POI2Vec: Geographical Latent Representation for Predicting Future Visitors

Shanshan Feng,[1] Gao Cong,[2] Bo An,[2] Yeow Meng Chee[3]

[1]Interdisciplinary Graduate School
[2] School of Computer Science and Engineering
[3]School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
{sfeng003@e., gaocong@, boan@, ymchee@}ntu.edu.sg

Abstract

With the increasing popularity of location-aware social media applications, Point-of-Interest (POI) recommendation has recently been extensively studied. However, most of the existing studies explore from the users' perspective, namely recommending POIs for users. In contrast, we consider a new research problem of predicting users who will visit a given POI in a given future period. The challenge of the problem lies in the difficulty to effectively learn POI sequential transition and user preference, and integrate them for prediction. In this work, we propose a new latent representation model *POI2Vec* that is able to incorporate the geographical influence, which has been shown to be very important in modeling user mobility behavior. Note that existing representation models fail to incorporate the geographical influence. We further propose a method to jointly model the user preference and POI sequential transition influence for predicting potential visitors for a given POI. We conduct experiments on 2 real-world datasets to demonstrate the superiority of our proposed approach over the state-of-the-art algorithms for both next POI prediction and future user prediction.

Introduction

With the increasing availability of location-aware social media data, such as check-ins to Points-of-Interests (POIs), the problem of modeling users' mobility behaviors and POI recommendation has been extensively studied (Cho, Myers, and Leskovec 2011; Ye et al. 2011). Many variants of the POI recommendation problem are also explored, among which the next-POI recommendation problem has recently attracted significant attention (Cheng et al. 2013; Feng et al. 2015; He et al. 2016), which is to recommend POIs for a user to visit in the near future.

Most of existing POI recommendation studies focus on suggesting POIs for users from the users' perspective. In contrast, very little research has been done to predict visitors for a location. In this paper, we investigate a new research problem from the perspective of POI: Given a POI, we predict the users who will visit this POI in the next few hours. For example, given a restaurant and the current time, we want to predict potential consumers who would visit this restaurant in the next several hours. Obviously, identifying the potential visitors for locations is of great value. Among many other applications, it can help POI holders to find potential customers for marketing.

A user's mobility is highly influenced by his recent visits and his personal interest (Cheng et al. 2013). However, the users' check-in data are very sparse, and it is challenging to model the sequential influence of POIs and user preference. This is also the main challenge for predicting future visitors for a given POI. The progress in neural network language models demonstrates that the word2vec technique (Mikolov and Dean 2013; Mikolov et al. 2013) is able to effectively capture the sequential semantic relationships among words. Very recently, the word2vec algorithm has been utilized to model users' sequential check-ins (Liu, Liu, and Li 2016). However, the method (Liu, Liu, and Li 2016) has two limitations. First, it fails to incorporate geographical influence of POIs, i.e., users tend to visit nearby POIs, in the word2vec model. The geographical influence is very important in modeling user movement behaviors as shown in almost all the previous works in POI recommendation. Second, it separately models the sequential transition and user preference. Since each check-in behavior is related to both the sequential transition and user preference, it is more reasonable to jointly model these two factors.

To incorporate geographical influence, we propose a novel latent representation model, called **POI2Vec**. In our model, each POI is represented as a vector in a latent low dimensional space, and the inner product of two vectors reflects the relevance between two POIs. To learn the latent vectors, we exploit the *hierarchical softmax* (HS) technique (Morin and Bengio 2005), which is widely used in neural network language models by exploiting a binary tree structure. For the hierarchical softmax, the key to achieve good performance is constructing a proper binary tree over items (Mnih and Hinton 2009). Instead of using Huffman tree in word2vec, we develop a novel binary tree building method, which is able to incorporate the geographical coordinates of POIs. We hierarchically split the POIs into different regions, and then build a binary tree over the POIs in each region. Since a POI may also influence POIs in adjacent regions, we assign a POI to multiple nearby regions. In the generated binary tree, one POI may occur more than once, which is able to capture the various relationships with other POIs.

Furthermore, we model the user preference with the proposed POI2Vec model, where each user is also represented with a latent vector. We propose a model to jointly learn the latent representations of the users and POIs by considering both the user preference and sequential transition. To predict potential visitors for a POI, we consider both the users with recent locations and the users whose recent positions are unknown. For users with recent locations, we aggregate the user preference and sequential transition for prediction. For the other users, we only make use of the user preference.

The main contributions are summarized as follows.

