

On successive point-of-interest recommendation

Yi-Shu Lu¹ · Wen-Yueh Shih¹ · Hung-Yi Gau¹ ·
Kuan-Chieh Chung¹ · Jiun-Long Huang¹ 

Received: 6 December 2017 / Revised: 28 March 2018 / Accepted: 24 May 2018 /
Published online: 23 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract With the increasing popularity of location-based social networks (LBSNs), users are able to share the Point-of-Interests (POIs) they visited by check-ins. By analyzing the users' historical check-in records, POI recommendation can help users get better visiting experience by recommending POIs which users may be interested in. Although recent successive POI recommendation methods consider geographical influence by measuring the distances among POIs, most of them ignore the influence of the regions where the POIs are located. Therefore, we propose in this paper two models to tackle the problem of successive POI recommendation. First, a feature-based successive POI recommendation method, named UGSE-LR, is proposed to take the influence of regions, named regional influence, into consideration when recommending POIs. UGSE-LR first splits an area into grids for estimating regional influence. Then, UGSE-LR applies Edge-weighted Personalized PageRank (EdgePPR) for modeling the successive transitions among POIs. Finally, UGSE-LR fuses user preference, regional influence and successive transition influence into a unified recommendation framework. In addition, with the aid of Recurrent Neural Network (RNN), we propose a latent-factor based successive POI recommendation method, named PEU-RNN, to integrate the sequential visits of POIs and user preference to recommend POIs. First, PEU-RNN adopts the word embedding technique to transform each POI into a latent vector. Then, RNN is used to recommend the POIs depend on the users' historical check-in records. Experimental results on two real LBSN datasets show that our methods are more accurate than the state-of-the-art successive POI recommendation methods in terms of precision and recall. In addition, experimental results also show that PEU-RNN is suitable for

This article belongs to the Topical Collection: *Special Issue on Social Media and Interactive Technologies*

Guest Editors: Timothy K. Shih, Lin Hui, Somchoke Ruengittinun, and Qing Li

✉ Jiun-Long Huang
jlhuang@cs.nctu.edu.tw

¹ Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

the datasets with many check-in records, while UGSE-LR is suitable for the datasets with moderate check-in records.

Keywords Successive POI recommendation · Recommendation · Location-based social network

1 Introduction

Location-based social networks (LBSNs) provide check-in services for users to share their visiting experiences to their friends. As shown in Figure 1, an LBSN is a combination of users, check-in timestamps and point-of-interests (POIs). People usually check in at a POI with a timestamp and then share their check-in records to their friends. Therefore, we can collect a user's friendship, check-in timestamps, coordinates and trajectories of POIs from their historical check-in records.

POI recommendation [2, 3, 10, 20, 26, 27, 32, 34–36] is a widely studied topic in recent years. POI recommendation is able to improve the user experience of the users in LBSNs by recommending some POIs which the users may be interested in through mining user's historical check-in records. Different from traditional POI recommendation, successive POI recommendation [4, 8, 12, 25, 38, 39] has to recommend a POI which the user may visit in the near future. Moreover, a user's intention for visiting one POI may be also influenced by the prior POI(s) that the user just visited. For example, users are usually willing to visit bars after visiting restaurants. Thus, successive POI recommendation has to take (1) the distances among the POIs and the user's current location and (2) the transitions among POIs (named *successive transition influence*) into consideration when performing recommendation.

Most POI recommendation methods tend to recommend the POIs that are close to the user's current location. In practice, the region where a POI is located usually has great influence on whether the user will check in at the POI. For example, if a new POI is in a region consisting of several famous POIs, it is easy for the new POI to attract check-ins since many users will check in at some POIs nearby. In view of this, we propose in this paper a feature-based successive POI recommendation method, named UGSE-LR, to take the influence of the regions where POIs are located (called *regional influence* in this

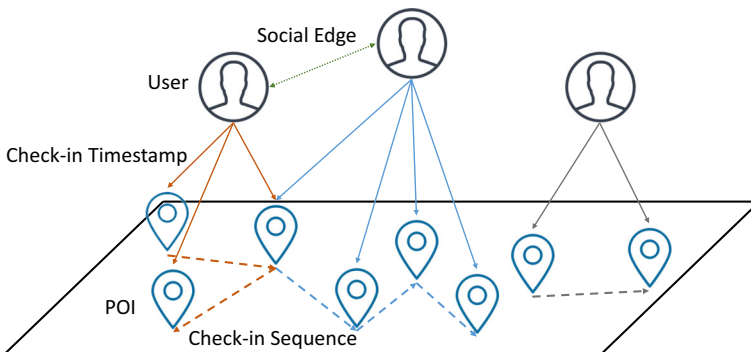


Figure 1 The architecture of LBSNs

paper) into consideration. Specifically, when recommending POIs to a user u , UGSE-LR first uses user-based collaborative filtering (User-based CF) [13] to find the users of check-in records similar to u and calculates the score of *user preference* of each POI with respect to u according to the historical check-in records of the users similar to u . Then, UGSE-LR divides the space into several grid cells and calculates the score of regional influence of each POI. UGSE-LR builds a reduced POI-to-POI transition graph according to user u 's historical check-in records and applies Edge-weighted Personalized PageRank [31] to calculate the score of the successive transition influence for each POI in the reduced POI-to-POI transition graph. Finally, UGSE-LR integrates the scores of user preference, regional influence and successive transition influence into the overall scores and recommends top- N POIs to user u .

In addition, due to the recent success of deep learning techniques in some areas such as natural language processing (NLP), we also proposed a latent factor-based successive POI recommendation method, named PEU-RNN, to utilize the word embedding technique and Recurrent Neural Network (RNN) for POI recommendation. It is obvious that the successive POI recommendation can be considered as a sequence prediction problem, RNN, which is successful in handling sequential data, is adopted in this paper to predict the users' future POI visits. However, traditional encoding methods such as one-hot encoding, cannot contain implicit relations between the prior and the next visited POIs. Therefore, PEU-RNN first utilizes the one of the word embedding techniques proposed in [24], named Continuous Bag-of-Word (CBOW), to encode the POIs and users according to historical check-in records. CBOW is able to give a unique latent vector for each user and POI based on user preference and the successive transitions among POIs so that these latent vectors are able to represent these relations. Then, based on the historical check-in data, PEU-RNN takes the latent vectors of POIs and users as the inputs of RNN to build a RNN-based model for predicting the top- N POIs of high probabilities to be the next visited POIs.

To evaluate the performance of our methods, several experiments are conducted on two real datasets. Experimental results show that when users' check-in records are sufficient, PEU-RNN outperforms the other successive POI recommendation methods in terms of precision and recall. On the other hand, when the users' check-in records are not sufficient, UGSE-LR is of the best performance. Such result agrees with the intuition that the RNN-based methods usually perform well in the cases with sufficient data.

The rest of this paper is organized as follows. We review some related work and formulate the problem of successive POI recommendation in Section 2. The proposed feature-based successive POI recommendation method, UGSE-LR, is described in Section 3 while the proposed latent factor-based successive POI recommendation method, PEU-RNN, is depicted in Section 4. Section 5 presents the performance comparisons among UGSE-LR, PEU-RNN and other prior methods. Finally, we conclude this paper in Section 6.

2 Preliminaries

2.1 Related work

2.1.1 Traditional point-of-interest recommendation

Ye et al. [32] raised two observations of geographic influence (known as spatial influence) from users' check-in data. First, users always check in at the places near their hometown

or offices. Second, if some places are very interesting but are far away, users still go there frequently. Based on these observations, they proposed a mixed recommendation system which combines user preference and geographic influence. Cheng et al. [3] utilized a matrix factorization method [16] to combine geographic influence and social influence. They had the following three observations. First, unlike the observations mentioned in [32], they found the POIs are usually distributed as Gaussian distributions around some POIs which are considered as centers. Second, although user preference is important, the distance can still affect the users' decisions. Third, although not a major factor, social influence can be observed in the datasets. About 9.6% check-in data is influenced by users' friends. Lian et al. [20] also considered geographic influence of neighboring regions, and then applied a matrix factorization method [16] to recommend POIs.

