Learning Graph-based POI Embedding for Location-based Recommendation

Min Xie† Hongzhi Yin‡* Hao Wang† Fanjiang Xu† Weitong Chen‡ Sen Wang⋄

† State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China

‡The University of Queensland, School of Information Technology and Electrical Engineering, Australia

⋄School of Information and Communication Technology, Griffith University, Australia

‡{h.yin1, w.chen9}@uq.edu.au †{xiemin2014, wanghao, fanjiang}@iscas.ac.cn

⋄sen.wang1982@gmail.com

ABSTRACT

With the rapid prevalence of smart mobile devices and the dramatic proliferation of location-based social networks (LBSNs), location-based recommendation has become an important means to help people discover attractive and interesting points of interest (POIs). However, extreme sparsity of user-POI matrix and cold-start issue create severe challenges, causing CF-based methods to degrade significantly in their recommendation performance. Moreover, location-based recommendation requires spatiotemporal context awareness and dynamic tracking of the user's latest preferences in a real-time manner.

To address these challenges, we stand on recent advances in embedding learning techniques and propose a generic graph-based embedding model, called GE, in this paper. GE jointly captures the sequential effect, geographical influence, temporal cyclic effect and semantic effect in a unified way by embedding the four corresponding relational graphs (POI-POI, POI-Region, POI-Time and POI-Word) into a shared low dimensional space. Then, to support real-time recommendation, we develop a novel time-decay method to dynamically compute the user's latest preferences based on the embedding of his/her checked-in POIs learnt in the latent space. We conduct extensive experiments to evaluate the performance of our model on two real large-scale datasets, and the experimental results show its superiority over other competitors, especially in recommending cold-start POIs. Besides, we study the contribution of each factor to improve location-based recommendation, and find that both sequential effect and temporal cyclic effect play more important roles than geographical influence and semantic effect.

1. INTRODUCTION

With the rapid development of Web 2.0, location acquisition and wireless communication technologies, a sufficient number of location-based social networks (LBSNs) have emerged in recent years, such as Foursquare, Facebook Places, Gowalla and Loopt, where users can check in at point-of-interests (POIs), e.g., stores, restaurants, sightseeing sites, and share life experiences in the physical world via mobile devices promptly. It is crucial to utilize user check-in data to make personalized recommendation in a real-time manner, which helps users know new POIs and explore new regions (e.g., cities), facilitate advertisers to launch mobile advertisements to targeted users.

Unlike traditional desktop recommendation systems delivering "digital" information, e.g., movie recommendation, music recommendation etc., location-based recommendation systems typically involve mobile users and "physical" entities (such as sightseeing sites), which suffers from more challenges. 1. **Data Sparsity**. To know and rate a POI, a user has to physically visit that POI, thus the cost is more expensive than rating a movie online. Even if a user makes the effort to visit the POI, he often does not check in due to privacy or safety concerns. Therefore, users' check-in data generated in LBSNs is much sparser than their generated rating data for movies and music. This issue plagues most of the existing collaborative filtering recommender systems. 2. **Context Awareness**. Location-based recommendation requires considering not only personal preferences, but also the spatiotemporal context [21], as a user tends to have different choices and needs at different time and places. 3. **Cold Start** is a critical problem in the domain of location-based recommendation, as many new POIs (e.g., business) are emerging every day, especially in the fast-developing countries. 4. **Dynamic of Personal Preferences** is the last, but the most critical challenge. As analyzed in [22], users' preferences are changing with the time going on. For instance, users will naturally be interested in visiting parenting-related POIs (e.g., the playground and amusement park) after they have a baby, and probably ignore their other interests. Accurately capturing this change has been proved to be commercially very valuable since it indicates visiting and purchasing intents. Thus, location-based recommendation requires provides satisfactory recommendations promptly based on users' latest preferences and their current spatiotemporal context, which requires producing recommendation results in a **real-time manner**.

To deal with all the above challenges and improve location-based recommendation, we exploit and integrate the following factors in a unified way.

*This author is the corresponding author.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983711

- **Sequential Effect**. It has been observed by research on human mobility that, human movement exhibits sequential patterns [4, 29], which means the transition probabilities from one checked-in POI to other POIs is a nonuniform distribution, e.g., users often sequentially check-in at airports and hotels.
- **Geographical Influence**. Many recent studies show that people tend to visit nearby POIs or explore POIs near the ones that they have visited before [20].
- **Temporal Cyclic Effect**. As suggested in [7, 28], users' mobility behaviors in the physical world exhibit strong temporal cyclic patterns, and the daily pattern (hours of the day) is one of the most fundamental patterns. For example, a user may regularly arrive to the office around 9:00 am, have dinner at a restaurant at 12:00 am, and watch movies at night around 10:00 pm. So he is more likely to go to a restaurant rather than a bar at lunch time.
- **Semantic Effect**. A recent analysis of the Whrrl dataset shows that the check-in activities of users exhibit a strong semantic regularity [19]. In other words, the contents of POIs checked-in by the same user tend to semantically similar.

While there are many recent studies that exploit one or two of the above factors to improve location-based recommendation effectiveness, they lack an integrated analysis of their joint effect to address all the above four challenges in a unified and generic way. Specifically, to overcome the data sparsity issue, most prior work of location-based recommendation focused on exploiting the geographical influence [10, 3, 20] and temporal cyclic effect [7, 28] to provide spatial or/and temporal context-aware recommendation. To further address the cold-start problem, some recent work explored the semantic effect and integrated the content information of POIs [25, 17]. Most recently, Yin et al. [27] developed a probabilistic generative model for joint modeling of geographical influence, temporal cyclic effect and semantic effect. However, their method is not a generic method and cannot be straightforwardly extended to incorporate other factors such as sequential influence.

On the other hand, almost all existing location-based recommendation methods are incapable of supporting real-time recommendation principally, and they would suffer from the following two drawbacks: 1) Delay on model updates caused by the expensive time cost of re-running the recommender model; and 2) Disability to track changing user preferences due to the fact that latest check-in records used for updating recommendation models are often overwhelmed by the large data of the past. Accurately capturing the change of user preferences in a real-time manner is very helpful for location-based recommendation. As each check-in provides valuable information about the user's preferences, recommender model must respond immediately to new check-in information. Although several Markov chain-based recommender models [4, 29] have been recently developed to predict next POIs based on the user's recent check-ins by mining the sequential patterns, they do not consider other three important factors, failing to overcome the challenges of data sparsity and cold start.

Recently, methods of embedding items in a low-dimension Euclidean space have been widely adopted in a variety of fields, including natural language processing, text mining and music information retrieval. Tang et al. [15] predicted text embeddings based on heterogeneous text networks which showed great potential in document classification. Chen et al. [1] proposed a Logistic Markov embedding (LME) model to map each song to one point (or multiple points) in a latent Euclidean space for playlists generating, which also verifies the effectiveness of embedding methods.