- We propose a novel latent representation model POI2Vec, which incorporates the geographical influence of POIs, to predict the potential visitors for a location in the next few hours. To the best of our knowledge, this is the first work to incorporate the geographical influence in the framework of word2vec.
- We develop a method to jointly learn the latent representation for users and POIs. We aggregate the user preference and POI sequential influence to predict the future visitors for a POI. This model can also be used to predict the next POIs for a user.
- We conduct extensive experiments to compare our algorithm with state-of-the-art approaches, and our method significantly outperforms the baselines for both next POI prediction and future user prediction.

Related Work

The location recommendation problem has been extensively investigated (Lian et al. 2014; Li, Ge, and Zhu 2016). Most of previous solutions are based on Collaborative Filtering methods (Yuan et al. 2013a; Ye et al. 2011) or Factorization models (Cheng et al. 2012; Li et al. 2015), and model user preference by exploiting the geographical influence. The problem of user recommendation for a location is also studied (Yuan et al. 2013b; Zhao et al. 2015), which is to recommend potential users for a POI. However, it does not predict future users for the next several hours.

Recently, the next POI recommendation task has attracted significant attention, which exploits the sequential transition, and various methods have been proposed. Some studies utilize the Markov Chain (Zhang, Chow, and Li 2014) or Hidden Markov Chain model (Ye, Zhu, and Cheng 2013) to model the sequential influence. Meanwhile, the other studies utilize the factorization model to model the personalized sequential transition: (Cheng et al. 2013) exploits the Factorized Personalized Markov Chain (FPMC) (Rendle, Freudenthaler, and Schmidt-Thieme 2010) to model the personalized POI transition; (Feng et al. 2015) uses the Metric Embedding to model the user preference and POI transition. Our work differs from these next POI recommendation studies in two aspects at least. First, we consider a new task from a different perspective: finding the potential visitors in the next few hours for a POI. Second, we use the latent representation algorithm to jointly learn the user preference and sequential transitions.

Recently, other features have been considered for the next POI recommendation problem. For example, (Zhang and Wang 2015; Zhao et al. 2016; Liu et al. 2016) consider the temporal influence; (He et al. 2016) incorporates the category transition pattern. Such works are complementary to our work.

Word2vec techniques (Mikolov and Dean 2013) are developed for the neural network language models. There are two efficient solutions (Mikolov et al. 2013): one is hierarchical softmax, and the other is negative sampling. Although word2vec techniques are proposed for natural language processing, they have been adopted by various applications, such as next product recommendation (Wang et al. 2015). In particular, (Liu, Liu, and Li 2016) utilizes the word2vec technique to model the check-in sequence. However, as mentioned earlier, it fails to incorporate the geographical influence when modeling the sequential POIs, and it is not a joint model. Different from existing word2vec techniques, we propose a new model POI2Vec, which learns the representation of POIs by considering the geographical influence of POIs.

Future Visitor Prediction Problem

We denote the set of users by $\mathcal{U}$, and the set of POIs by $\mathcal{L}$, where each POI l is associated with its geographical coordinates $\langle l^{Lat}, l^{Lon} \rangle$. Let $\mathcal{H}$ be the historical check-in dataset. Each check-in (u, l, t) means that user u visits location l at time t. Given a POI, we aim to identify the potential visitors who will visit the POI in the next several hours. The problem is defined as follows.

Definition 1 ***(Future Visitor Prediction Problem)*** *Consider a set of users $\mathcal{U}$ and a set of POIs $\mathcal{L}$. Given a POI l, current time t, and time threshold τ, the problem aims to identify a set of users who will visit this POI in the time period $[t, t+\tau]$.*

POI2Vec Representation Model

We first present a POI-to-vector (POI2Vec) representation method to model the POI sequential transition. The novelty of POI2vec is that it is able to incorporate geographical influence. Then we extend the POI2Vec to jointly model user preference and POI sequential transition. Lastly, we introduce approaches to predict future visitors for a given POI.

POI2Vec Model for Sequential Transition

The sequential transition between check-ins is very important for predicting future visitors since a user's subsequent movement behavior is highly influenced by the previously visited locations. When the time interval of two successive check-ins is small, the next check-in is influenced by the previous one (Cheng et al. 2013). However, due to the sparsity of check-in data, it is challenging to model the sequential transition.

Latent Representation Method The recent progress in neural network language models (Mikolov and Dean 2013; Le and Mikolov 2014) indicates that latent representation methods can effectively capture the sequential semantic relationships among words. Our analysis on two check-in datasets show that the distribution of POI frequency also follows the power law distribution as does the word frequency distribution. This motivates us to utilize the well-known word2vec (Mikolov et al. 2013) technique to model the check-in sequences.