However, these approaches did not consider temporal influence. Thus, Yuan et al. proposed two methods in [34] and [35] to utilize temporal influence in POI recommendation. For example, if some people check in at the same time frequently, their preferences are very similar [34]. To recommend POIs to a target user, the method proposed in [34] first found the users of check-in behavior similar to the target user, and then recommended the POIs of these similar users' interests to the target user. They also proposed two hypotheses about time into their method proposed in [35]. First, users' interests change with time. Second, a user's behavior is similar in a specific time period. There are still many factors that can influence users' check-in behavior, such as check-in topics and check-in sequences. Wang et al. [27] used the topics of check-in records to find the interest distribution for a target user according to the type (i.e., a native person or a traveler) of the user, and then proposed a method to recommend POIs based on LDA-LCA [33]. In [36], Zhang et al. proposed to construct a transition graph to describe sequential influence of users' check-in records and score its impact by Edge-weighted Personalized PageRank (EdgePPR) [31].

2.1.2 Successive point-of-interest recommendation

In [4], Cheng et al. first formulated the problem of successive POI recommendation, and then incorporated Factoring Personalized Markov Chain (FPMC) [25] to solve the problem of successive POI recommendation. FPMC was originally used to solve the next-based recommendation problem. Cheng et al. revised FPMC by also considering geographic influence to recommend a user the POI that the user may be interested in. Since there are many POIs and people usually checked in at a few POIs in a time period, this phenomenon results in the sparse data problem. He et al. [12] not only used a Markov chain to infer users' preferences on POIs but also categorized users into different groups by their sequential check-in behavior such as the types of POIs and check-in time. Zhao et al. [39] argued that sequential check-in behavior is influenced by time. So, they used a time factor to measure the importance of sequential check-in behavior and proposed a successive POI recommendation method accordingly.

Feng et al. [8] is the first research to employ a metric embedding (ME) technique to model the relation among POIs in a potential latent space based on successive check-in records. They assumed that the related POIs should have short Euclidean distance in the latent space, and the latent space could be used to represent the transitions of each pair of POIs in a short time period according to the observation in [36]. Furthermore, Liu et al. [21] explored the POI's context according to the order of the POI visits. First, they

utilized the Skip-Gram model [24], which was originally designed for nature language processing (NLP), to explore the influence of the order of the POI visits. Then, based on the personalized recommendation model proposed in [30], they took the visiting frequency into the model. Although the Skip-gram [24] was originally a word representation, it has good performance in successive POI recommendation. Feng et al. [9] also utilized the word embedding technique on the POIs to retrieve the latent representation of each POI. Based on [24], they modified the Continuous Bag-of-Word (CBOW) model to fit the geographic influence by replacing the original Huffman tree with their proposed geographical binary tree. They also put user preference as the input of their model to represent each user. Finally, the probability of the next visited POI of a user can be calculated by using an aggregate function to fuse the latent vectors of users and POIs.

2.1.3 Neural network

Neural network (NN) has been widely used in natural language processing, such as language model [22], summarization [1], machine translation [5], and question answering [28]. Although the early models of neural network, such as artificial neural network (ANN) and multi-layer perceptron (MLP), were simple, [15] pointed out that even containing only one layer, the MLP can be still used to simulate any continuous function as long as having enough nonlinear units.

On the other hand, due to the significant advance of graphics processing unit (GPU), the modern neural network models, such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), can utilize a large number of hidden layers to extract the useful factors for prediction. CNN is currently widely used in image processing [17], face recognition [18], and handwriting recognition [7]. However, CNN does not perform well on sequential data. Instead, RNN can sequentially consider the pervious results into current stage, thereby making RNN capable of "memory." Therefore, RNN is usually suitable for the sequential data, such as sequential click prediction [37], language model [23], and speech recognition [11]. However, RNN still has some problems such as exploding gradients or vanishing gradients. When the sequence is too long, the prior results may not be able to be kept to the posterior input. To tackle these issues, two variants of RNN, named Long Short-Term Memory (LSTM) [14] and Gated Recurrent Unit (GRU) [6], were proposed. Due to the characteristic that the POI check-in records usually contain long sequential check-in sequences, we adopt LSTM as the prediction model in our latent factor-based successive POI recommendation method PEU-RNN.

2.2 Problem definition

Similar to [4], the problem of successive POI recommendation can be formulated as follows. Let \mathcal{U} and \mathcal{L} be the sets of users and POIs, respectively. Also let \mathcal{L}_u be the set of POIs that a user u has visited. Given a query, $q(u, l_c, t_c)$ where $u \in \mathcal{U}$ is the target user, $l_c \in \mathcal{L}_u$ is the POI where u is located, and t_c is current time, the problem of successive POI recommendation is to recommend N unvisited POIs $R_{u,N}$ to user u where each POI l in $R_{u,N}$ satisfies the following conditions.

- The distance between l and l_c is shorter than or equal to a distance threshold d .
- User u is likely to check in at l within time period $[t_c, t_c + \tau]$.

Table 1 Statistics of the datasets

Dataset	Brightkite	Gowalla
#Users	3,178	6,415
#Locations	6,792	44,938
#Check-ins	291,655	819,283
#Edges	45,308	73,410
Periods	Apr. 2008 - Oct. 2010	Feb. 2009 - Oct. 2010

2.3 Dataset descriptions

Two real datasets, Gowalla and Brightkite, are used in this paper and they are collected by Stanford Network Analysis Project¹ [19]. Since some check-in sequences are of insufficient information (e.g., the users of few check-in records), the check-in sequences of these inactive users and unpopular POIs are removed from the datasets due to the reason that these noises may severely affect the performance of recommendation. In this paper, the data satisfying one of the following conditions are filtered out.

- The users of less than 80 check-in records
- The POIs of less than 5 users' check-in records
- The users of less than 5 friends

After data cleaning, the statistics of two real datasets are listed in Table 1. According to the prior study [36], the successive check-in POIs are of high probability to be affected by the check-in POIs in the previous time period. Figure 2 shows the cumulative distribution functions of the distributions of the time differences between two successive check-in records in the Gowalla and Brightkite datasets. In the Gowalla dataset, about 52% successive check-ins happened within 6 hours, and over 72% data did within 24 hours. In the Brightkite dataset, about 40% successive check-ins happened within 6 hours, and over 60% did in a single day. Thus, we can observe that the Brightkite dataset is sparser in time domain than the Gowalla dataset.

In addition to time difference, distance difference is also an important factor to influence users' check-in behavior. Figure 3 shows the cumulative distribution functions of the distributions of the distance differences between two successive check-in records within 6 hours. From the result, over 90% and 50% successive check-ins happened within 15 km in the Gowalla and the Brightkite datasets, respectively. It's obvious that the Brightkite dataset is also sparser in spatial domain than the Gowalla dataset.

3 UGSE-LR: the proposed feature-based successive POI recommendation method

3.1 Architecture

The architecture of the proposed feature-based successive POI recommendation method, named UGSE-LR, is shown in Figure 4. UGSE-LR performs successive POI recommendation according to three properties, namely user preference, regional influence and successive

¹Stanford Network Analysis Project (SNAP): <http://snap.stanford.edu/data/index.html>.

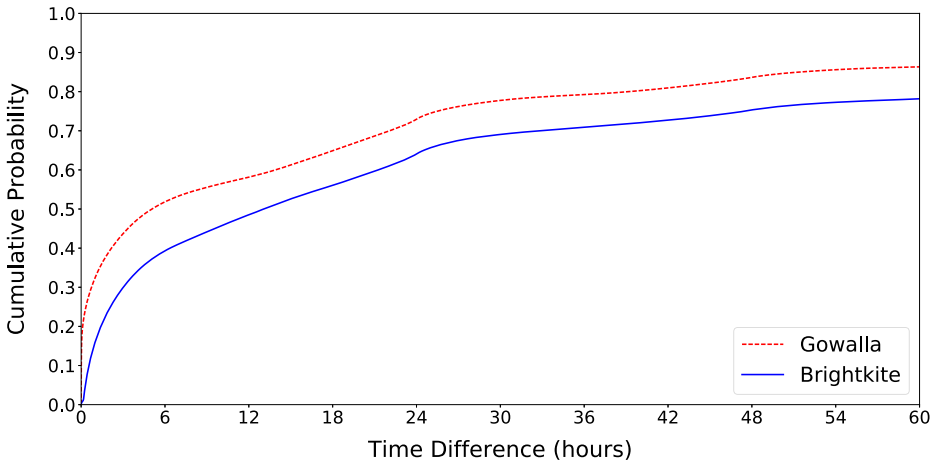


Figure 2 The cumulative distribution functions of the distributions of the time differences between two successive check-in records

transition influence. In calculating the score of user preference, UGSE-LR finds the set of users, say U' , whose historical check-in records are similar to the target user u 's, and calculates the user preference scores of POIs with respect to u according to the historical check-in records of the users in U' . In calculating the score of regional influence, UGSE-LR divides the space into several disjoint grid cells, and calculates the score of regional influence of each grid cell based on the check-in records in the grid cell. Edge-weighted Personalized PageRank [31] is adopted to calculate the successive transition influence scores of POIs based on u 's historical check-in records. Finally, UGSE-LR calculates the overall scores of POIs and recommends top- N POIs to user u .