In this paper, we stand on the recent advances in embedding learning techniques and propose a graph-based embedding method called GE to encode the above four factors in a low-dimension latent space to effectively address the issues of data sparsity, cold start and context awareness in a unified manner. Specifically, we use a bipartite graph model to represent (or capture) the relationships between POIs and POIs (sequential effect), POIs and geographical regions (geographical influence), POIs and time slots (temporal cyclic effect), POIs and content words (semantic effect), respectively. Then, we embed the four heterogenous information graphs into a shared low-dimension space, in which each vertex, i.e., each POI (including cold-start ones), time slot, geographical region and content word, is represented as a low-dimensional vector. Note that our proposed GE model is a generic flexible model that can be easily extended to incorporation of other factors, not limited to the above four.

To keep track of the dynamics of user preferences, we adopt a time-decay manner to compute the user's latest preferences based on the embeddings of the user's checked-in POIs and associated timestamps, i.e., if a POI is visited by the user more recently, it will be more important and assigned with a higher weight. Thus, just like the classic item-based collaborative filtering method [11], our proposed dynamic user preference modeling method has the nice properties of making fast response to new check-in information and scaling to massive datasets without retraining model. To support real-time & context-aware recommendation, the final recommendations are made based on the embeddings of the user's latest preferences and the spatiotemporal context in the shared latent space.

The primary contributions of our research are summarized as follows.

- To the best of our knowledge, we are the first to investigate the joint effect of sequential effect, geographical influence, temporal cyclic effect and semantic effect to address the challenges of data sparsity, cold start, context awareness and dynamic user preferences in a unified way.
- We develop a graph-based embedding model to learn the representations of POIs, time slots, geographical regions and content words in a shared low-dimension space. Then, to support real-time recommendation, we propose a novel method for dynamic user preferences modeling based on the learnt embedding of POIs.
- We conduct extensive experiments to evaluate the performance of our recommender method on two real large-scale datasets. The results show the superiority of our proposals in location-based recommendation by comparing with the state-of-the-art techniques.

The remainder of the paper is organized as follows. Section 2 details our proposed graph-based embedding methods. We report the experimental results in Section 3. Section 4 reviews the related work and Section 5 concludes the paper.

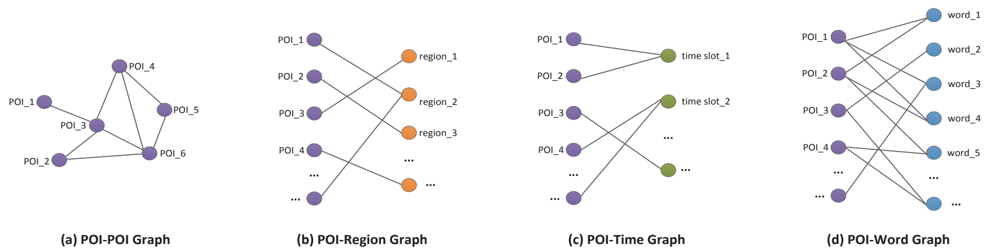


Figure 1: Illustration of encoding sequential effect, geographical influence, temporal cyclic effect and semantic effect into a low dimensional metric utilize graph-based method. POI-Region graph, POI-Time graph and POI-Word graph are all bipartite graphs, while POI-POI graph is a general graph which can also be treated as a bipartite graph when one POI is on one side and others are on the opposite side. Diverse factors can be connected through POIs.

Variable	Interpretation
$\mathcal{U}, \mathcal{V}$	the set of users and POIs
$\mathcal{R}$, $\mathcal{T}$, $\mathcal{W}$	the set of regions, time slots and words
l_v, $\mathcal{W}_v$,	POI v's location and content word set
τ, t,	the timestamp of check-in and its discretized time slot
$\mathcal{D}_u$	the profile of user u
$\mathbb{R}^d$	d dimensional latent space
$\vec{u}_\tau$	time-aware user preference embedding
$\vec{v}$, $\vec{r}$, $\vec{t}$, $\vec{w}$	embeddings of POI v, region r, time slot t and word w, respectively
$\triangle T$	the time interval

Table 1: Notations used in this paper.

2. GRAPH-BASED EMBEDDING MODEL

In this section, we first introduce the key data structures and definitions used in this paper, and then present our proposed graph-based embedding (GE) model and the optimization method.

2.1 Problem Formulation

For ease of presentation, we define the key data structures and notations used in this paper. Table 1 also lists them.

Definition 1. **(*POI*)** A POI is defined as a uniquely identified specific site (e.g., a restaurant or a cinema). In our model, a POI has three attributes: identifier, geographical location and content. We use v to represent a POI identifier and l_v to denote its corresponding geographical attribute in terms of longitude and latitude coordinates. Besides, there is textual semantic information associated with a POI, such as the category and tag words. We use the notation $\mathcal{W}_v$ to denote the set of words describing POI v.

Definition 2. **(*Check-in Activity*)** A check-in activity is made of a five tuple (u, v, l_v, τ, $\mathcal{W}_v$) that means user u visits POI v at time τ.

Definition 3. **(*User Profile*)** For each user u, we create a user profile $\mathcal{D}_u$, which is a set of check-in activities associated with u and sorted by timestamp. The dataset $\mathcal{D}$ used in our model includes all user profiles, i.e.,$\mathcal{D} = \{\mathcal{D}_u : u \in \mathcal{U}\}$.

Definition 4. **(*POI-POI Graph*) POI-POI** graph, denoted as $G_{vv} = (\mathcal{V} \cup \mathcal{V}, \mathcal{E}_{vv})$, captures the check-in sequence of POIs in a user profile $\mathcal{D}_u$. $\mathcal{V}$ is a set of POIs and $\mathcal{E}_{vv}$ is the set of edges between POIs. Given a time interval $\triangle T$, for each check-in pair $\{(v_i, \tau_i), (v_j, \tau_j)\}$ in a user profile $\mathcal{D}_u$, if $0 < \tau_j - \tau_i \leq \triangle T$, there will be an edge e_{ij} from v_i to v_j. The weight w_{ij} of edge e_{ij} is defined as the number of times that v_j is checked in after v_i in the whole dataset $\mathcal{D}$ within a predefined time interval $\triangle T$.

The POI-POI graph captures the sequential patterns of check-in POIs. Intuitively, if v_i and v_j are often checked in sequentially, their correlation will be larger: if a user has visited v_i, he has a high probability to visit v_j next.

To capture the geographical influence, temporal cyclic effect and semantic effect, the bipartite graphs POI-Region, POI-Time and POI-Word are defined as below. Fig. 1 gives an illustration of our graph-based model. As vertices are discrete in graphs, we need first transform the continuous values in the check-in records (such as location coordinates and timestamps) into discrete ones. Specifically, we divide the whole geographical space into a set of regions $\mathcal{R}$ according to administrative divisions such as cities or suburbs. All check-in timestamps are divided into a set of time slots $\mathcal{T}$ to capture the temporal cyclic patterns, such as 24 hours for the daily patterns or 7 days for the weekly patterns. We will study the impact of different division methods of time in Section 3.4.3.