We next introduce the basic word2vec framework to capture the POI sequential transition. Given a user u, and his current location l_c^u, the context $C(l_c^u)$ is the POIs that u have visited before l_c^u within a predefined time period. We define $C(l_c^u) = \{l_i^u, 0 < \Delta(l_i^u, l_c^u) < \tau\}$, where $\Delta(l_i^u, l_c^u)$ is the time interval between visiting l_i^u and l_c^u. The goal of POI sequential modeling is to estimate the probability of visiting a POI given its contextual POIs.

We represent each POI l with a vector $w(l) \in \mathcal{R}^D$ in D dimension latent space. We adopt the Continuous Bag-of-Words (CBOW) architecture (Mikolov and Dean 2013), which is to predict an item given its contexts. The probability of $\Pr(l|C(l))$ is defined with a softmax function:

$$\Pr\left(l|C(l)\right) = e^{(w(l)\cdot\Phi(C(l)))} \Big/ Z(C(l)) \tag{1}$$

where $\Phi(C(l)) = \sum_{l_c \in C(l)} w(l_c)$ is the sum of vector of contextual POIs, and $Z(C(l)) = \sum_{l_i \in \mathcal{L}} e^{(w(l_i)\cdot\Phi(C(l))}$ is the normalization term.

It is computationally expensive to directly compute Eq. (1) since calculating $Z(C(l))$ needs to enumerate each item $l_i \in \mathcal{L}$. To alleviate this issue, we adopt the *hierarchical softmax* (Morin and Bengio 2005), which is widely used to compute softmax. The hierarchical softmax utilizes a binary tree to organize the items. The leaves of the binary tree correspond to the items. In the hierarchical softmax, the structure of the binary tree is important for achieving good performance (Mnih and Hinton 2009). The mostly widely used tree generating method is Huffman tree based on frequency of items since it can get the shortest average path length, thus obtaining the best learning efficiency.

Incorporating Geographical Influence The geographical influence is a very important factor to model both the sequential transition and user preference as shown in previous studies on POI recommendation. For example, a user is more likely to visit nearby places in the next several hours. However, the existing frequency-based structure of hierarchical softmax fails to capture the geographical influence. To solve this problem, we develop a geographical binary tree structure for the POI2Vec model, which incorporates the spatial information into the binary tree.

Since the nearby POIs have high relevance, they should be assigned closely in the binary tree. We split the POIs into a hierarchy of binary regions such that the nearby POIs are more likely to be clustered into the same region. To build the binary hierarchy for POIs, we recursively split each region into two equal size sub-regions, until at least one edge of the regions is smaller than $2 \times \theta$, where θ is the region size threshold. In this way, the regions are rectangles larger than $\theta \times \theta$ square. The regions should also be represented in a binary tree manner. As illustrated in Figure 1, we firstly divide the POIs into two equal regions R_0 and R_1. Then we split R_0 into regions R_{00} and R_{01}, and split R_1 into regions R_{10} and R_{11}.

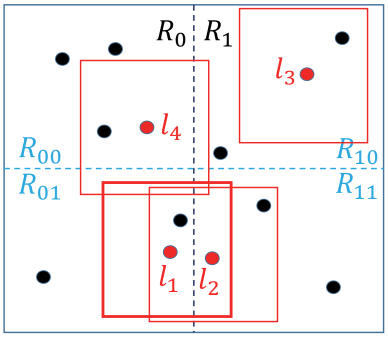


Figure 1: The illustration of buliding binary tree over POIs

In the resulted binary tree, each POI is assigned to a single region. For example, l_1 is in region R_{01} and l_2 is in region R_{11} in Figure 1. However, this fails to capture the spatial influences of POIs in different regions. For example, two POIs l_1 and l_2 are close to each other, but they belong to two different regions. To solve this issue, we further improve the method by assigning a POI to multiple regions.

To assign a POI to multiple regions whose POIs may be influenced by the POI, we consider the influence area of each POI as users' movement is influenced by the spatial distance. The influence area is defined as a $\theta \times \theta$ square centered at each POI. If the influence area of a POI l overlaps with a region R in the binary tree, we assign l to region R. For instance, as shown in Figure 1, the square centered at POI l_1 overlaps with region R_{01} and R_{11}, and thus POI l_1 is assigned to region R_{01} and region R_{11}.

Since the size of the influence area $\theta \times \theta$ is smaller than the regions, it is easy to prove the following result.