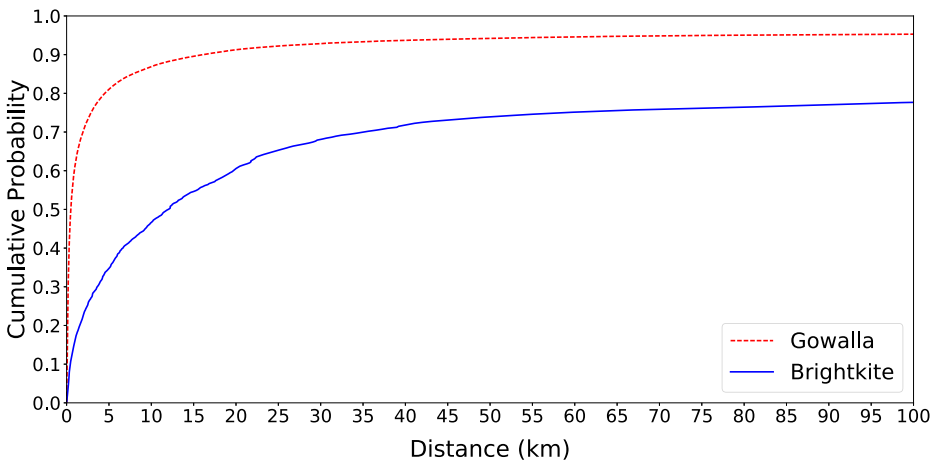


Figure 3 The cumulative distribution functions of the distributions of the distance differences between two successive check-in records within 6 hours

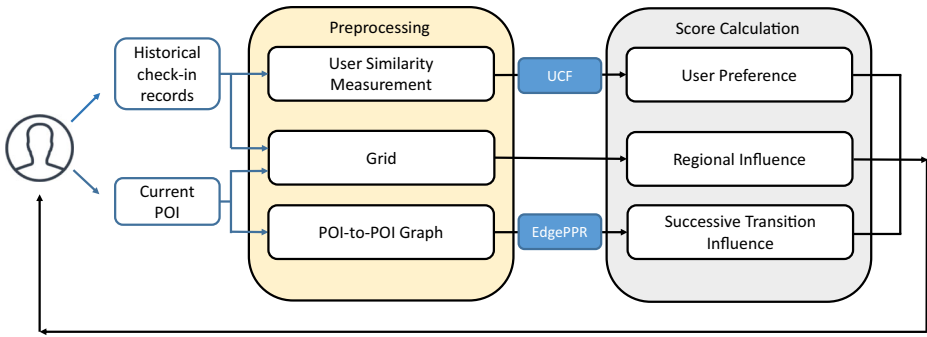


Figure 4 The architecture of UGSE-LR

3.2 User preference

We use user-based collaborative filtering (User-based CF) [32] to calculate the user preference scores of POIs with respect to the target user u . Consider two users u and v . Let $c_{u,l} = 1$ indicates that user u had checked in at POI l . Otherwise, $c_{u,l}$ is set to zero. We use the following equation to measure the similarity of check-in behavior of user u and user v based on u 's and v 's historical check-in records.

$$w_{u,v} = \frac{\sum_{\forall l \in \mathcal{L}} c_{u,l} \cdot c_{v,l}}{\sqrt{\sum_{\forall l \in \mathcal{L}} c_{u,l}^2} \sqrt{\sum_{\forall l \in \mathcal{L}} c_{v,l}^2}} \tag{1}$$

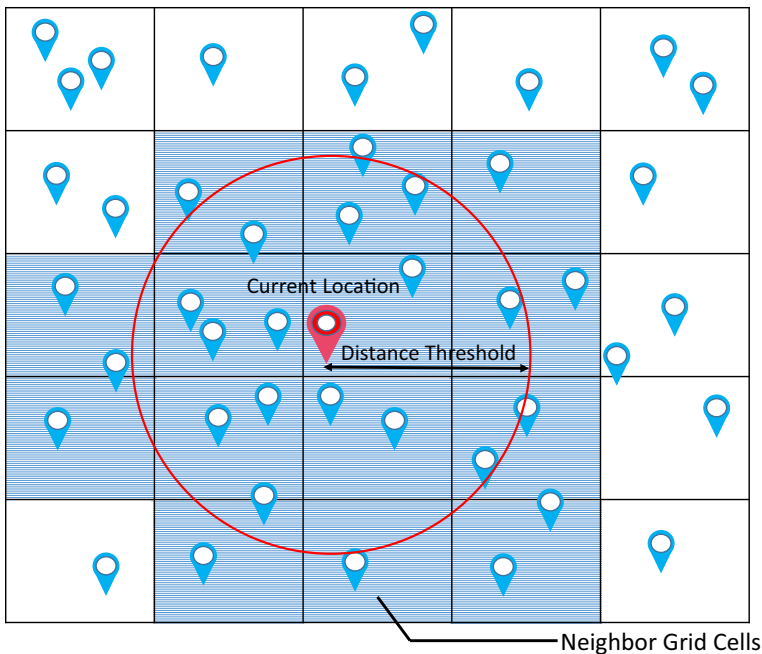


Figure 5 An example grid

Let U' be the set of users of check-in behavior similar to u . The user preference score of a POI l with respect to u can be calculated by the following equation.

$$p_{u,l}^{user} = \frac{\sum_{\forall v \in U'} w_{u,v} \cdot c_{v,l}}{\sum_{\forall v \in U'} w_{u,v}} \tag{2}$$

3.3 Regional influence

As mentioned in Section 1, different from traditional POI recommendation, successive POI recommendation should consider distances among POIs and user’s current location. According to Figures 2 and 3, the next visiting POIs are usually located close to the current locations of users, meaning the users seldom check in at POIs which are of interest but far away. Therefore, we only consider the POIs close a user as the candidates to be recommended to the user. As shown in Figure 5, the space is divided into several grid cells. Let l_c be user u ’s current location and d be the distance threshold in UGSE-LR. The grid cells overlapping the circle with center l_c and radius d are called the *neighbor grid cells* of u . The idea is that if a grid cell g_i is popular (i.e., of many check-in records), user u is likely to check in at a/some POI(s) in g_i . Let $Checkins(g_i)$ be the summation of the numbers of check-in records at all POIs in a grid cell g_i . We use the following equation to measure the popularity of a grid cell g_i ,

$$g_i^p = \frac{Checkins(g_i)}{\sum_{\forall g \in G_s} Checkins(g)}, \tag{3}$$

where G_s is the set of neighbor grid cells of u .

When user u had many check-ins at some POIs in a grid cell g_i , it is possible that grid cell g_i is in user u ’s favorite area and user u is of high likelihood to check in at other POI(s) in grid cell g_i . Let $Checkins(g_i, u)$ be the number of check-in records of user u at all POIs in grid cell g_i . Then, we use the following equation to measure such influence.

$$g_i^u = \frac{Checkins(g_i, u)}{\sum_{\forall g \in G_s} Checkins(g, u)} \tag{4}$$

As mentioned in [32], a user tends to check in at the POIs close to the user’s current location. Thus, we argue that a user tends to check in at the POIs in the grid cell where the user is current located. Therefore, we devise the following equation to measure such influence.

$$g_i^c = \begin{cases} 1 & \text{if the user’s current location is in } g_i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Then, we combine the above three properties into the score of a grid cell g_i by the linear function shown in (6)

$$GridScore_{g_i} = \alpha g_i^p + \beta g_i^u + \gamma g_i^c, \tag{6}$$

where α , β and γ are weights satisfying the following constraints.

$$\begin{aligned} 0 &\leq \alpha, \beta, \gamma \leq 1 \\ \alpha + \beta + \gamma &= 1 \end{aligned} \tag{7}$$

Finally, the score of regional influence of a POI l with respect to u can be calculated as

$$p_{u,l}^{reg} = \frac{GridScore_{g_i}}{\sum_{\forall g \in G_s} GridScore_g}, \tag{8}$$

where g_i is the grid cell where POI l is located.

3.4 Successive transition influence

In this subsection, we use a POI-to-POI transition graph to model the relationship of successive check-ins. Let (l, t) indicate that a user had checked in at POI l at time t . Then, the POI-to-POI transition graph is defined as follows.