Definition 5. **(*POI-Region Graph*) POI-Region** graph, denoted as $G_{vr} = (\mathcal{V} \cup \mathcal{R}, \mathcal{E}_{vr})$, is a bipartite graph where $\mathcal{R}$ is a set of regions and $\mathcal{V}$ is a set of POIs. $\mathcal{E}_{vr}$ is the set of edges between POIs and regions. If POI v_i is located in region r_j, there will be an edge e_{ij} between them, otherwise none. The weight w_{ij} is set to 1 when the edge e_{ij} exists.

Definition 6. **(*POI-Time Graph*) POI-Time** graph, denoted as $G_{vt} = (\mathcal{V} \cup \mathcal{T}, \mathcal{E}_{vt})$, is a bipartite graph where $\mathcal{T}$ is a set of time slots and $\mathcal{V}$ is a set of POIs. $\mathcal{E}_{vt}$ is the set of edges between POIs and time slots. The weight w_{ij} of the edge between POI v_i and time slot t_j is defined as the frequency of POI v_i checked in at time slot t_j.

Definition 7. **(*POI-Word Graph*) POI-Word** graph, denoted as $G_{vw} = (\mathcal{V} \cup \mathcal{W}, \mathcal{E}_{vw})$, is a bipartite graph where $\mathcal{W}$ represents the vocabulary set. $\mathcal{E}_{vw}$ is the set of edges between POIs and words. If POI $w_i \in \mathcal{D}_{v_j}$, there will be an edge e_{ij} between word w_i and POI v_j, otherwise none. As $\mathcal{D}_{v_j}$ is a bag of words, we use the standard $tf.idf$ to compute the edge weight w_{ij}.

The three types of graphs above can well capture the geographical influence, temporal cyclic effect and semantic effect, respectively. Take POI-Time graph as an example, we can interpret it as following: if a POI v_i is often visited in time slot t_j, the edge weight w_{ij} is large. Thus, given a target user u at the temporal context t_j, he/she is most likely to visit POI v_i. It is worth mentioning that our graph-based embedding learning model can be straightforwardly generalized to integrate other types of graphs such as POI-image graph which captures the visual effect. In this work, we only consider four types of graphs : POI-POI, POI-Region, POI-Time, POI-Word. Our goal is to embed the above four graphs into a shared low dimensional space $\mathbb{R}^d$ where d is the dimension. Then, we can get the vector representations of POIs, regions, time slots and words, i.e., $\vec{v}$, $\vec{r}$, $\vec{t}$ and $\vec{w}$.

Finally, we formally define the problem investigated in our work. Given a dataset $\mathcal{D}$ as the union of a collection of user profiles, we aim to provide location-based recommendations stated as follows.

PROBLEM 1. ***(Location-based Recommendation)*** *Given a user activity dataset $\mathcal{D}$ and a querying user u with his/her current location l and time τ (that is, the query is $q = (u, l, \tau)$), our goal is to recommend top-k POIs that u would be interested in. To fit the time τ and location l into our graphs, we first map them into the corresponding region r and time slot t.*

2.2 Model Description and Optimization

In this section, we first propose a graph-based embedding learning approach using bipartite graph and its optimization method, and then present the integrated learning of POI embedding, region embedding, time slot embedding and word embedding in the latent space. Last, we show how to track and represent the dynamic user preferences promptly.

2.2.1 Bipartite Graph Embedding

Inspired by the *LINE* model [16] which learns the embedding of large-scale homogeneous information networks, i.e., networks with the same types of nodes, we extend it to learn embeddings of heterogeneous graph nodes.

Given a bipartite graph $G = (\mathcal{V}_A \cup \mathcal{V}_B,\ \mathcal{E})$, where $\mathcal{V}_A$ and $\mathcal{V}_B$ are two disjoint sets of vertices of different types, and $\mathcal{E}$ is the set of edges between them. We first define the conditional probability of vertex v_j in set $\mathcal{V}_B$ generated by vertex v_i in set $\mathcal{V}_A$ as:

$$p(v_j|v_i) = \frac{\exp(\vec{v}_j^{\mathrm{T}} \cdot \vec{v}_i)}{\sum_{v_k \in \mathcal{V}_B} \exp(\vec{v}_k^{\mathrm{T}} \cdot \vec{v}_i)} \tag{1}$$

where $\vec{v}_i$ is the embedding vector of vertex v_i in $\mathcal{V}_A$, and $\vec{v}_j$ is the embedding vector of vertex v_j in $\mathcal{V}_B$, Eq. (1) defines a conditional distribution $p(\cdot|v_i)$ over all the vertices in the set $\mathcal{V}_B$. To preserve the weight w_{ij} on edge e_{ij}, we make the conditional distribution $p(\cdot|v_i)$ close to its empirical distribution $\hat{p}(\cdot|v_i)$, which can be defined as $\hat{p}(v_j|v_i) = \frac{w_{ij}}{deg_i}$. Then, we minimize the following objective function:

$$O = \sum_{v_i \in \mathcal{V}_A} \lambda_i d(\hat{p}(\cdot|v_i), p(\cdot|v_i)) \tag{2}$$

where $d(\cdot,\cdot)$ is the KL-divergence between two distributions, λ_i is the importance of vertex v_i in the graph, which can be set as the degree $deg_i = \sum_j w_{ij}$. Omitting some constants, the objective function Eq. (2) can be calculated as:

$$O = -\sum_{e_{ij} \in \mathcal{E}} w_{ij} \log p(v_j|v_i) \tag{3}$$

By learning $\{\vec{v}_i\}_{i=1...|\mathcal{V}_A|}$ and $\{\vec{v}_j\}_{j=1...|\mathcal{V}_B|}$ that minimize Eq. (3), we are able to represent different types of vertices with a d dimensional embedding $\vec{v}_i$ and $\vec{v}_j$ in metric $\mathbb{R}^d$.