Proposition 1 *In the POI2Vec model, the number of regions of a POI will be 1, 2 or 4.*

We denote the set of regions of POI l by Ω^l. For instance, in Figure 1, $\Omega^{l_3} = \{R_{10}\}$, $\Omega^{l_1} = \{R_{01}, R_{11}\}$, and $\Omega^{l_4} = \{R_{00}, R_{01}, R_{11}, R_{10}\}$. The number of regions of POI for l_3, l_1 and l_4 are: $|\Omega^{l_3}| = 1$, $|\Omega^{l_1}| = 2$, and $|\Omega^{l_4}| = 4$.

If a POI is assigned to several regions, we compute the probability distribution over these regions as follows: The probability of l belonging to region R_i is $\Pr(R_i) = S_{R_i}^l \Big/ \sum_{R_k \in \Omega^l} S_{R_k}^l$, where $S_{R_i}^l$ is the size of overlap between influence area of l and region R_i. For instance, the region probability distribution of POI l_1's is: $\Pr(R_{01}) = 0.7$ and $\Pr(R_{11}) = 0.3$.

A region may contain many POIs, and we need to further construct a hierarchy to organize these POIs. For each region, we build a Huffman tree based on the frequency of POIs by following the work (Mikolov and Dean 2013). Therefore, in the generated binary tree, the upper levels are the hierarchy of spatial regions, and the lower levels are hierarchy of POIs in each region. Figure 2 illustrates the binary tree generated by POI2Vec model. A POI may have several paths. For example, l_1 occurs twice in the generated binary tree: one is in region R_{01} and the other is in R_{11}.

In summary, the proposed POI2Vec method has two advantages over the conventional Huffman tree struc-

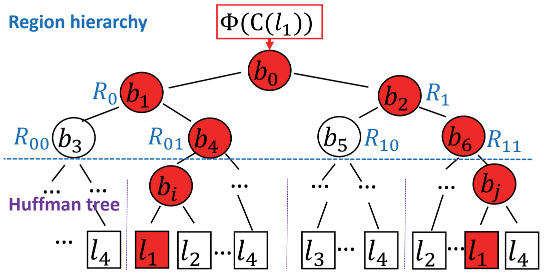


Figure 2: The binary tree generated by POI2Vec model

ture (Mikolov and Dean 2013). First, it considers the geographical influence of POIs in the process of building binary tree, and POIs under the same region node are geographically close to each other. Second, in the POI2Vec model, a POI may occur multiple times in the binary tree, while in conventional model, each POI only occurs once. As demonstrated in (Mnih and Hinton 2009), the multiple occurrence model would perform better than single occurrence model, since an item may have various relationships with other items.

Once the binary tree is generated, we can effectively compute the probability in Eq. (1) as follows.

Probability Estimation The hierarchical softmax model approximates the softmax by estimating the probability of a path from the root to the leaf node. In the binary tree, leaf nodes are POIs and the others are inner nodes. Each inner node can be treated as a binary classifier. The path to leaf node l is defined as a sequence of inner nodes $path = (b_0^l, b_1^l, ..., b_n^l)$. Then the probability of observing l along the path $path$ can be estimated by

$$\Pr\left(l|C(l)\right)^{path} = \prod\nolimits_{b_i^l \in path} \Pr(b_i^l|\Phi(C(l)). \qquad (2)$$

Each inner node b in the binary tree has a latent vector $\Psi(b_i^l) \in \mathcal{R}^D$, which can be viewed as the parameters of a binary classifier. Here $\Pr(b_i^l|\Phi(C(l)))$ is defined as

$$\Pr\left(b_i^l|\Phi(C(l))\right) = \sigma\left(\Psi(b_i^l) \cdot \Phi(C(l))\right), \qquad (3)$$

where $\sigma(x) = 1/(1+e^{-x})$ is a sigmiod function.

For example, as illustrated in Figure 2, one path for item l_1 is $path_1 = (b_0, b_1, b_4, b_i)$. For a classifier in the binary tree, we define the left as “true” and right as “false”. Then the probability of this path $\Pr\left(l_1|C(l_1)\right)^{path_1} = \sigma\left(\Psi(b_0) \cdot \Phi(C(l_1))\right) \times \left(1 - \sigma\left(\Psi(b_1) \cdot \Phi(C(l_1))\right)\right) \times \sigma\left(\Psi(b_4) \cdot \Phi(C(l_1))\right) \times \sigma\left(\Psi(b_i) \cdot \Phi(C(l_1))\right)$.

