feature learning framework for mining location check-ins," in *Proc. 11th Int. AAAI Conf. Web Social Media (ICWSM)*, 2017, pp. 652–655.

[29] Y. Qiao, X. Luo, C. Li, H. Tian, and J. Ma, "Heterogeneous graph-based joint representation learning for users and pois in location-based social network," *ACM Trans. Knowl. Discovery Data*, vol. 57, no. 2, 2020, Art. no. 102151.

[30] Y. Takahiro, T. Kota, S. Toru, S. Yoshihide, and V. U. Satish, "City2city: Translating place representations across cities," in *Proc. 27th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2019, pp. 412–415.

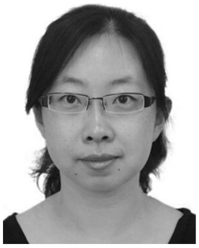
[31] K. Takeuchi, R. Tomioka, K. Ishiguro, A. Kimura, and H. Sawada, "Non-negative multiple tensor factorization," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 1199–1204.

[32] Y. Wang, M. Chen, X. Yu, and Y. Liu, "LCE: A location category embedding model for predicting the category labels of POIs," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 710–720.

[33] K. Xie, K. Deng, and X. Zhou, "From trajectories to activities: A spatio-temporal join approach," in *Proc. Int. Workshop Location Based Soc. Netw.*, 2009, pp. 25–32.
[34] G. Xun, Y. Li, J. Gao, and A. Zhang, "Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 535–543.
[35] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, 2017, Art. no. 36.
[36] 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, 2016, Art. no. 30.
[37] D. Yang, D. Zhang, B. Qu, and P. Cudre-Mauroux , "PrivCheck: Privacy-preserving check-in data publishing for personalized location based services," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 545–556.
[38] J. Yang and C. Eickhoff, "Unsupervised learning of parsimonious general-purpose embeddings for user and location modeling," *ACM Trans. Inf. Syst.*, vol. 36, no. 3, 2018, Art. no. 32.
[39] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2411–2414.
[40] H. Ying *et al.*, "Time-aware metric embedding with asymmetric projection for successive poi recommendation," *World Wide Web J.*, vol. 22, pp. 1–16, 2018.
[41] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 153–162.
[42] W. X. Zhao, F. Fan, J.-R. Wen, and E. Y. Chang, "Joint representation learning for location-based social networks with multi-grained sequential contexts," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 2, 2018, Art. no. 22.
[43] W. X. Zhao, N. Zhou, A. Sun, J.-R. Wen, J. Han, and E. Y. Chang, "A time-aware trajectory embedding model for next-location recommendation," *Knowl. Inf. Syst.*, vol. 56, pp. 559–579, 2018.
[44] J. Zheng and L. M. Ni, "An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 153–162.
[45] N. Zhou, W. X. Zhao, X. Zhang, J.-R. Wen, and S. Wang, "A general multi-context embedding model for mining human trajectory data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 1945–1958, Aug. 2016.
[46] Y. Zhou and Y. Huang, "Deepmove: Learning place representations through large scale movement data," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 2403–2412.

Meng Chen (Member, IEEE) received the PhD degree in computer science and technology from Shandong University, China, in in 2016. He worked as a postdoctoral fellow with the School of Information Technology, York University, Canada, from 2016 to 2018. He is currently an assistant professor with the School of Software, Shandong University, China. His research interest include the area of trajectory data mining and traffic management.

Yan Zhao received the master's degree in geographic information system from the University of Chinese Academy of Sciences, in 2015. She is currently working toward the PhD degree at Soochow University. Her research interests include spatial database and trajectory computing.



Yang Liu (Member, IEEE) received the PhD degree in computer science and engineering from York University, Canada, in 2008. She is currently an associate professor with the Department of Physics and Computer Science, Wilfrid Laurier University, Canada. Her main areas of research are data mining and information retrieval.

Xiaohui Yu (Member, IEEE) received the PhD degree in computer science from the University of Toronto, Canada, in 2006. He is an associate professor with the School of Information Technology, York University, Toronto. His research interests include the areas of database systems and data mining.

Kai Zheng (Senior Member, IEEE) received the PhD degree in computer science from the University of Queensland, in 2012. He is a professor of computer science with the University of Electronic Science and Technology of China. He has been working in the area of spatio-temporal databases, uncertain databases, social-media analysis, in-memory computing and blockchain technologies. He has published more with 100 papers in prestigious journals and conferences in data management field such as SIGMOD, ICDE, VLDB Journal, ACM Transactions and the *IEEE Transactions*.