Definition 1 Consider a user u 's check-in records $(l_1, t_1), (l_2, t_2), \dots, (l_n, t_n)$ where $t_1 \leq t_2 \leq \dots \leq t_n$. We say that there exists a *successive transition* from POI l_i to l_{i+1} with respect to user u if $t_n - t_1 \leq \tau$.

Definition 2 The POI-to-POI transition graph of all users is a directed graph $G = (\mathcal{L}, E)$ where \mathcal{L} is the set of all POIs and E is the set of all successive transitions among POIs in \mathcal{L} . That is, there exists a directed edge (l_i, l_j) in E if there is a successive transition from l_i to l_j in all users' historical check-in records. The weight of an edge (l_i, l_j) is defined as

$$w_{l_i, l_j} = \frac{Transitions(l_i, l_j)}{\sum_{\forall l \in \mathcal{L}} Transitions(l_i, l)}, \tag{9}$$

where $Transitions(l_i, l_j)$ is the number of successive transitions from l_i to l_j in all users' historical check-in records.

Since only the POIs of distances to user u 's current location shorter than or equal to d are candidate POIs to be recommended, we then extract a subgraph of G , named the reduced POI-to-POI transition graph $G' = (\mathcal{L}', E')$ of u , by removing the POIs not in the neighbor grid cells of u from G . We then utilize Edge-weighted Personalized PageRank (EdgePPR) [31] to calculate the scores/importance of all nodes in the reduced POI-to-POI transition graph of u since EdgePPR is able to operate efficiently in a local computation manner by the model reduction technique. The successive transition influence of u to l is defined as the normalized EdgePPR score of l and can be obtained by the following equation.

$$P_{u,l}^{suc} = \frac{EdgePPR(G', l)}{\sum_{\forall l' \in \mathcal{L}'} EdgePPR(G', l')} \tag{10}$$

Then, we use min-max normalization to normalize user preference, regional influence and successive transition influence by the following equations

$$\begin{aligned} S_{u,l}^{user} &= \frac{p_{u,l}^{user} - m_u^{user}}{M_u^{user} - m_u^{user}}, \\ S_{u,l}^{reg} &= \frac{p_{u,l}^{reg} - m_u^{reg}}{M_u^{reg} - m_u^{reg}}, \\ S_{u,l}^{suc} &= \frac{p_{u,l}^{suc} - m_u^{suc}}{M_u^{suc} - m_u^{suc}}, \end{aligned} \tag{11}$$

where M_u^{user}/m_u^{user} , M_u^{reg}/m_u^{reg} and M_u^{suc}/m_u^{suc} are the maximum/minimum scores of user preference, regional influence and successive transition influence, respectively, of all POIs in \mathcal{L}' . Finally, the overall score of user u to POI l is determined by the following equation.

$$S_{u,l} = \delta \times S_{u,l}^{user} + \epsilon \times S_{u,l}^{reg} + \zeta \times S_{u,l}^{suc}, \tag{12}$$

where δ, ϵ, ζ are weights satisfying the following constraints.

$$\begin{aligned} 0 &\leq \delta, \epsilon, \zeta \leq 1 \\ \delta + \epsilon + \zeta &= 1 \end{aligned} \tag{13}$$

4 PEU-RNN: the proposed latent factor-based successive POI recommendation method

4.1 Architecture

Figure 6 shows the architecture of the proposed latent factor-based successive POI recommendation PEU-RNN. PEU-RNN contains two stages. First, according to [24], the latent vectors of all users and POIs can be retrieved from the historical check-in records which contain the users' visiting preferences and the influence of POIs. In the first stage, two steps are performed to find the latent vectors for all users and POIs. First, the successive check-in records of all users within a specific time period are used to find the transition relations among POIs by distributed representation (Word2Vec is one of the popular distributed representations). Then, with the same skill, the historical check-in records of each user is adopted to find the user's latent vector from his or her sequence of POI visits. After building the latent vectors to represent each user and POI, the historical check-in records and the obtained latent vectors are utilized in the second stage to build a RNN model for POI recommendation. The softmax function is adopted as the activation function to estimate the next visiting probabilities of all POIs based on the current and visited POIs.

4.2 Latent vector establishment

The successive transitions among POIs are important for POI recommendation. However, the POI-to-POI transition graph adopted in UGSE-LR (described in Section 3.4) is too complicated, thereby not suitable for RNN. Based on the significant success of RNN in NLP, each POI-to-POI transition should be transformed into a latent vector based on a vector encoding method. The popular vector encoding methods are one-hot encoding and distributed representation. It is difficult to use one-hot encoding to model successive transitions among POIs. When one-hot encoding is used, successive transitions among POIs will be transformed into high dimensional vectors, thereby introducing huge computation overhead.

On the other hand, using the distributed representation, such as word2vec [24], is able to control the computation overhead by reducing the data dimensionality to an acceptable

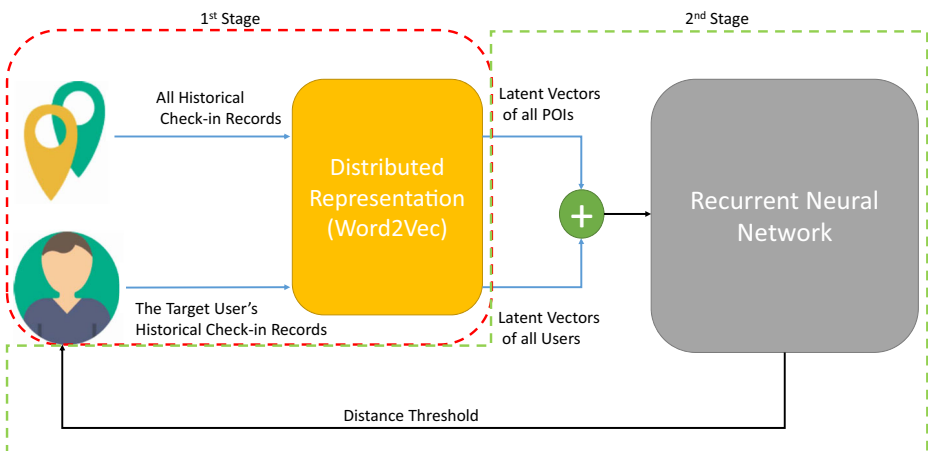


Figure 6 The architecture of PEU-RNN

range. In addition, the distributed representation also takes the relation between prior and posterior POI visits into consideration. Thus, we argue that the distributed representation is more suitable for POIs than one-hot encoding, and we adopt the word2vec technique [24] to model the successive transitions among POIs. The word2vec technique consists of two models, namely Continuous-Bag-of-Word (CBOW) and Skip-gram. Both CBOW and Skip-gram adopt Huffman trees to model the context relation. CBOW can be used to predict the next POIs based on the current POIs, while Skip-gram can be used to predict the previous POIs and the next POIs based on the current POIs. Due to the characteristic of successive POI recommendation, CBOW is adopted in PEU-RNN to model the successive transitions among POIs.

The detailed procedures are as follows. Each user’s check-in records within the time interval τ is gathered as one *check-in sequence*, $\mathcal{L}_i = (l_1, \dots, l_m | \tau \geq (t_m - t_1))$, $\{l_j | 1 \leq j \leq m, l_j \in \mathcal{L}\}$. The \mathcal{L}_i contains the visiting sequences and frequencies of POIs, which can be used to build a Huffman tree in CBOW. As shown in Figure 7, each inner node of the Huffman tree can be considered as a binary classifier to decide which way to go and the leaf nodes are the POIs. The hierarchical softmax function is adopted in [24] to calculate the probability of the next visited POI, as shown in (14).

$$P(l = l_O | h) = \prod_{j=1}^{|Path(l)|-1} \sigma([m(l, j + 1) = ch(m(l, j))] \cdot v'_{m(l, j)}{}^T h),$$

$$where \ [x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ -1 & \text{otherwise} \end{cases}, \ h = \frac{1}{Q} \sum_{q=1}^Q v_{l_q}.$$
(14)

The meaning of the notations in (14) is listed as follows.

- l_O represents the actual output POI.
- $Path(l)$ is the path from root to POI l .
- $m(l, j)$ means the j -th inner node on the path from root to POI l .
- $ch(m)$ indicates the left child of inner node m .
- $v'_{m(l, j)}$ is the vector representation of inner node $m(l, j)$.
- Q is the number of POIs in the check-in sequence.
- v_{l_q} is the input vector of POI l_q .
- $\sigma(\cdot)$ is the sigmoid function.