Since each POI may have several regions in the POI2Vec model, a POI may correspond to multiple paths in the generated tree. We need to consider all of them for calculating the probability. We denote the set of paths for POI l by $\mathcal{P}(l)$. Each path $path_k$ is associated with a probability $\Pr(path_k)$, which is the same with the probability of its corresponding region $\Pr(R_k)$, where R_k is the region contained in the path $path_k$. For instance, in Figure 2, POI l_1 has two paths in the binary tree: $path_1 = (b_0, b_1, b_4, b_i)$ and $path_2 = (b_0, b_2, b_6, b_j)$. Here $path_1$ contains the region R_{01}, and $path_1$ contains the region R_{11}. Recall that the region probability distribution of POI l_1 is $\Pr(R_{01}) = 0.7$ and $\Pr(R_{11}) = 0.3$. Thus $\Pr(path_1) = 0.7$ and $\Pr(path_2) = 0.3$. The probability of observing l given context $C(l)$ is redefined as

$$\Pr\left(l|C(l)\right) = \prod\nolimits_{path_k \in \mathcal{P}(l)} \Pr(path_k) \times \Pr\left(l|C(l)\right)^{path_k} \quad (4)$$

Note that in Eq. (1), the time complexity of calculating $\Pr\left(l|C(l)\right)$ is $O(|\mathcal{L}|)$. The number of leaf nodes in the POI2Vec model is $(a \times |\mathcal{L}|)$, where a is the average number of paths of all POIs, and is a constant $(1 \leq a \leq 4)$ based on Proposition 1. In a binary tree, the average path length for $(a \times |\mathcal{L}|)$ leaf nodes is $O(\log(|\mathcal{L}|))$. Therefore, in Eq. (4), the time complexity of calculating one observation is $O(\log(|\mathcal{L}|))$, which is greatly reduced from $O(|\mathcal{L}|)$.

Parameter Learning The objective of POI2Vec model is to maximize the posterior probability of observing all sequential POIs, assuming the observations are independent with each other:

$$\Theta = \arg\max_{\Theta} \prod\nolimits_{(l,C(l)) \in \mathcal{H}} \Pr(l|C(l)), \qquad (5)$$

where $\Theta = \{W(\mathcal{L}), \Psi(\mathcal{B})\}$ is the set of parameters. Here $W(\mathcal{L})$ indicates the latent representations of all the POIs $l \in \mathcal{L}$, and $\Psi(\mathcal{B})$ is the set of parameters for the inner nodes. Since there are $(a \times |\mathcal{L}|)$ leaf nodes in the binary tree, the number of inner nodes is $(a \times |\mathcal{L}| - 1)$. We can learn all the parameters by Stochastic Gradient Descent (SGD) method (Rong 2014).

Extend POI2Vec Model for User Preference

User preference is another important factor for modeling users’ mobility, and predicting potential users for the target POI. It is necessary to model user preference in the model. Previous work (Liu, Liu, and Li 2016) models user preference and sequential transition separately, which fails to capture their interaction. We extend the POI2Vec model to jointly learn the user preference and sequential transition of POIs. Similar to POIs, each user u is also represented as a vector $x(u) \in \mathcal{R}^D$ in the latent space.

As reported in (Feng et al. 2015), the time interval between some consecutive POIs is large. For a check-in, if there is no check-in in the last τ hours, only the user preference will be considered. In other words, for a check-in without recent context, this check-in is only related to the user preference. Similar to Eq. (1), the probability that user u visits POI l is estimated by

$$\Pr(l|u)) = e^{(w(l) \cdot x(u))} \Big/ Z(u), \qquad (6)$$

where $Z(u) = \sum_{l_i \in \mathcal{L}} e^{(w(l_i) \cdot x(u)}$ is the normalization term. Similar to $\Pr(l|C(l))$, $\Pr(l|u))$ can be computed by the hierarchical softmax.

For a check-in for which there exist recent historical check-ins, this check-in is related to both the user preference and contextual influence. We assume that user preference and contextual influence are independent. Given a user u and location context $C(l)$, the probability of observing l is

$$\Pr(l|u, C(l)) = \Pr(l|u) \times \Pr(l|C(l)). \qquad (7)$$

Depending on the presence of context or not, the probability of observing a check-in is computed by

$$\Pr(u, l, t) = \begin{cases} \Pr(l|u, C(l)) & \text{if } C(l) \text{ exists} \\ \Pr(l|u) & \text{otherwise} \end{cases} \tag{8}$$

The objective of the model is to maximize the posterior probability of observing all check-ins:

$$\Theta = \arg\max_{\Theta} \prod\nolimits_{(u,l,t)\in\mathcal{H}} \Pr(u, l, t), \tag{9}$$

where $\Theta = \{W(\mathcal{L}), X(\mathcal{U}), \Psi(\mathcal{B})\}$ is the set of parameters. Here $X(\mathcal{U})$ is the set of latent representations for all the users. We learn the parameters similarly as we do for the POI2Vec model.