2.2.2 Optimization

Optimizing objective function Eq. (3) is computationally expensive, as calculating the conditional probability $p(\cdot|v_i)$ need to sum over the entire set of vertices. To address this problem, we sample multiple negative edges according to some noisy distribution for each edge e_{ij} following the negative sampling approach proposed in [13]. For each edge e_{ij}, it specifies the following objective function:

$$\log \sigma(\vec{v}_j^{\mathrm{T}} \cdot \vec{v}_i) + \sum_{n=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-\vec{v}_n^{\mathrm{T}} \cdot \vec{v}_i)] \tag{4}$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function, K is the number of negative edges. We set $K = 5$, $P_n(v) \propto d_v^{3/4}$ from the empirical setting of [13], d_v is the out-degree of vertex v. Then we adopt the asynchronous stochastic gradient algorithm (ASGD) [14] for Eq. (4). If an edge e_{ij} is sampled, the gradient w.r.t. the embedding vector $\vec{v}_i$ of vertex v_i will be calculated as:

$$\frac{\partial O}{\partial \vec{v}_i} = w_{ij} \cdot \frac{\partial \log p(v_j|v_i)}{\partial \vec{v}_i} \tag{5}$$

However, when the weights of edges have a high variance there will be a problem, because it is very hard to find a good learning rate. If we select a large learning rate according to the edges with small weights, the gradients on edges with large weights will explode, while the gradients will become too small if we select the learning rate according to the edges with large weights. To overcome this dilemma, we follow the edge sampling approach used in [16]. Let $W = (w_1, w_2, ..., w_E)$ denote the ranking sequence of edge weights where $E = |\mathcal{E}|$. First, we calculate the sum of the weights $w_{sum} = \sum_{i=1}^{E} w_i$. Then, sample a value within $[0, w_{sum}]$ to see which interval $[\sum_{j=0}^{i-1} w_j, \sum_{j=0}^{i} w_j)$ the random value falls into. In the latter procedure, we use alias table method [9] to draw a sample, thus reduce the sampling complexity to $O(1)$. Moreover, optimization with negative sampling takes $O(\eta \times (K + 1))$ time cost, where K is the number of negative samples and η is the time taking for one sampling. Thus, the entire step takes $O(\eta \times K)$ time. In fact, the number of steps used for optimization is usually proportional to the number of edges E. Therefore, the overall time complexity of optimization is $O(\eta \times K \times E)$, while η, K are all constants. The proposed edge sampling method is very efficient since it is linear to the number of edges E, and does not depend on the number of vertices V ($V = |\mathcal{V}|$).

Now we can efficiently learn the embeddings of different types of vertices in a heterogeneous bipartite graph, to learn various graph embeddings simultaneously, let us introduce the integrated embedding learning method.

2.2.3 Joint Embedding Learning

There are four bipartite graphs: POI-POI, POI-Region, POI-Time and POI-Word graphs to be embed integrated in our GE model. An intuitive approach is to collectively embed the four bipartite graphs by minimizing the sum of all objective functions as following:

$$O = O_{vv} + O_{vr} + O_{vt} + O_{vw} \tag{6}$$

where

$$O_{vv} = -\sum_{e_{ij} \in \mathcal{E}_{vv}} w_{ij} \log p(v_i|v_j) \tag{7}$$

Algorithm 1: Joint training

Data: G_{vv}, G_{vr}, G_{vt}, G_{vw}, number of samples N, number of negative samples K

Result: POI embeddings $\vec{v}$, region embeddings $\vec{r}$, time slot embeddings $\vec{t}$, word embeddings $\vec{w}$

while $iter \leq N$ **do**

- sample an edge from $\mathcal{E}_{vv}$ and draw K negative edges, and update POI embeddings;
- sample an edge from $\mathcal{E}_{vr}$ and draw K negative edges, and update POI embeddings and region embeddings;
- sample an edge from $\mathcal{E}_{vt}$ and draw K negative edges, and update POI embeddings and time slot embeddings;
- sample an edge from $\mathcal{E}_{vw}$ and draw K negative edges, and update POI embeddings and word embeddings;

end

$$O_{vr} = -\sum_{e_{ij} \in \mathcal{E}_{vr}} w_{ij} \log p(v_i | r_j) \tag{8}$$

$$O_{vt} = -\sum_{e_{ij} \in \mathcal{E}_{vt}} w_{ij} \log p(v_i | t_j) \tag{9}$$

$$O_{vw} = -\sum_{e_{ij} \in \mathcal{E}_{vw}} w_{ij} \log p(v_i | w_j) \tag{10}$$

The objective function (6) can be optimized by training all types of graphs simultaneously by merging all the edges in the four sets $\mathcal{E}_{vv}$, $\mathcal{E}_{vr}$, $\mathcal{E}_{vt}$, $\mathcal{E}_{vw}$ together, and then deploy edge sampling, which samples an edge for model updating in each step, with the sampling probability proportional to its weight. However, the graphs are heterogeneous in our model, the weights of the edges between different graphs are not comparable to each other. A more reasonable solution is to alternatively sample from the four sets of edges respectively which called *joint training*. We summarize the detailed training algorithm in Algorithm 1.

Hence, we have learnt the embeddings of POIs, regions, time slots and words integrated, then we will introduce how to represent users in the shared latent space.

2.2.4 Dynamic User Preference Modeling

General recommender models (e.g. latent factor models) achieves the dynamic update of user preferences via re-training the model or applying the online learning techniques, which is very time-consuming. We aim to propose an efficient approach that tracks the dynamic of user preferences in a linear time complexity. To achieve this, we map dynamic user preferences to the same latent space $\mathbb{R}^d$ as POIs, and utilize the learnt POI embeddings to represent the embedding of dynamic user preferences. More precisely, we assume that an individual's preferences at time τ can be represented by the collection of POIs he has visited before τ. Note that, the check-ins in $\mathcal{D}_u$ are ranked according to their check-in timestamps in an increasing order. Therefore, we can learn the embedding $\vec{u}_\tau$, i.e., u's preferences at time τ, by utilizing the vectors of POIs he has visited before τ in the form of exponential decay. That is, if a user u has checked in a set of POIs before time τ, his/her preferences at time τ can be computed as:

$$\vec{u}_\tau = \sum_{(u, v_i, \tau_i) \in \mathcal{D}_u \cap (\tau_i < \tau)} exp^{-(\tau - \tau_i)} \cdot \vec{v}_i \tag{11}$$

where $\vec{v}_i$ is the embedding of POI v_i, (u, v_i, τ_i) is u's check-in record in $\mathcal{D}_u$ before time τ; the later the POI is visited, the bigger the exponential is. In this way, we can dynamically track the user's preferences in an efficient and scalable way. Rather than directly learning latent factors for users, we compute the vector representations of users based on the embedding of their checked-in POIs. This method allows us to update user preferences without retraining GE model once the embedding of POIs has been learnt.

2.3 Recommendation Using GE

Once we have learnt the embeddings of POIs, regions, time slots and words, given a query user u with the query time τ and location l, i.e., $q = (u, \tau, l)$, we first project time τ and location l into time slot t and region r, and then select top-k POIs with the highest score that u has not visited before. More precisely, given a query $q = (u, \tau, l)$, for each POI v which has not been visited by u, we compute its ranking score as in Eq. (12), and then select the k ones with the highest ranking scores as recommendations.

$$S(q, v) = \vec{u}_\tau^{\mathrm{T}} \cdot \vec{v} + \vec{r}^{\mathrm{T}} \cdot \vec{v} + \vec{t}^{\mathrm{T}} \cdot \vec{v} \tag{12}$$

where $\vec{u}_\tau$ is the representation of u's latest preferences, which can be computed in Eq. (11), and $\vec{v}$ is the embedding of POI v, $\vec{r}$ is the embedding of region r where query location l is located in, $\vec{t}$ is the embedding of time slot t where query time τ belongs to. The learnt POI embedding $\vec{v}$ also automatically capture the semantic content information of item v through the POI-Word graph, as our GE model jointly learns the embedding of multiple relational networks in the same latent space.