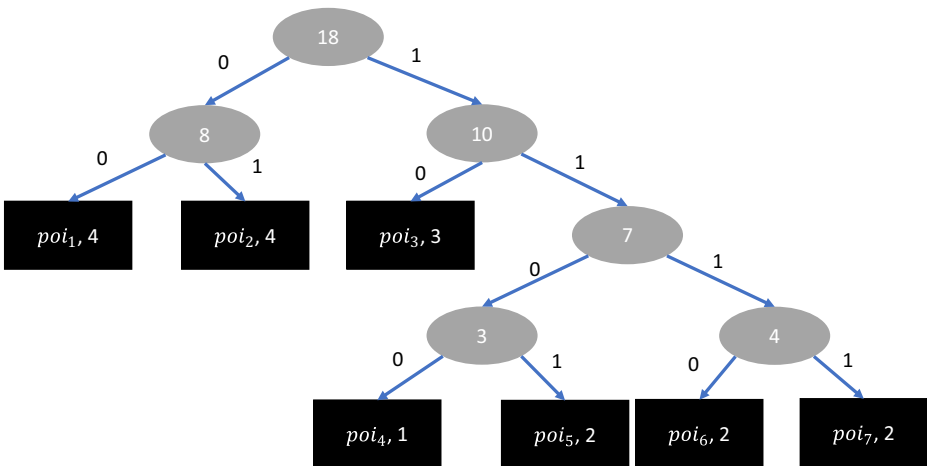


Figure 7 The illustration of Huffman Tree

– h is the aggregation of the previous visited POIs before l_O .

Based on the hierarchical softmax function, the objective function of CBOW is as (15), where V'_m is the coefficient set of all inner node in the Huffman tree, and V_l is the set of latent vectors of all POIs. After the optimization, the optimized \hat{V}'_m and \hat{V}_l can be derived.

$$\hat{V}'_m, \hat{V}_l = \arg \max_{V'_m, V_l} \prod_i P(l_{O,i} | h_i) \tag{15}$$

The users' latent vectors, $V_u = \{v_{u_1}, v_{u_2}, \dots, v_{u_{|U|}}\}$, can also be obtained by the similar method.

4.3 Visiting probability prediction

As mentioned, a user's check-in behavior is usually affected by user preference (indicated by users' latent vectors) and successive transition influence (embedded to the latent vectors of all POIs). To consider both factors, the latent vectors of all users and POIs, denoted as V_u and V_l , are combined together as the *willing vector*, V_w according to (16). Then the V_w is considered as the input x_t of the LSTM. After the softmax layer, the visiting probabilities of user u to all POIs are output by the LSTM.

$$V_w = V_l + V_u, V_w \in \mathcal{V}_w \tag{16}$$

4.4 POI recommendation

According to the observations in Figure 3, the users are of high probability to visit the POIs close to them. In other words, the recommendation method should put more weights on the nearby POIs to have more chances to match the user's visiting intention. Therefore, after the visiting probability of each POI has been estimated, distance threshold is applied to prune the POIs of the distances to the user's current location longer than d . The set of POIs for user u , denoted as \mathcal{R}_u , can be obtained by (17), where l_c is user u 's current location, $(l_q, P(l_q|u, l_c))$ is the result of the LSTM $LSTM(\cdot)$, and $P(l_q|u, l_c)$ is the probability of user u to visit POI l_q in time period $[t_c, t_c + \tau]$.

$$\mathcal{R}_u = \{(l_q, P(l_q|u, l_c)) | (l_q, P(l_q|u, l_c)) \in LSTM(V_w), \wedge distance(l_q, l_c) \leq d\} \tag{17}$$

Finally, the top- N POIs in \mathcal{R}_u , denoted as $R_{u,N}$, are recommended to user u .

5 Performance evaluation

5.1 Experimental setting

Two real check-in datasets, Gowalla and Brightkite [19], are used in our experiments. We filter out the POIs checked-in by less than 80 users and the users whose check-in records are less than five. After preprocessing, the statistics of the datasets are shown in Table 1.

Precision and recall, denoted as Precision@N and Recall@N, respectively, are adopted as the performance metrics since they are widely used to evaluate the performance of recommendation methods. Precision@N and Recall@N are formulated as

$$Precision@N = \frac{|\mathcal{R}_{u,N} \cap \mathcal{L}'_{u,l_c,t_c}|}{N} \tag{18}$$

Table 2 Comparison among the methods

Methods	User	Geographical influence		Successive
	Preference	Distance threshold	Regional influence	Transition
FPMC	✓			✓
FPMC-LR	✓	✓		✓
POI2VEC	✓		✓	✓
UGSE-LR	✓	✓	✓	✓
PEU-RNN	✓	✓		✓

$$Recall@N = \frac{|\mathcal{R}_{u,N} \cap \mathcal{L}'_{u,l_c,t_c}|}{|\mathcal{L}'_{u,l_c,t_c}|}, \tag{19}$$

where $\mathcal{R}_{u,N}$ is the set of recommended POIs which contains top- N POIs, l_c is the POI where user u is current located, and \mathcal{L}'_{u,l_c,t_c} is the set of new POIs visited by user u within the time period $[t_c, t_c + \tau]$ at POI l_c .

To evaluate the performance of the proposed methods, UGSE-LR and PEU-RNN, we also implemented the prior methods FPMC [25], FPMC-LR [4], and POI2VEC [9]. The characteristics of these recommendation methods are shown in Table 2. To conduct experiments, 70% of the datasets are used as the training data, while 10% are used as the validation data for parameter tuning. The remaining 20% datasets are used as the test data.

5.2 Parameter determination for UGSE-LR

5.2.1 Performance tuning on weights

In UGSE-LR, two sets of parameters, $\{\alpha, \beta, \gamma\}$ in (6) and $\{\delta, \epsilon, \zeta\}$ in (12), should be determined for calculating the regional influence and the overall scores of all POIs, respectively. Since each parameter set has the constraint that the summation of the weights in each parameters should equal to 1, only the values of two parameters in each parameter set are needed to be shown in the figures. The tuning results in two datasets are shown in Figures 8, 9, 10 and 11, and the final optimal values of the parameters for both datasets are listed in Table 3.

Figures 8 and 10, among the weights of regional influence, γ is of the highest value in both datasets, meaning that the current location is the main factor for a user to decide the