Predict Future Visitors

Once we learn the latent representations of users $X(\mathcal{U})$ and representations of POIs $W(\mathcal{L})$, we can utilize them to find users who will visit a POI in the future.

We first consider users who have recent check-ins in the last few hours. Given a time point t, if a user has check-ins in the time period $[t - \tau, t]$, we say this user is a user with recent positions. For such users, we make use of their recent positions to determine their inclination to visit the given POI. If a user visits several POIs in the period $[t - \tau, t]$, we only keep the latest POI l^c, which represents his latest location. Additionally, we also utilize the user preference. Given a target POI l, and user u with his latest position l^c, the likelihood that u visits l in the near future is defined by: $\mathcal{F}\left(x(u) \cdot w(l), w(l^c) \cdot w(l)\right)$, where $x(u) \cdot w(l)$ reflects the user preference and $w(l^c) \cdot w(l)$ indicates the sequential influence. Here $\mathcal{F}()$ is an aggregation function to fuse them.

In this paper, we employ two popular aggregate functions to combine the user preference and sequential influence. The first aggregation is Max, which chooses the more important factor by taking the larger value: $\mathcal{F}(a, b) = Max(a, b)$. The second aggregation function is Sum, which linearly combines the two factors: $\mathcal{F}(a, b) = Sum(a, b)$.

Due to the sparsity of check-in data, most of the users do not have recent check-ins in the past several hours. For such users, since their latest positions are unknown, we can only exploit the user preference to predict the likelihood that these users visit the given POI. The prediction score is computed by $x(u) \cdot w(l)$.

To identify the potential visitors for a given POI, we need to consider users with recent positions and users without recent positions. For each user $u \in \mathcal{U}$, we compute the score:

$$s(u, l) = \begin{cases} \mathcal{F}\left((x(u) \cdot w(l), w(l^c) \cdot w(l)\right) & \text{with recent positions} \\ x(u) \cdot w(l) & \text{otherwise} \end{cases} \tag{10}$$

We rank all the users by their scores and select top-K users as the potential visitors for the target location.

Experiments

Experimental Setup

We conduct experiments on two publicly available datasets. The first one is the Foursquare check-ins within Singapore (Yuan et al. 2013a) and the second one is the Gowalla check-ins within Houston (Liu et al. 2013). We remove the users who have fewer than 5 check-ins, and the POIs which have been visited by fewer than 5 users by following the previous work. Table 1 summarizes the statistics of datasets.

Dataset	#User	#POI	#Check-in
Foursquare	2321	5596	194108
Gowalla	4627	15234	362783

Table 1: Statistics of two datasets

For both datasets, we use the first 90% chronological check-ins as the training set, the $90 - 95\%$ as the tuning set, and the last 5% as test set. We consider two evaluation tasks. The first one is to evaluate the quality of latent representation for sequential transition in the next POI prediction task. The second task is to evaluate the accuracy of future visitors prediction. Following (Feng et al. 2015), we set $\tau = 6$ hours in this paper. Based on the tuning set, the number of dimensions $D = 200$, region size threshold $\theta = 0.1$, learning rate is set at 0.005.

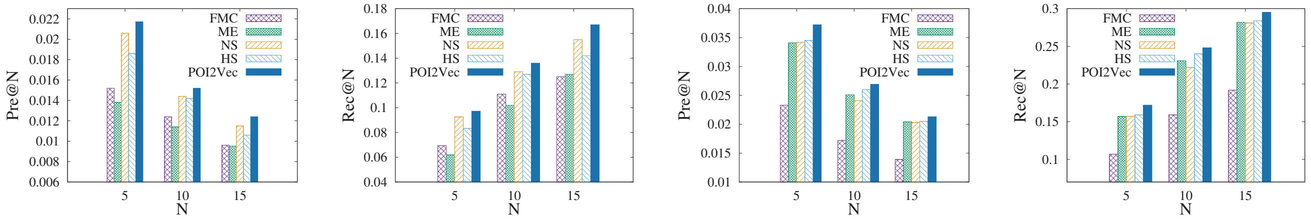
Latent Representation for Sequential Transition