The above equation encodes the joint effect of dynamic user preferences, sequential effect and semantic effect ($\vec{u}_\tau^{\mathrm{T}} \cdot \vec{v}$), the geographical influence ($\vec{r}^{\mathrm{T}} \cdot \vec{v}$) and temporal cyclic effect ($\vec{t}^{\mathrm{T}} \cdot \vec{v}$)in a unified way. As for the cold start POIs, our GE model can still learn their representations in the latent space based on the POI-Region and POI-Word graphs. Thus, both cold-start and normal POIs can be recommended together by the same ranking function - Eq. (12), which distinguishes from other existing recommender models [2, 27, 17] that use different functions to compute the scores for cold-start and normal POIs, separately.

3. EXPERIMENTS

In this section, we first describe the settings of experiments and then demonstrate the experimental results.

3.1 Experimental Settings

3.1.1 Data Sets

Our experiments are performed on two real large-scale LBSNs datasets: Foursquare and Gowalla. The basic statistics of them are shown in Table 2. The two real datasets are publicly available[1].

Foursquare. Foursquare is one of the most popular online LBSNs. We collected its public check-in data from Sep

[1] https://sites.google.com/site/dbhongzhi/

	Foursquare	Gowalla
# of users	114,508	107,092
# of POIs	62,462	1,280,969
# of check-ins	1,434,668	6,442,892
time span	Sep 2010-Jan 2011	Feb 2009-Oct 2010

Table 2: Basic statistics of datasets

2010 to Jan 2011 through Twitter with the same crawling strategy as proposed in [5]. This dataset contains 62,462 POIs and 1,434,668 check-ins generated by 114,508 users who live in the USA. Each check-in is stored as user-ID, POI-ID, POI-location in the form of latitude and longitude, check-in timestamp, and POI-content.

Gowalla. This dataset contains 6,442,892 check-in histories which is much more than Foursquare dataset. However, it does not contain the content information about POIs. Therefore, each check-in record has the same format with the above Foursquare dataset except for *POI-content*.

3.1.2 Comparative Approaches

We compare our GE with the following four methods representing the state-of-the-art location-based recommendation techniques.

SVDFeature. SVDFeature [2] is a machine learning toolkit designed to solve the feature-based matrix factorization. To compare with our model GE fairly, we implement it by incorporating more side information beyond the user-POI matrix, including POI content, POI geographical location and check-in time.

JIM. JIM [27] is a joint probabilistic generative model that simulates the decision-making process of users' check-in behaviors. It strategically integrates semantic effect, temporal effect, geographical influence and word-of-mouth effect to overcome the issues of data sparsity and cold start.

PRME-G. PRME [6] is a personalized ranking metric embedding algorithm that jointly models the sequential transition of POIs and user preferences. PRME-G further incorporates geographical influence based on PRME. Although PRME-G is also based on embedding techniques, it utilize two latent spaces: sequential transition space and user preferences space.

Geo-SAGE. Geo-SAGE [17] is a geographical sparse additive generative model for location-based recommendation which incorporates and exploits POI content information and the crowd's preference at a region.

Note that all the above four recommender models fail to keep track of the dynamics of user preferences, thus they cannot well support real-time recommendation.

To further validate the benefits brought by each factor, we design four variants of GE. **GE-S1** is the first simplified version of the GE model where we set $\triangle T = \infty$ in graph G_{vv} to eliminate the sequential effect. **GE-S2** is the second variation of the GE model where we neglect the check-in location, which means we remove the POI-Region graph G_{vr}. As the third simplified version of GE, **GE-S3** does not consider the temporal cyclic effect by removing POI-Time graph G_{vt}. **GE-S4** means our model without considering the semantic effect by removing POI-Word graph G_{vw}.

3.1.3 Evaluation Methods

Given a user profile $\mathcal{D}_u$ in terms of a collection of user check-in records, we first rank them according to their check-in timestamps. Then, we use the 80-th percentile as the cut-off point so that check-ins before this point will be used for training and the rest are for testing. In the training dataset, we choose the last 10% check-ins as the validation data to tune the model hyper-parameters such as the dimension of the latent space. According to the above dividing strategies, we split the dataset $\mathcal{D}$ into the training set $\mathcal{D}_{train}$ and test set $\mathcal{D}_{test}$. To evaluate the recommendation methods, we adopt the evaluation methodology and measurement Accuracy@k proposed in [17, 25, 27, 26]. Specifically, for each check-in record (u, v, τ, l_v, $\mathcal{W}_v$) in $\mathcal{D}_{test}$:

- we calculate the ranking score for the ground truth POI v and all other POIs unvisited previously by u by Eq. (12).
- we form a ranked list by ordering all of these POIs according to their ranking scores. Let p denote the position of v within this list. The best result corresponds to the case where v precedes all the unvisited POIs (that is, $p = 1$).
- We formed a top-k recommendation list by picking the k top ranked POIs from the list. If $p \leq k$ (i.e., the ground truth POI v appears in the top-k recommendation list), we have a hit. Otherwise, we have a miss.

We define hit@k for a single test case as either the value 1, if the ground truth POI v appears in the top-k results, or the value 0, if otherwise. The overall Accuracy@k is defined by averaging over all test cases:

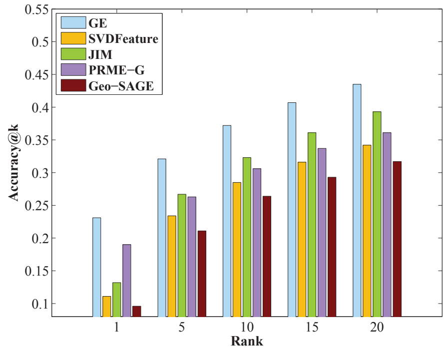
$$Accuracy@k = \frac{\#hit@k}{|\mathcal{D}_{test}|} \tag{13}$$

where #hit@k denotes the number of hits in the whole test set, and $|\mathcal{D}_{test}|$ is the number of test cases.

