TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/11 pp208–218
DOI: 10.26599/TST.2019.9010059
Volume 26, Number 2, April 2021

POI Neural-Rec Model via Graph Embedding Representation

Kang Yang, Jinghua Zhu*, and Xu Guo

Abstract: With the booming of the Internet of Things (IoT) and the speedy advancement of Location-Based Social Networks (LBSNs), Point-Of-Interest (POI) recommendation has become a vital strategy for supporting people's ability to mine their POIs. However, classical recommendation models, such as collaborative filtering, are not effective for structuring POI recommendations due to the sparseness of user check-ins. Furthermore, LBSN recommendations are distinct from other recommendation scenarios. With respect to user data, a user's check-in record sequence requires rich social and geographic information. In this paper, we propose two different neural-network models, structural deep network Graph embedding Neural-network Recommendation system (SG-NeuRec) and Deepwalk on Graph Neural-network Recommendation system (DG-NeuRec) to improve POI recommendation. combined with embedding representation from social and geographical graph information (called SG-NeuRec and DG-NeuRec). Our model naturally combines the embedding representations of social and geographical graph information with user-POI interaction representation and captures the potential user-POI interactions under the framework of the neural network. Finally, we compare the performances of these two models and analyze the reasons for their differences. Results from comprehensive experiments on two real LBSNs datasets indicate the effective performance of our model.

Key words: Point-Of-Interest (POI) recommendation; graph embedding; neural networks; Deepwalk; deep learning; Location-Based Social Networks (LBSNs)

1 Introduction

With the booming of the Internet of Things (IoT), more and more embedded mobile devices are appearing in people's everyday lives[1]. As a result, a large volume of user check-in data are generated by social networks, such as Gowalla, Yelp, and Brightkite. Many researchers have discussed popular topics associated with Location-Based Social Networks (LBSNs) and big data applications, such as privacy protection[2–7], recommendation[8–10], and crowd-sourcing[2–4, 11, 12]. Point-Of-Interest (POI) recommendation systems encourage users to participate in more promotional merchant activities, thereby bringing substantial economic profits to third-party merchants and service providers. POI recommendations have received considerable attention in industrial and academic circles in recent years. However, in the field of LBSNs, classical recommendation methods, such as collaborative filtering, are not well suited to such complex scenarios[13, 14]. Traditional recommendation methods often face the following problems: First, the social relationships of users, the constraint of geographical location, and the preference cycles of users cannot be considered by traditional recommendation methods. Second, there is always some sparsity in the check-in data of users, and a lack of data can significantly affect the accuracy of traditional model predictions. Third, unlike the traditional commodity recommendation scenario, LBSN recommendations involve the interactive social relationships and geographical locations of users,

• Kang Yang, Jinghua Zhu, and Xue Guo are with the Department of Computer Science, Heilongjiang University, Harbin 150080. E-mail: 2181453@s.hlju.edu.cn; zhujinghua@hlju.edu.cn; 981256496@qq.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2019-10-04; revised: 2019-11-10; accepted: 2019-12-13

and the integration of these kinds of data is arduous for classical models. At present, one of the most fashionable tendencies in recommendation system exploration is the application of Deep Neural Networks (DNNs) in traditional models. Much research has shown that despite the sparsity of behavioral data, DNNs have an exceptional ability for representing implicit user-item interactions[15, 16].

Thus, we propose two neural-network POI recommendation models combined with the embedding of social and geographical graph information. Our model integrates user social relationship information, user-POI check-in interaction information, and POI geographical information to make a DNN recommendation. SG-NeuRec and DG-NeuRec differ from the methods that are initially used to obtain the correlation between user's social relations and geographical location. When capturing user's social relationship information and POI geographical information, the SG-NeuRec model represents these information types in the LBSNs as unweighted graphs, and uses an unsupervised feed-forward neural-network learning technology to obtain the user- and POI-graph embedding representations. Specifically, DG-NeuRec is based on a social-geographic two-layer heterogeneous graph. DG-NeuRec proposes the use of POI as a pseudo-sequence that integrates the user's social information and POI's geographic information, which are obtained by a random walk. Then the sequence is input into a Skip-gram model to learn the embedding representation of the POI. The embedding method for the user is similar to that of the POI in DG-NeuRec. The second parts of the SG-NeuRec and DG-NeuRec models are the same, in which potential low-dimensional user-POI representations are combined with graph embedding representations to learn the potential user-POI interaction representation under the DNN model of a multi-layer perceptron. In the meantime, considering that users' check-in behaviors have an apparent periodic inclination, we apply a shallow network to obtain the relevance representation between the time patterns of user check-ins and POIs under the same model. Finally, we regard the check-in sequence information of users as a positive sample of a binary classification and utilize a cross-entropy function as the model's loss computational formula. The main contributions of this paper are as follows.