To evaluate the quality of latent representation based on sequential POIs, we compare our proposed algorithm **POI2Vec** with 4 baselines in the task of predicting next POIs. (1) **FMC**: the factorized Markov chain model (Rendle, Freudenthaler, and Schmidt-Thieme 2010), which utilizes the Matrix Factorization to model the sequential transition. (2) **ME**: Metric Embedding model (Feng et al. 2015), which projects every POI into one object in a latent Euclidean space. (3) **NS**: the negative sampling technique for word2vec (Mikolov and Dean 2013), which is used to model POI sequences (Liu, Liu, and Li 2016). (4) **HS**: conventional hierarchical softmax with Huffman tree (Mikolov and Dean 2013). Given a user and his current POI, we utilize the subsequent POIs in the next 6 hours as the ground truth. Following the work (Feng et al. 2015), we use two metrics: Pre@N and Rec@N.

We compare the results of various embedding methods in Figure 3. The latent representation models (NS, HS and POI2vec) perform better than factorization model (FMC) and Metric Embedding model (ME). This indicates that latent representation methods are effective in modeling check-in sequences. Our POI2vec consistently performs better than HS, and this demonstrates that the proposed binary tree structure is better than the conventional Huffman tree. For example, POI2Vec outperforms HS by 16% and 9% on Foursquare and Gowalla, respectively, in terms of Pre@5. This is because POI2Vec incorporates the geographical influence into the binary tree. Overall, POI2Vec outperforms the other benchmark algorithms, which implies that POI2Vec can effectively learn the representations of POIs.

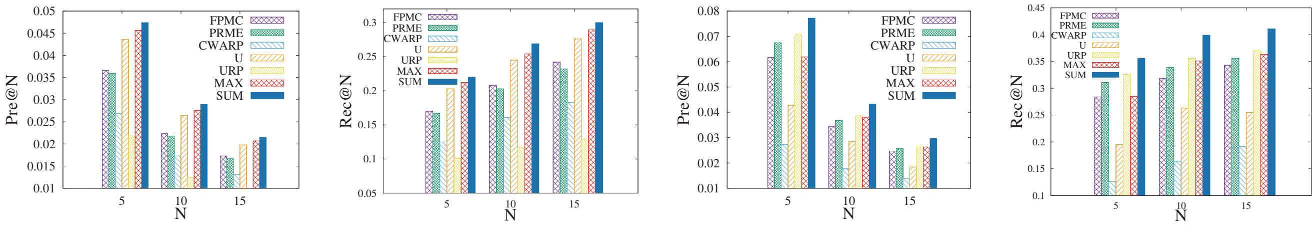
Future Visitor Prediction

To evaluate the performance of predicting future visitors, we compare our methods with 3 state-of-the-art baselines. (1) **FPMC**: factorizing personalized Markov chains, which



(a) Precision on Foursquare (b) Recall on Foursquare (c) Precision on Gowalla (d) Recall on Gowalla

Figure 3: The experimental results for sequential transition.



(a) Precision on Foursquare (b) Recall on Foursquare (c) Precision on Gowalla (d) Recall on Gowalla

Figure 4: The experimental results of predicting future visitors.

linearly combines the user preference and Markov transition (Cheng et al. 2013). (2) **PRME**: personalized ranking metric embedding, which linearly fuses the user preference and Markov transition (Feng et al. 2015). (3) **CWRAP**: which explores the context of locations to model user preference. In our methods, we learn the latent representations of users and POIs by using the POI2Vec method. Then we investigate 4 approaches to predict future visitors for POIs. (1) **U**: We only utilize user preference to predict potential visitors. (2) **URP**: We only consider users with recent positions. (3) **MAX**: Max aggregation function is used in Eq. (10). (4) **SUM**: We utilize the Sum aggregation function to integrate the user preference and sequential influence. Given a target POI and a time point, we utilize the users who visit this POI in the next τ hours as the ground truth. We use Pre@N and Rec@N as the metrics.

Figure 4 shows the experimental results. SUM achieves better results than FPMC and PRME. For instance, SUM outperforms FPMC and PRME by 29% and 32%, respectively, in terms of Pre@5 on Foursquare dataset. Note that all the three methods utilize the same linear aggregation. The experimental result demonstrates that the quality of the representation learned by POI2Vec is better than that by the others. U preforms better than CWRAP, both of which utilize the user preference to predict the potential visitors. This also demonstrates that POI2Vec generates better users' representations than CWRAP. CWARP learns user representation and POI representation separately: It first learns the POI representation, and the user representation is learned when POI representation is fixed. The result shows that it is more reasonable to jointly model the user preference and sequential transition. The performance of U is not as good as that of SUM, since it does not exploit users' recent positions. URP is worse than SUM, because it only considers the users with recent positions. This result shows that it is useful to consider both users with recent positions and users without recent positions. SUM outperforms MAX, and this indicates that combining user preference and sequential transition helps to predict potential users.