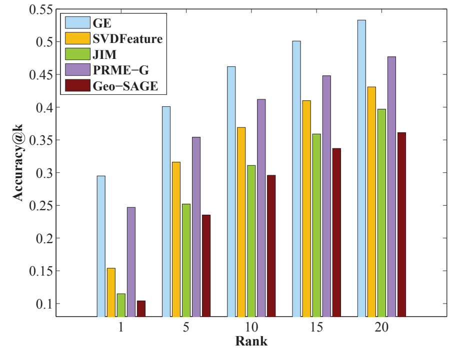
3.2 Recommendation Effectiveness

In this section, we present the experimental results of all recommendation methods with well-tuned parameters. Fig. 2 reports the performance of the recommendation methods on Foursqaure and Gowalla datasets respectively. Note that, we only show the performance when $k = \{1, 5, 10, 15, 20\}$, since a greater value of k is usually ignored for the top-k recommendation task.

Recommendation on Foursquare. Fig. 2(a) present the recommendation accuracy on Foursquare dataset. Clearly, our proposed GE model outperforms other competitor models significantly. Several observations are made from the results: 1) GE and PRME-G achieves much higher recommendation accuracy than other comparison methods in top-1 recommendation, showing the benefits brought by metric embedding method. Besides, GE beats PRME-G. This may be because GE embeds all information into one shared latent space while PRME-G uses two latent spaces to embed user preferences and sequential patterns, respectively. Additionally, our GE model is much more versatile and exploits more side information in a unified way. 2) JIM perform better than PRME-G except for top-1 recommendation, as JIM considers more side information than PRME-G, such as content information of POIs and their popularity. 3) Geo-SAGE drops behind other methods, this is because Geo-SAGE ignores both sequential effect and temporal effect which play critical roles in location-based recommendation. 4) Although GE and SVDFeature use the same types of features and information, GE achieves much higher recommendation accuracy, showing the advantages of our developed graph-based embedding method and dynamic user preference modeling technique.



(a) Performance on Foursquare Dataset



(b) Performance on Gowalla Dataset

Figure 2: Recommendation Effectiveness

Methods \ Acc@k	$k=1$	$k=5$	$k=10$	$k=15$	$k=20$
GE-S1	0.183	0.252	0.293	0.322	0.347
GE-S2	0.225	0.296	0.339	0.370	0.396
GE-S3	0.214	0.278	0.323	0.348	0.368
GE-S4	0.224	0.288	0.333	0.366	0.394
GE	**0.231**	**0.321**	**0.372**	**0.407**	**0.435**

Table 3: Recommendation Accuracy of GE Variants.

Recommendation on Gowalla. Fig. 2(b) reports the performance of the recommendation models on Gowalla dataset. From the figure, we can see that the trend of comparison result is similar to that presented in Fig. 2(a). However, JIM slightly outperforms PRME-G on Foursquare dataset while PRME-G exceeds JIM on Gowalla. This is because Gowalla dataset has no content information. Moreover, all recommendation methods achieve higher accuracy on this dataset except JIM. Analysing the dataset, we find that users in the Gowalla dataset have more check-in records than users in the Foursquare dataset on average, which enables the models to capture users' preferences more accurately. Besides, the loss of content information degenerates the performance of JIM, but does not have much impact on our GE model, which reveals the robustness of our model.

3.3 Impact of Different Factors

To explore the benefits of incorporating the sequential effect, geographical influence, temporal cyclic effect and semantic effect into GE model respectively, we compare our GE model with four variations, GE-S1, GE-S2, GE-S3, GE-S4, which are introduced in Section 3.1.2. Since Gowalla dataset has no content information, we only show the results on Foursquare dataset in Table 3.

From the result, we first observe that GE consistently outperforms the four variants, indicating that GE benefits from simultaneously considering the four factors in a joint way. Second, we find that the contribution of each factor to improving recommendation accuracy is different. Specifically, according to the importance of the four factors, they can be ranked as follows: Sequential Effect > Temporal Effect > Content Effect > Geographical Influence. The performance gap between GE and GE-S1 is the most significant, showing that sequential effect plays an important role in location-based recommendation. The improvement of GE over GE-S3 is also great, which validates human movements exhibit strong temporal cyclic patterns. This shows that temporal related factors matter a lot in users' daily routines.

(a) Impact of N and d on Foursquare Dataset

$N(*m)$ \ d	70	80	90	**100**	110	120
10	0.259	0.260	0.260	0.261	0.261	0.261
50	0.341	0.345	0.347	0.351	0.351	0.351
100	0.352	0.361	0.365	0.368	0.371	0.372
150	0.357	0.364	0.370	**0.372**	0.373	0.373
200	0.357	0.364	0.369	0.372	0.373	0.373
250	0.356	0.364	0.369	0.372	0.373	0.373

(b) Impact of N and d on Gowalla Dataset

$N(*m)$ \ d	70	80	90	**100**	110	120
100	0.418	0.422	0.424	0.426	0.426	0.426
150	0.435	0.440	0.443	0.446	0.447	0.447
200	0.443	0.449	0.453	0.456	0.456	0.457
250	0.448	0.453	0.458	**0.462**	0.462	0.463
300	0.448	0.454	0.459	0.462	0.463	0.463
350	0.449	0.454	0.459	0.463	0.464	0.464

Table 4: Recommendation Accuracy@10

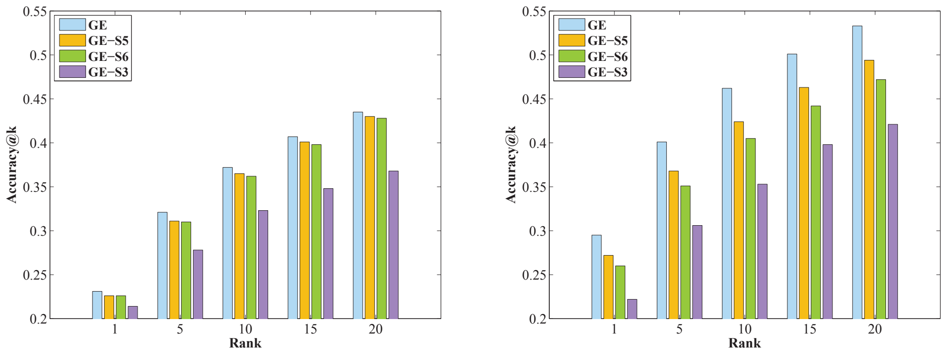
3.4 Sensitivity Analysis of Model Parameters

Tuning model parameters is critical to the performance of the proposed model, such as the embedding dimension d, the number of samples N, and the time interval $\triangle T$ in our GE model. We therefore study the impact of these model parameters in this section. Besides, we will also study the impact of different time split strategies that capture different temporal cyclic patterns.

3.4.1 Impact of Model Parameters N and d

Table 4 presents the performance of our proposed GE in terms of $Accuracy@10$ with different number of samples N and embedding dimensions d. Note that, the unit of N is set to 1 million.

Similar observations can be made on both datasets. From the results, we observe that the recommendation accuracy of GE is not highly sensitive to the dimension d, but still presents a tendency that its recommendation accuracy increases with the increasing number of dimension d holistically, and then does not change much when d is larger than 100. GE is sensitive to the number of samples N, the accuracy varies a lot w.r.t N. First, the performance of GE increases quickly with the increasing of N, this is because the model has not achieved convergence. Then, it does not change significantly when the number of samples becomes



(a) Performance on Foursquare Dataset (b) Performance on Gowalla Dataset

Figure 3: Recommendation effectiveness of different temporal cyclic patterns.