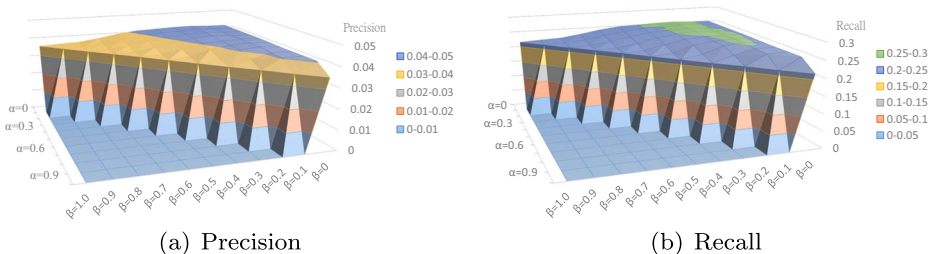


Figure 8 The parameter tuning of α, β and γ on the Gowalla dataset

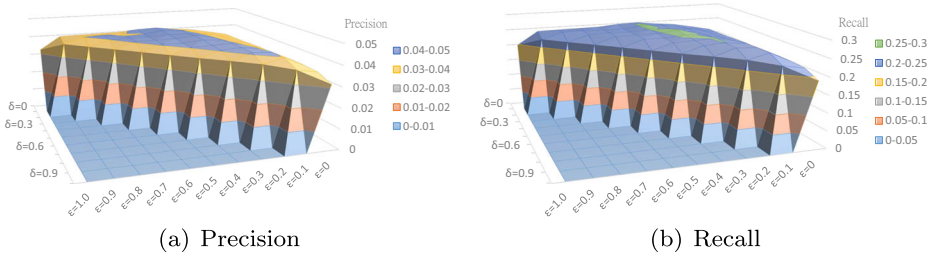


Figure 9 The parameter tuning δ , ϵ and ζ on the Gowalla dataset

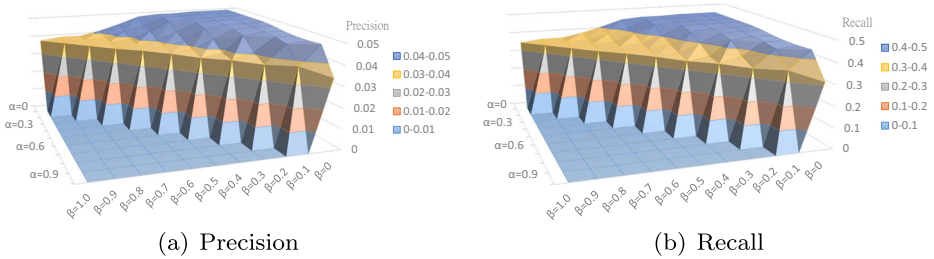


Figure 10 The parameter tuning of α , β and γ on the Brightkite dataset

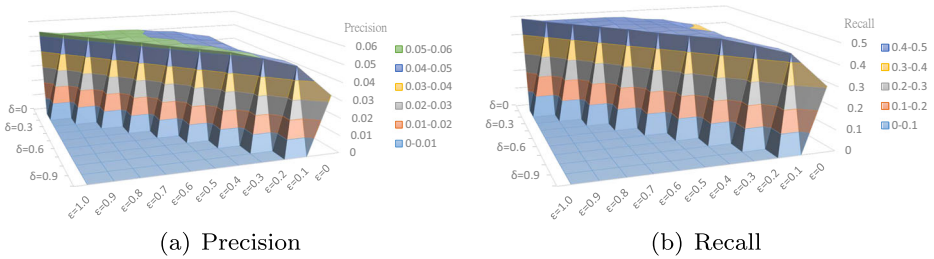


Figure 11 The parameter tuning of δ , ϵ and ζ on the Brightkite dataset

Table 3 Optimal values of the parameters of UGSE-LR

Dataset	α	β	γ	δ	ϵ	ζ
Gowalla	0.3	0.2	0.5	0.2	0.3	0.5
Brightkite	0.2	0.2	0.6	0.5	0.3	0.2

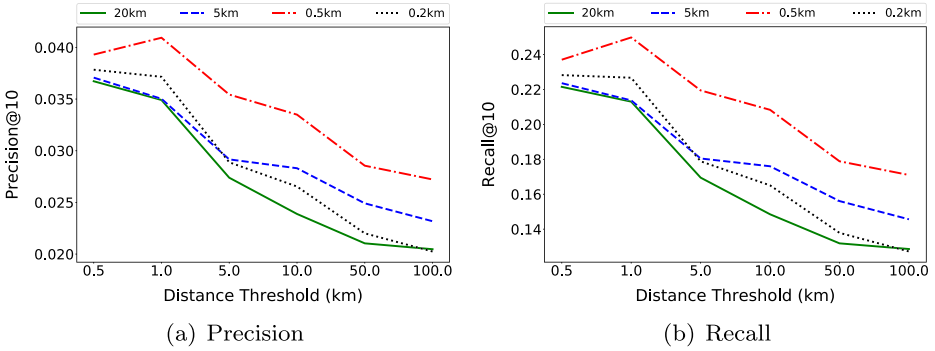


Figure 12 The effect of grid size and distance threshold on the Gowalla dataset

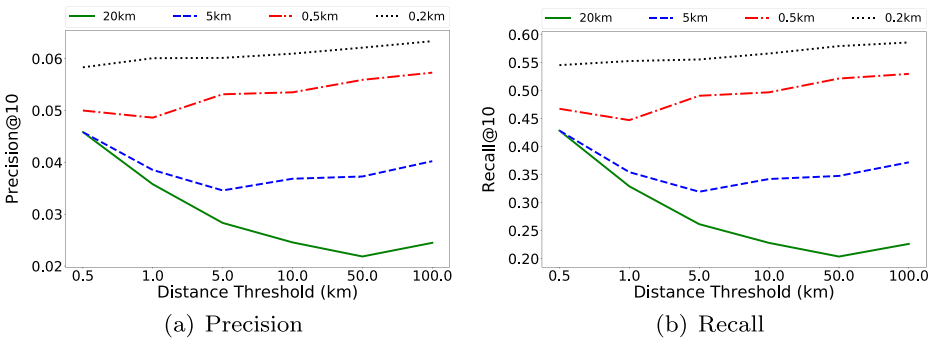


Figure 13 The effect of grid size and distance threshold on the Brightkite dataset

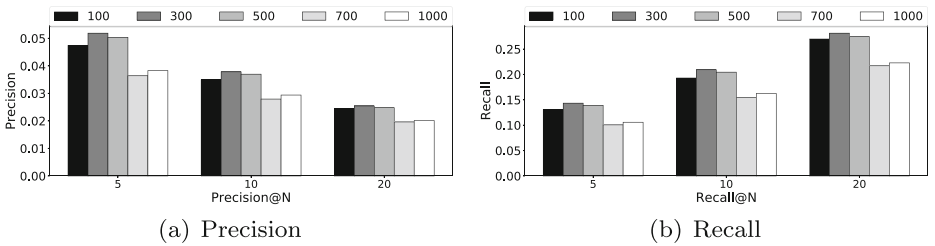


Figure 14 The performance on Gowalla with different number of units in one layer

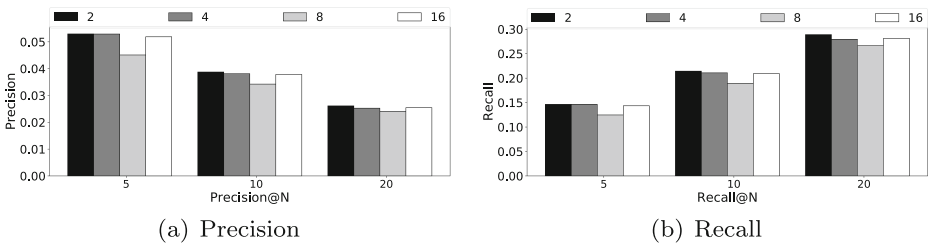


Figure 15 The performance on Gowalla with different number of layers

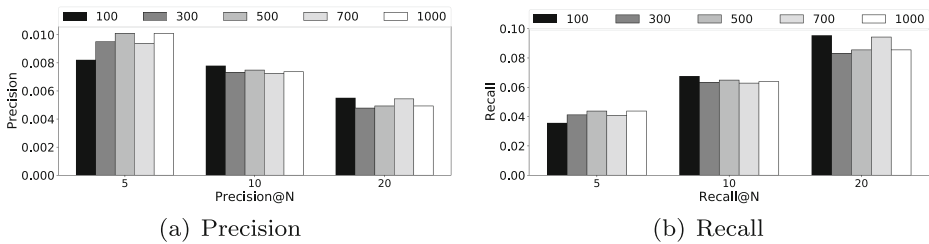


Figure 16 The performance on Brightkite with different number of units in one layer

next visited POI. As shown in Figures 9 and 11, among the elements in the overall scores (i.e., user preference, regional influence and successive transition influence), successive transition influence is the most important factor in the Gowalla dataset, while user preference is the key to influence the visiting intention in the Brightkite dataset. We argue that this difference is possibly caused by user behavior. In the Gowalla system, there is a trip recommender system,² making users tend to follow the trip advices. Therefore, successive transition influence takes an important role in the Gowalla dataset.

On the other hand, according to Figures 2 to 3, in the Brightkite dataset, the differences of distance and time between two successive check-ins are longer than those in the Gowalla dataset. Interestingly, the Brightkite system makes a user free to check in at any POI (even it is impossible for the user to visit the POI within a short time period in reality) [29]. Such mechanism makes user preference be the key factor in the Brightkite dataset.

5.2.2 Effect of grid size and distance threshold

In this subsection, we investigate the effect of grid size and distance threshold d on the precision and recall of UGSE-LR. The value of τ is set to two hours and the number of recommended POIs (i.e., N) is set to 10. The grid size is set to 0.2 km, 0.5 km, 5 km and 20 km while the distance threshold d is set to 0.5 km, 1 km, 2 km, 5 km, 10 km, 50 km and 100 km. The experimental results are shown in Figures 12 and 13.

We can observe from Figure 12 that the best performance of UGSE-LR is the case of setting distance threshold d to 1 km on the Gowalla dataset. When the value of distance threshold d increases, the performance of UGSE-LR decreases. This means that if distance threshold is set too large, UGSE-LR must consider more POIs as candidate POIs, making POI recommendation more challenging since a user usually moves to the POI near his or her current location. It is interesting in Figure 13 that in the Brightkite dataset, increasing distance threshold d results in better performance of UGSE-LR. We believe that it is because the Brightkite system enables a user to be able to check in at a POI without physically in the POI.