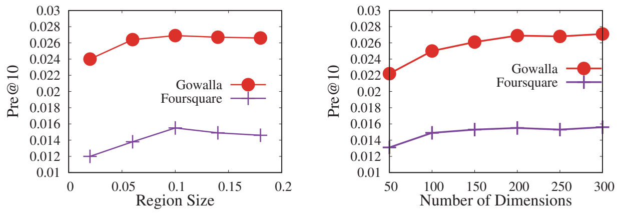
Effects of Parameters

Effect of region size To investigate the effect of region size threshold θ, we show the precision results in Figure 5(a). We observe that the performance first increases as we enlarge the region threshold, and then it decreases slightly. Small θ means that only very close POIs are clustered into the same region, which fails to capture the relationships with other POIs. The best performance is obtained at $\theta = 0.1$, which is about $11km$ on the earth. Hence, we set $\theta = 0.1$ in other experiments.

Effect of number of dimensions This experiment is to evaluate the effect of the number of dimensions D. As shown in Figure 5(b), the performance improves with the increase of D. We set $D = 200$ empirically in our experiments by considering the trade-off between effectiveness and efficiency.

Conclusions and Future Work

In this paper, we consider the problem of predicting the potential users who will visit a given POI in the near future. We propose a new latent representation model POI2Vec, which incorporates the geographical influence of POIs in learning latent representations. We further develop a method to jointly model POI sequential transition and user preference. Experiments on two datasets demonstrate that our algorithm significantly outperforms the state-of-the-art methods.



(a) Region size threshold θ (b) Number of dimensions D

Figure 5: Effect of parameters.

Several interesting research problems exist for further exploration. First, users' movements are influenced by many factors and we can consider other information such as the temporal influence for the visitor prediction problem. Second, our method of building binary tree is not tied for spatial items and can be adopted for other applications, such as using the taxonomy of items for product recommendation.

Acknowledgements

This work is supported in part by a Singapore MOE Tier-2 Grant (MOE-2016-T2-1-137). Shanshan Feng would like to acknowledge the Ph.D. grant from Advanced Environmental Biotechnology Centre, Nanyang Environment and Water Research Institute, Interdisciplinary Graduate School, Nanyang Technological University, Singapore.

References

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

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

Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*, 1082–1090.

Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new POI recommendation. In *IJCAI*, 2069–2075.

He, J.; Li, X.; Liao, L.; Song, D.; and Cheung, W. K. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*, 137–143.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

Li, X.; Cong, G.; Li, X.-L.; Pham, T.-A. N.; and Krishnaswamy, S. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, 433–442.

Li, H.; Ge, Y.; and Zhu, H. 2016. Point-of-interest recommendations: Learning potential check-ins from friends. In *SIGKDD*, 975–984.

Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; and Rui, Y. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *SIGKDD*, 831–840.

Liu, X.; Liu, Y.; Aberer, K.; and Miao, C. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *CIKM*, 733–738.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 194–200.

Liu, X.; Liu, Y.; and Li, X. 2016. Exploring the context of locations for personalized location recommendations. In *IJCAI*, 1188–1194.

Mikolov, T., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In *NIPS*, 1081–1088.

Morin, F., and Bengio, Y. 2005. Hierarchical probabilistic neural network language model. In *AISTATS*, 246–252.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*, 811–820.

Rong, X. 2014. Word2vec parameter learning explained. In *arXiv preprint arXiv:1411.2738*.

Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; and Cheng, X. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 403–412.

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

Ye, J.; Zhu, Z.; and Cheng, H. 2013. Whats your next move: User activity prediction in location-based social networks. In *SDM*, 171–179.

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

Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; and Thalmann, N. M. 2013b. Who, where, when and what: discover spatio-temporal topics for twitter users. In *SIGKDD*, 605–613.

Zhang, W., and Wang, J. 2015. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In *CIKM*, 1221–1230.

Zhang, J.-D.; Chow, C.-Y.; and Li, Y. 2014. LORE: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*, 103–112.

Zhao, K.; Cong, G.; Yuan, Q.; and Zhu, K. Q. 2015. SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *ICDE*, 675–686.

Zhao, S.; Zhao, T.; Yang, H.; Lyu, M. R.; and King, I. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*, 315–322.