(a) Impact of $\triangle T$ on Foursquare Dataset

$\triangle T$ \ Acc@k	1	5	10	15	20
5	0.186	0.244	0.277	0.301	0.321
10	0.210	0.284	0.326	0.356	0.380
15	0.222	0.303	0.350	0.383	0.410
20	0.226	0.311	0.362	0.397	0.426
25	**0.231**	**0.321**	**0.372**	**0.407**	**0.435**
30	0.220	0.313	0.366	0.403	0.433
35	0.201	0.309	0.360	0.396	0.424
40	0.193	0.277	0.322	0.358	0.387

(b) Impact of $\triangle T$ on Gowalla Dataset

$\triangle T$ \ Acc@k	1	5	10	15	20
1	0.168	0.239	0.276	0.303	0.324
3	0.236	0.320	0.370	0.405	0.433
5	0.270	0.369	0.426	0.465	0.495
7	0.287	0.390	0.449	0.489	0.520
9	**0.295**	**0.401**	**0.462**	**0.501**	**0.533**
11	0.290	0.385	0.457	0.498	0.529
13	0.268	0.361	0.430	0.469	0.498
15	0.245	0.338	0.416	0.451	0.488

Table 5: Impact of time interval

large enough, since the model GE has converged. Thus, to achieve a satisfying trade off between effectiveness and efficiency of model training, we set $N = 150M$ and $d = 100$ on Foursquare dataset, and $N = 250M$ and $d = 100$ on Gowalla dataset. There are more edges on Gowalla dataset than that on Foursquare dataset, so more samples are needed. It should be noted that the performance of our GE reported in other experiments is achieved with this parameter settings.

3.4.2 Impact of time interval in sequential patterns

Table 5 investigates the impact of time interval $\triangle T$ in mining POI sequential patterns. From the experimental results, we observe that the performance first improves quickly with the increase of $\triangle T$ and then drop down gradually. The reason of accuracy disparity is that, when $\triangle T$ is small, GE prunes too many POI co-occurrence edges, leading to a extremely sparse POI-POI graph; while $\triangle T$ becomes too large, GE may incorporate too many noise edges and fail to capture the sequential patterns of check-in POIs. Thus, we choose $\triangle T = 25$ days on Foursquare dataset and $\triangle T = 9$ days on Gowalla dataset to achieve the best results. Due to the denser check-in data in the Gowalla dataset on average, the $\triangle T$ is much smaller on Gowalla dataset than that on Foursquare dataset.

3.4.3 Exploring Various Temporal Patterns

So far, we have evaluated the recommendation performance of GE with daily pattern (24 hours of a day), as shown in Fig. 2, in which we divide time into 24 time slots that corresponds to 24 hours. But, our GE model is not limited to one specific temporal cyclic pattern. By taking different definitions of temporal state, many other temporal patterns can be integrated into our GE model, as long as they contain the non-uniformness and consecutiveness properties [7]. In this experiment, we design two additional variants of GE, GE-S5 and GE-S6, to study the effect of weekly pattern (day of the week) and weekday/weekend pattern, respectively. We divide time into 7 time slots in GE-S5, and 2 time slots in GE-S6. Fig. 3 shows the recommendation results of GE, GE-S5, GE-S6 and GE-S3. GE, GE-S5 and GE-S6 exploit daily pattern, weekly pattern and weekday/weekend pattern, respectively, while GE-S3 does not consider any temporal cyclic pattern. The results indicate that exploiting daily pattern, weekly pattern or weekday/weekend pattern can largely improve the location-based recommendation performance, and the improvement brought by exploiting daily pattern is the most significant.

3.5 Test for Cold Start Problem

In this experiment, we conduct experiments to study the effectiveness of different recommendation algorithms in addressing cold-start POI recommendations on the Foursquare dataset. As there is not any check-in information available for the cold-start POIs, PRME-G model does not work in the cold-start scenario. Thus, we compare our GE model with other three recommender models that are able to leverage geographical influence and semantic effect to recommend cold-start POIs.

The experimental results are shown in Fig. 4, from which we have the following observations: 1) our proposed GE model still performs best consistently in recommending cold-start POIs, and its advantage over other three methods is more significant; and 2) by comparing the recommendation results in Fig. 2(a) and Fig. 4, the recommendation accuracy of all algorithms decreases, to different degrees, for cold-start POIs, i.e., the recommendation accuracy of Geo-SAGE and SVDFeature drop drastically while our GE model deteriorate slightly. This is because all JIM, SVDFeature and

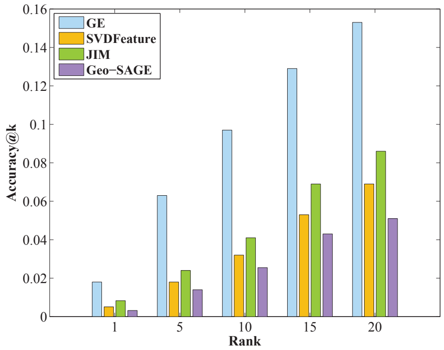


Figure 4: Recommendation for Cold-start POIs

Geo-SAGE recommend cold-start POIs according to their content information and geographical locations; while our GE model can still learn vector representations for cold-start POIs by embed them into the shared latent space, and thus the potential sequential relation between cold-start POIs and normal POIs and the potential correlation between cold-start POIs and time slots are well captured through the shared latent space. In other words, our GE model leverages not only the geographical influence and semantic effect, but also the potential sequential effect and temporal cyclic effect, when recommending cold-start POIs.

4. RELATED WORK

In this section, we discuss existing research related to our work, including location-based recommendation integrating various factors and metric embedding.

Recently, with the easy access of large-scale user activity records in LBSNs, many recent work has tried to improve location-based recommendation by exploiting various side effects such as sequential effect, geographical influence, temporal cyclic effect and semantic effect. **1) Sequential effect.** Most of the studies developed the Markov chain-based methods to capture the sequential patterns of POIs and to reduce the size of the prediction space, Cheng et al. [4] investigated sequential influence using the first-order Markov chain to recommend a new location for the user by only considering user's latest location. Zhang et al. [29] predicted the next location probability through an additive Markov chain, assuming that recent check-in locations usually have stronger influence than those locations checked-in long time ago. **2) Geographical influence.** Many recent studies [10, 3, 20, 24] showed that there is a strong correlation between user check-in activities and geographical distance, thus leveraging the geographical influences to improve recommendation accuracy has been noticed by most of current location-based recommendation work. For example, Lian et al. [10] incorporated spatial clustering phenomenon resulted by geographical influence into a weighted matrix factorization framework to address the challenge from matrix sparsity. **3) Temporal cyclic effect.** Since human movements exhibit strong temporal cyclic patterns, the temporal effect of user check-ins in LBSNs has also attracted much attention from researchers. Gao et al. [7] studied the temporal cyclic patterns of user check-ins in terms of temporal non-uniformness and temporal consecutiveness. Yuan et al. [28] incorporated the temporal cyclic information into a user-based collaborative filtering framework for time-aware POI recommendation. **4) Semantic effect.** Researchers explored the semantic information of POIs mainly to alleviate the problem of data sparsity. Hu et al. [8] proposed a spatial topic model for location-based recommendation considering both spatial aspect and textual aspect of user posts from Twitter. Liu et al. [12] studied the effect of POI associated tags for POI recommendation with an aggregated LDA and matrix factorization method. Yin et al. [25, 17] exploited both personal interests and local preferences based on the contents associated with spatial items.