(1) We propose a two-DNN recommendation model that fuses user's social relationship information and POI geographical relationship information. SG-NeuRec fuses user and POI graph embedding representations, seperately, and mines the potential interactions of users and POIs in a multi-layer perception model. Also, the DG-NeuRec combines the graph embedding representations of users and POIs in the same implicit space and models their potential interactions using Skip-gram.

(2) We consider the time seriality of the user check-in sequence in our model and integrate the time information of the user check-in sequence. Then, we develop a superficial network structure to extract the relevance between the time patterns of the users' check-ins and POIs.

(3) We evaluate the model's performance on two real datasets in subsequent experiments. Compared with other POI recommendation models, our proposed models achieve the state-of-the-art experimental results in terms of recommendation accuracy and ranking quality.

2 Related Work

With the ongoing and rapid development of the IoT and LBSNs, POI recommendation has become increasingly important[17]. Traditional recommendation models rely on user ratings or other explicit feedback to statically model user preferences. Many researchers, such as Zhang and Chow[18], have broadly utilized geographical, social, and temporal information to boost POI recommendation performance. Lian et al.[19] integrated the dimensional gathering of geographical information into a weighted matrix factorization model to deal with the matrix sparsity problem. Yuan et al.[20] proposed a method for fusing time-cycle information into a user-based Collaborative Filtering (CF) model for making time-aware POI recommendations.

Applying DNN models to extract features or mine interactions between users and items is becoming more and more popular in the field of recommendation system research. For instance, He et al.[21] regarded user-item implicit interaction as a binary classification task and proposed a DNN recommendation method. This method integrates the implicit feedback behavior of users and uses DNNs to model the potential interactions of users and items. Yang et al.[22] proposed a POI recommendation method based on DNNs, which fuses context information and CF methods. This method greatly exploits the implicit relationship between users and POIs.

Based on the geographic and time information fusion method, Yuan et al.[23] proposed a bounded-probabilistic-polynomial-based global site tag method that embeds the user's time information and POI's spatial information. This method also effectively alleviates the cold-start problem. In view of the different contributions of each kind of user's preference information, Baral and Li[24] proposed two fusion models based on ranking and matrix factorization to effectively combine various side information and proposed a new method for its utilization. The first work on graph embedding representation from the viewpoint of edges was proposed in 2018 by Wang et al.[25], who developed a novel network embedding approach, named DeepDirect, that preserves network topology. With the development of graph embedding methods, Christoforidis et al.[26] proposed the LBSN graph-embedding method RLINE based on a large-scale information-network-embedding method that effectively embeds side information, such as the user's social information, geographical distance information, and preference cycle information of an LBSN into the embedding space. In LBSN research, Cai et al.[27] used crowdsourced data to explore the influence maximization problem, and proposed a novel influence propagation model that effectively merges users' social and geographical relationships.

3 Our Model

3.1 Problem definition

A POI consists of two main parts: a geographical coordinate l and a description number p. Each POI is identified by its geographic coordinates of latitude and longitude. When a user accesses a POI, an interaction is formed, which consists of a triple (u, p, t), i.e., user u, accessed POI p, and a time stamp t. The user set is represented as U, the POI set is represented as P, the user's social graph is represented as G_U, and the POI geographical graph is represented as G_P. Our POI recommendation task involves a situation in which a user is on the time of stamp t, and we must determine which POI (p) user (u) may be interested in, according to the users' historical check-in behavior sequence.