Grid size is important for regional influence. If the grid size is set too large, the number of POIs in each grid increases and may result in overestimating regional influence of each grid. On the other hand, if the grid size is set too small, the number of POIs in each grid decrease, thereby reducing the importance of regional influence. According to Figures 12 and 13, we set the grid size to 0.5 km in the Gowalla dataset and 0.2 km in the Brightkite dataset for better performance.

²<https://techcrunch.com/2010/01/28/gowalla-opens-trips-to-all-a-simple-way-to-organize-pub-crawls/>

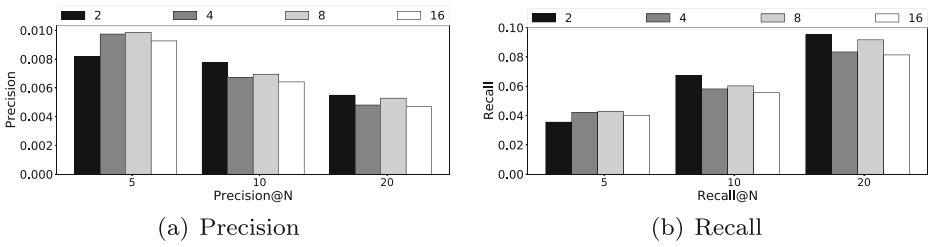


Figure 17 The performance on Brightkite with different number of layers

5.3 Parameter determination for PEU-RNN

In PEU-RNN, the LSTM is the basis model for POI recommendation. In order to achieve better performance, the number of units per layer and the number of layers, which are the hyper-parameters to build a neural network, should be fine-tuned in advance. The distance thresholds are set to 1 km and 5 km in the Gowalla and Brightkite datasets, respectively. To speed up the tuning, the number of epoch is set to a small number for tuning parameters. Figures 14 and 15 show the effects of the unit number and the layer number in the Gowalla dataset, while Figures 16 and 17 are the effects of the unit number and the layer number in the Brightkite dataset. The best number of units in one layer is 300 and 100 for the Gowalla dataset and the Brightkite dataset, respectively, while the optimal numbers of layers for the Gowalla dataset and the Brightkite dataset are both set to 2, respectively. The final optimal parameters for PEU-RNN are shown in Table 4.

5.4 Performance comparison

5.4.1 Effect of the number of recommended POIs

In this subsection, we investigate the effect of the number of recommended POIs (i.e., N). PEU-RNN adopts CBOW as the underlying word embedding technique, while PEU-RNNS uses Skip-gram to replace CBOW in PEU-RNN. The numbers of epoch for PEU-RNN and PEU-RNNS are set to 100 in the Gowalla dataset and 500 in the Brightkite dataset for better performance. The time constraints t is set to 3 hours in each dataset. Distance threshold d is set to 1 km in the Gowalla dataset and 5 km in the Brightkite dataset. For UGSE-LR, the grid size is set to 0.5 km in the Gowalla dataset and 0.2 km in the Brightkite dataset. The value of N is set to 5, 10, 15, 20, and 25, respectively. The experimental results are shown in Figures 18 and 19. We can see in Figure 18 that PEU-RNN is the best in most cases in the Gowalla dataset and PEU-RNNS is in the second place. As mentioned in Section 5.2.1, successive transition influence plays an important role in the Gowalla dataset. We believe that it is because RNN is able to extract successive transition influence from the Gowalla dataset. PEU-RNN outperforms PEU-RNNS in all cases, agreeing with our intuition mentioned in

Table 4 Optimal parameters of PER-RNN

Dataset	Number of units per layer	Number of layers
Gowalla	300	2
Brightkite	100	2

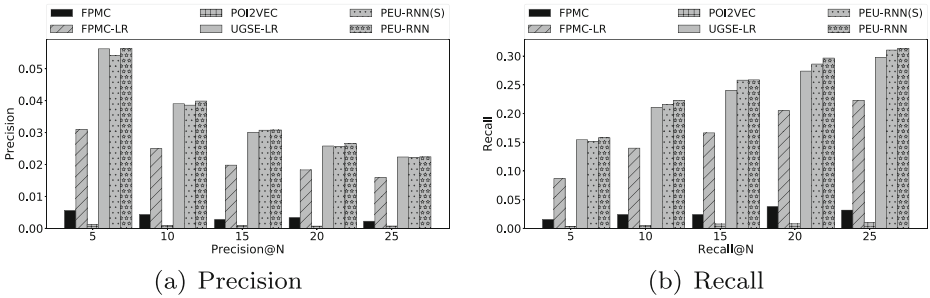


Figure 18 The result of Top-N recommendation in Gowalla

Section 4.2 that CBOW is more suitable for successive POI recommendation than Skip-gram. On the other hand, UGSE-LR also outperforms FPMC, FPMC-LR, and POI2VEC. The reason is that UGSE-LR not only considers the distances among POIs and the users but also incorporates the regional influence of the grids where the POIs are located.

However, in the Brightkite dataset, as shown in Figure 19, PEU-RNN and PEU-RNNS are of poor performance. In addition to the different characteristics of these two datasets mentioned in Section 5.2.1, the quantity and the quality of the datasets are also the matter of poor performance. Figure 20a and b show the cumulative distribution functions of the distributions of number of successive check-ins in check-in sequences when τ is set to 3 and 6 hours, respectively, on both datasets. The training data in the Brightkite dataset has 90% check-in sequence of only one check-in within 3 hours, but the Gowalla dataset has over 30% check-in sequences with more than one check-in. As mentioned in Section 5.2.1, in the Brightkite dataset, user preference is of the most importance and the importance of successive influence is low. The characteristic of the Brightkite dataset makes the strength of PEU-RNN and PEU-RNNS in extracting successive transition influence not significant in recommending POIs in the Brightkite dataset. In addition, the number of check-in sequences in the Brightkite dataset seems not sufficient for PEU-RNN and PEU-RNNS. The number of training check-in sequences in the Brightkite dataset is 6189 within 3 hours and 5688 check-in sequence within 6 hours. On the other hand, there are 17160 and 16791, respectively, check-in sequences in the Gowalla dataset within 3 and 6 hours, respectively. Due to insufficient check-in sequences in the Brightkite dataset, PEU-RNN and PEU-RNNS (deep learning based methods) does not perform well, agreeing with our intuition that deep learning methods requires sufficient data for training. On the other hand, UGSE-LR (a

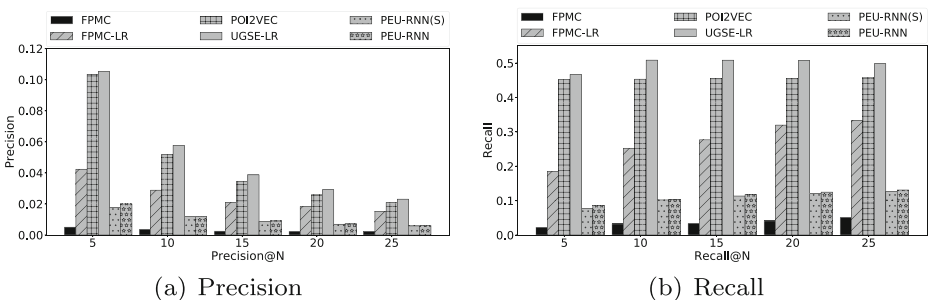


Figure 19 The result of Top-N recommendation in Brightkite

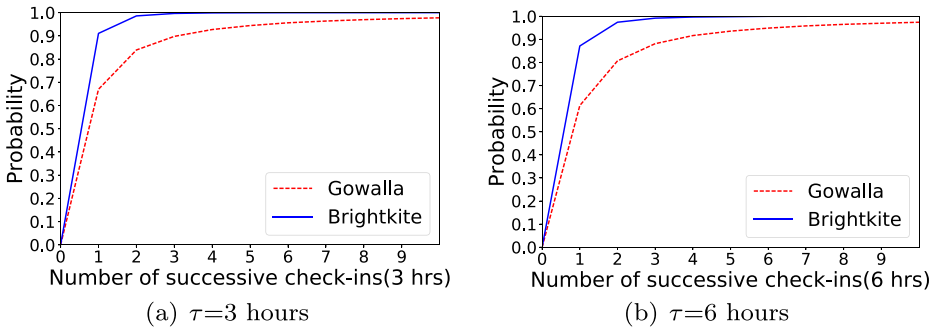


Figure 20 The cumulative distribution function of the number of check-ins within different hours

feature-based method) still outperforms the other methods in most cases due to the consideration of user preference, regional influence and successive transition influence together. Besides, it’s worth noting that the POI2VEC is of the poorest performance in the Gowalla dataset which conflicts with the experimental results in [9], and is of good performance in Brightkite dataset especially when N is small. We argue that the conflict experimental results in the Gowalla dataset is due to the scale of the Gowalla dataset used to conduct experiments. Specially, only a portion of the Gowalla dataset (only one city) is used in [9], while all the Gowalla dataset is used in our study. In the Brightkite dataset, when the number of recommended POIs is small, POI2VEC outperforms others, but gets worse when N gets larger. Since POI2VEC does not consider the distance constraint, it may recommend the POIs far away from users, when the number of recommended POIs gets larger. On the other hand, UGSE-LR can always perform well and stably due to the consideration of multiple factors.