Recently, there are also some work turned their eyes on integrating analysis of joint effect of the above factors to alleviate the issue of data sparsity, cold start and spatiotemporal context-aware recommendation [23, 27]. For example, Yin et al. [27] proposed a probabilistic generative model for jointly modeling of geographical influence, temporal cyclic effect and semantic effect. However, our work is a generic method, we not only gain the benefits brought by exploiting various kinds of factors, but also utilize the metric embedding methods to integrate different factors more smoothly in a shared latent space.

Embedding methods have been long studied and proved to be effective in capturing latent semantics of how items (e.g. words in sentences) interact with each other. For example, Tang et al. [16, 15] learned words embedding to make document classification, and verified its effectiveness. Chen et al. [1] adopted metric embedding in the music playlist prediction and proposed a Logistic Markov embedding model for generating the playlists. The location-based recommendation using metric embedding methods is relatively less. In our previous work [18], we applied the graph-based embedding method to the next POI recommendation, but only employed one homogeneous graph to embed the CF related information and sequential effect, as we mainly focused on the problem of successive POI recommendation in [18]. In this work, we proposed to utilize the multi-heterogenous graphs to capture and model the four factors in a unified framework. Cold start problem can also be solved easily in our framework, as POIs without any check-in can be still embedded by leveraging their interactions with time, locations and textual words. PRME proposed by Feng et al. [6] is also the typical one which exploits pair-wise ranking scheme. However, our work is a graph-based method, which integrates various factors into a shared metric by different bipartite graphs while PRME embeds user preference and sequential patterns in two different metric respectively, and only considered sequential patterns of POIs and geographical influence.

Our work in this paper distinguishes itself from other researches in several aspects. Firstly, to the best of our knowledge, it is the first effort that investigates the joint effect of sequential effect, geographical influence, temporal cyclic effect and semantic effect to address the challenges of data sparsity, cold start, context-aware recommendation and dynamic user preferences in a unified way utilizing the graph-based embedding method. Secondly, although research [6] exploited the metric embedding for location-based recommendation, it embedded user preferences and sequential transition into two different spaces which may lose some potential relationship between users and POIs. In contrast, our proposal encode all the factors into a shared latent space via graph-based method. Moreover, incapability in the cold-start scenario makes PRME-G drop behind our model. Thirdly, to support real-time recommendation, we propose a novel method for dynamic preference modeling based on the learnt embedding of POIs.

5. CONCLUSIONS

In this paper, we developed a graph-based embedding (GE) model to integrated capture the sequential effect, geographical influence, temporal cyclic effect and semantic effect in a unified way by embedding the four corresponding relational graphs (POI-POI, POI-Region, POI-Time, POI-Word) into a shared low dimensional space for the ease of data sparsity, cold-start problem and context-aware recommendation in LBSNs. Besides, to support recommendation in a real-time manner, we developed a novel time-decay method to dynamically compute the user's latest preferences based on the embedding of his/her checked-in POIs learnt in the latent space. To the best of our knowledge, this is the first work that uses the metric embedding method to unify dynamic user preferences and various of factors in a principled manner. We conducted extensive experiments to evaluate the performance of our GE model on two real large-scale datasets. The results showed superiority of our proposal over other competitor methods, especially in recommending cold-start POIs. Moreover, we studied the importance of each factor in improving recommendation under the same framework, and found that both sequential effect and temporal cyclic effect play a dominant role in location-based recommendation and the daily pattern is the most significant temporal cyclic pattern in users' daily behaviors.

6. ACKNOWLEDGMENTS

This work was supported by ARC Discovery Early Career Researcher Award (DE160100308) and National Basic Research Program of China (2013CB329305). It was also partially supported by and Natural Science Foundation of China (61303164, 61572335, 61402447 and 61502466), the Development Plan of Outstanding Young Talent from Institute of Software, Chinese Academy of Sciences (ISCAS2014-JQ02) and Jiangsu Natural Science Foundation of China (BK20151223).

7. REFERENCES

[1] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In *KDD*, pages 714–722, 2012.

[2] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. Svdfeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.*, 13(1):3619–3622, Dec. 2012.

[3] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, pages 17–23, 2012.

[4] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, pages 2605–2611, 2013.

[5] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

[6] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan. Personalized ranking metric embedding for next new poi recommendation. In *AAAI*, pages 2069–2075, 2015.

[7] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, pages 93–100, 2013.

[8] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32, 2013.

[9] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola. Reducing the sampling complexity of topic models. In *KDD*, pages 891–900, 2014.

[10] 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 *KDD*, pages 831–840, 2014.

[11] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan 2003.

[12] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, pages 396–404, 2013.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[14] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

[15] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, pages 1165–1174, 2015.

[16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[17] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou. Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In *KDD*, pages 1255–1264, 2015.

[18] M. Xie, H. Yin, F. Xu, H. Wang, and X. Zhou. Graph-based metric embedding for next poi recommendation. In *WISE*, 2016.

[19] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz. On the semantic annotation of places in location-based social networks. In *KDD*, pages 520–528, 2011.

[20] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[21] H. Yin and B. Cui. *Spatio-Temporal Recommendation in Social Media*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[22] H. Yin, B. Cui, L. Chen, Z. Hu, and X. Zhou. Dynamic user modeling in social media systems. *ACM Trans. Inf. Syst.*, 33(3):10:1–10:44, Mar. 2015.

[23] H. Yin, B. Cui, Z. Huang, W. Wang, X. Wu, and X. Zhou. Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In *ACM Multimedia*, pages 819–822, 2015.

[24] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen. Lcars: A spatial item recommender system. *ACM Trans. Inf. Syst.*, 32(3):11:1–11:37, July 2014.

[25] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *KDD*, pages 221–229, 2013.

[26] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen. Adapting to user interest drift for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1–14, June 2016.

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

[28] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.

[29] J.-D. Zhang, C.-Y. Chow, and Y. Li. Lore: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*, pages 103–112, 2014.