Definition 1 (Pseudo sequence) This sequence is obtained by sampling the real world dataset. Although it does not exist in the real-world datasets, its distribution roughly conforms to the distribution of real-world datasets. Thus, this sequence can complement a sparse dataset.

Definition 2 (Block) For a user u_i, there is a historical POI check-in sequence S_i. Users who check into a location within a region have a high probability of attending locations that are proximate to it, so the sequence S_i has significant regional proximity. There is a check-in sequence SP_i, the distance between each POI in the sequence does not exceed the threshold D_δ and the check-in time for each POI in the sequence does not exceed the threshold D_τ. C_i is the total set of POI meeting the above criteria, $C_i = \mathrm{SP}_1 \cup \mathrm{SP}_2 \cup \cdots \cup \mathrm{SP}_n$, $\mathrm{SP}_i \cap \mathrm{SP}_j = \varnothing$, where SP_i is called a block.

Definition 3 (SG-NeuRec and DG-NeuRec) In this paper, we propose two models—Structural deep network Graph embedding Neural-network Recommendation system (SG-NeuRec) and Deepwalk on Graph Neural-network Recommendation system (DG-NeuRec). In SG-NeuRec, we use the Structural Deep Network Embedding (SDNE) model to obtain the embedding of a POI in an LBSN graph. A structural deep network is used to capture a highly nonlinear network structure and preserve the first-order and second-order proximities of the graph topology. DG-NeuRec represents an exploration and improvement of the SG-NeuRec model. The DG-NeuRec model considers the user's social relationship to be related to the geographic proximity of the POI, and puts both of them into the embedding representation learning process of the POI. In the SG-NeuRec and DG-NeuRec models, there are differences in the learning processes for POI embedding.

The main symbols in this paper are shown in Table 1.

3.2 SDNE Graph (SG) embedding representation

In this study, we used an unweighted graph to represent the social relation and geographic information in the dataset. This unweighted graph is described as an

Table 1 Symbol table.

Symbol	Description
D_δ	Distance limit threshold
N	Vertex set of adjacency matrix of unweighted graph
i	Vertex of adjacency matrix of unweighted graph
V_i	Vertex sequence made up of the joining edges' weight
E_{UU}	Set of all POIs
E_{PP}	Edge set of POIs
E_{PU}	Edge set of all users' POIs
S_i	Check-in sequence for user i
G	A heterogeneous map of LBSN that contains user social relationships and POI geographical relationships
α	Times of random walk
α_s	Hyperparameter of the regular term in the loss function
θ	Threshold of the distance of the POI point which selected by random walk
β	Maximum length of pseudo sequence generation

adjacency matrix. In the adjacency matrix of a user's social graph, when any two users have social relations, their corresponding edge weight is 1, otherwise, it is 0. In the adjacency matrix of the POI geographic graph, if the distance between any two points is no longer than 5, the corresponding edge weight between them is 1, otherwise, it is 0.

This task adopts an unsupervised automatic encoder to learn the embedding representation of the vertex in a graph network. This inspiration comes from the SDNE model[28], as shown in Fig. 1.

The adjacency matrix of an unweighted graph has N vertexes, and for each vertex i, $\boldsymbol{V}_i = (\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \ldots, \boldsymbol{v}_{iN})$ indicates a vertex sequence consisting of the weight of the adjoining edges. $\boldsymbol{V}_i$ is entered into the encoder as the input sequence. The formulation of the encoder's hidden layers is as follows:

$$\boldsymbol{h}_i^{(1)} = (\boldsymbol{W}_1 \boldsymbol{V}_i + \boldsymbol{b}_1) \tag{1}$$

$$\boldsymbol{h}_i = \sigma(\boldsymbol{W}_2 \boldsymbol{h}_i^{(1)} + \boldsymbol{b}_2) \tag{2}$$

and the formulation of decoder's hidden layers is as follows:

$$\hat{\boldsymbol{h}}_i^{(1)} = \sigma(\hat{\boldsymbol{W}}_1 \boldsymbol{h}_i + \hat{\boldsymbol{b}}_1) \tag{3}$$