5.4.2 Effect of time constraint τ

We now investigate the effect of the time constraint of the interval between successive check-ins (i.e., τ). The settings of all methods are the same as those in Section 5.4.1. The value of τ is set from 1 to 6 hour(s) and the experimental results are shown in Figures 21 and 22. We can observe that the precision and recall decrease as the time constraint τ increases in both datasets. When the time constraint τ increases, an user may move to a POI far away from the POI where the user is current located, thereby reducing the performance of successive POI recommendation.

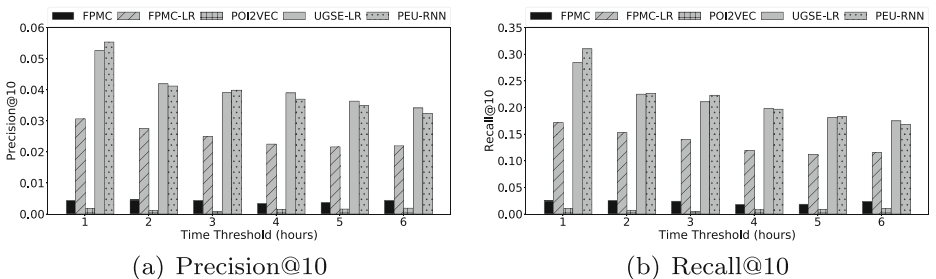


Figure 21 Time interval impact on accuracy in Gowalla

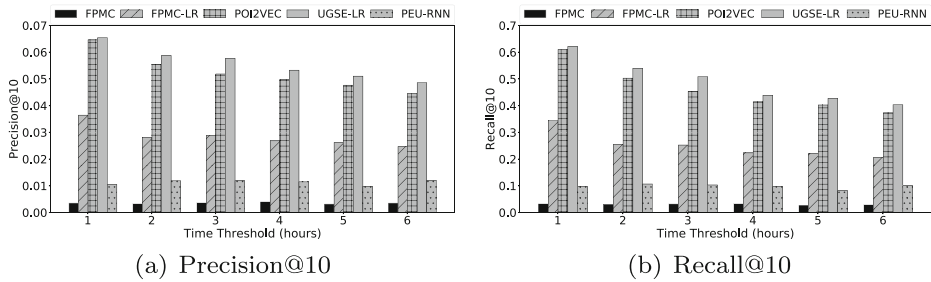


Figure 22 Time interval impact on accuracy in Brightkite

6 Conclusion

In this paper, we proposed a feature-based and a latent factor-based successive POI recommendation methods, named UGSE-LR and PEU-RNN, respectively. UGSE-LR not only considered user preference obtained by user-based collaborative filtering, but also considered the influence of the regions where POIs are located (i.e., regional influence) and successive transition influence obtained from the user's check-in records. On the other hand PEU-RNN utilized the LSTM model to find the POI transition from historical check-in records by using CBOW to encode users and POIs. Experimental results on two real datasets showed that the precision and recall of UGSE-LR and PEU-RNN are higher than the other existing successive POI recommendation methods. In addition, experiment results also showed that PEU-RNN is suitable for the dataset with many check-in sequences, while UGSE-LR is suitable for the dataset with moderate check-in sequences.

Acknowledgements This work was supported in part by Ministry of Science and Technology, Taiwan, under contracts 105-2221-E-009-160 and 105-2218-E-009-011.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- Allamanis, M., Peng, H., Sutton, C.A.: A convolutional attention network for extreme summarization of source code. CoRR, arXiv: [1602.03001](https://arxiv.org/abs/1602.03001) (2016)
- Bao, J., Zheng, Y., Mokbel, M.F.: Location-based and preference-aware recommendation using sparse geo-social networking data. In: Proceedings of the 20th ACM International Conference on Advances in Geographic Information Systems, pp. 199–208 (2012)
- Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. In: Aaai, vol. 12, pp. 17–23 (2012)
- Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: Successive point-of-interest recommendation. In: Proceedings of the 23rd International Conference on Artificial Intelligence, pp. 2605–2611 (2013)
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, arXiv: [1406.1078](https://arxiv.org/abs/1406.1078) (2014)
- Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, arXiv: [1412.3555](https://arxiv.org/abs/1412.3555) (2014)

7. Cireşan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: 2011 International Conference on Document Analysis and Recognition, pp. 1135–1139 (2011)
8. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 2069–2075 (2015)
9. Feng, S., Cong, G., An, B., Chee, Y.M.: Poi2vec: Geographical latent representation for predicting future visitors. In: AAAI Conference on Artificial Intelligence (2017)
10. Ferenc, G., Ye, M., Lee, W.-C.: Location recommendation for out-of-town users in location-based social networks. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 721–726 (2013)
11. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649 (2013)
12. He, J., Li, X., Liao, L., Song, D., Cheung, W.K.: Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (2016)
13. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval, pp. 230–237 (1999)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput* **9**(8), 1735–1780 (1997)
15. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Netw* **2**(5), 359–366 (1989)
16. Koren, Y., Bell, R., Volinsky, C., et al.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)
18. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw.* **8**(1), 98–113 (1997)
19. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection <http://snap.stanford.edu/data> (2014)
20. Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., Rui, Y.: Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining, pp. 831–840 (2014)
21. Liu, X., Liu, Y., Li, X.: Exploring the context of locations for personalized location recommendations. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI'16, pp. 1188–1194. AAAI Press (2016)
22. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech, pp. 1045–1048 (2010)
23. Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 3111–3119. Curran Associates Inc. (2013)
25. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
26. Wang, H., Terrovitis, M., Mamoulis, N.: Location recommendation in location-based social networks using user check-in data. In: Proceedings of the 21st ACM International Conference on Advances in Geographic Information Systems, pp.374–383 (2013)
27. Wang, W., Yin, H., Chen, L., Sun, Y., Sadiq, S., Zhou, X.: Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In: Proceedings of the 21st ACM International Conference on Knowledge Discovery and Data Mining, pp. 1255–1264 (2015)
28. Wang, B., Liu, K., Zhao, J.: Inner attention based recurrent neural networks for answer selection. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 Long Papers), pp. 1288–1297. Association for Computational Linguistics, Berlin (2016)
29. Waters, J.K.: *The Everything Guide to Social Media: All You Need to Know About Participating in Today's Most Popular Online Communities*. Adams Media Corporation (2010)

30. Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: Learning to rank with joint word-image embeddings. In: European Conference on Machine Learning (2010)
31. Xie, W., Bindel, D., Demers, A., Gehrke, J.: Edge-weighted personalized PageRank: Breaking a decade-old performance barrier. In: Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (2015)
32. Ye, M., Yin, P., Lee, W.-C., Lee, D.-L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th ACM International Conference on Research and Development in Information Retrieval, pp. 325–334 (2011)
33. Yin, H., Sun, Y., Cui, B., Hu, Z., Chen, L.: Lcars: A location-content-aware recommender system. In: Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining, pp. 221–229 (2013)
34. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware point-of-interest recommendation. In: Proceedings of the 36th ACM International Conference on Research and Development in Information Retrieval, pp. 363–372 (2013)
35. Yuan, Q., Cong, G., Sun, A.: Graph-based point-of-interest recommendation with geographical and temporal influences. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 659–668 (2014)
36. Zhang, J.-D., Chow, C.-Y., Li, Y.: Lore: Exploiting sequential influence for location recommendations. In: Proceedings of the 22nd ACM International Conference on Advances in Geographic Information Systems, pp. 103–112 (2014)
37. Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B., Liu, T.: Sequential click prediction for sponsored search with recurrent neural networks. CoRR, arXiv:1404.5772 (2014)
38. Zhang, W., Wang, J.: Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1221–1230 (2015)
39. Zhao, S., Zhao, T., Yang, H., Lyu, M.R., King, I.: Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (2016)