$$\hat{\boldsymbol{V}}_i = \sigma(\hat{\boldsymbol{W}}_2 \hat{\boldsymbol{h}}_i^{(1)} + \hat{\boldsymbol{b}}_2) \tag{4}$$

where $\hat{\boldsymbol{W}}_*$, $\hat{\boldsymbol{b}}_*$, $\boldsymbol{W}_*$, and $\boldsymbol{b}_*$ are the parameters of the encoder-decoder model. σ is a nonlinear activation function. $\boldsymbol{h}_i$ is the output layer of the encoder, which is the embedding of vertex i. Through the decoding process, we can reconstruct the vertex sequence $\hat{\boldsymbol{V}}_i = (\hat{\boldsymbol{v}}_{i1}, \hat{\boldsymbol{v}}_{i2}, \ldots, \hat{\boldsymbol{v}}_{iN})$.

To minimize the difference between the reconstructed vertex sequence and the original input, which is a binary classification task, we use a logarithmic loss function. By minimizing the loss function, we can obtain the low-dimensional embedding of each vertex. So, our loss function is defined as follows:

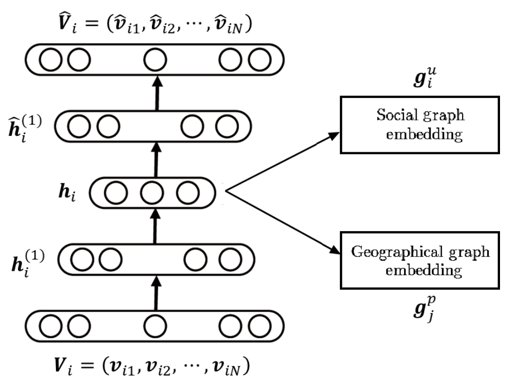

Fig. 1 SDNE graph embedding representation.

$$L_{\text{GE}} = -\sum_{i,j=1}^{N} [\boldsymbol{v}_{ij} \log \hat{\boldsymbol{v}}_{ij} + (1 - \boldsymbol{v}_{ij}) \log(1 - \hat{\boldsymbol{v}}_{ij})] + \alpha_s \sum_{i,j=1}^{N} \boldsymbol{v}_{ij} \left\| \boldsymbol{h}_i - \boldsymbol{h}_j \right\|^2 \tag{5}$$

where α_s is the hyperparameter. Then we obtain graph embeddings of POI $\boldsymbol{p}_j$ and user $\boldsymbol{u}_i$ in a low-dimensional implicit space via the graph embedding representation model. The graph embeddings of POI $\boldsymbol{p}_j$ and user $\boldsymbol{u}_i$ are denoted as $\boldsymbol{g}_i^u$ and $\boldsymbol{g}_j^p$, respectively.

3.3 Deepwalk on Graph (DG) neural network to improve POI recommendation

The DG is used to generate a user POI check-in pseudo-sequence on the heterogeneous graph $G = (U, P, E_{\text{UU}}, E_{\text{PP}}, E_{\text{PU}})$. To do so, first, DG inserts a POI p_i from the check-in set P into the pseudo sequence VS_i. Then, it extracts the user u_i in the check-in set $U_{u \in E_{\text{PU}(p_i)}}$ according to the normal distribution $N()$. It randomly walks α times on the graph $G_U = (U, E_{\text{UU}})$ and uses $N()$ to extract the POI p_j in the check-in set $P_{p \in E_{\text{PU}(u_{i+\alpha})}}$ of the user u_j obtained by a random walk. The geographic distance between p_i and p_j does not exceed the threshold θ, and then p_j is inserted into VS_i, DG repeats the above steps β times. Thus, we obtain a pseudo sequence VS_i of length β for the POI point p_i, the procedure for which is shown in Algorithm 1.

Then, because we generate a pseudo sequence VS_i for each POI p_i, we regard the point in VS_i as the high correlation probability point of the p_i point and the POI

Algorithm 1 Heterogeneous graph random walk

Input: Heterogeneous graph G, geographic distance threshold θ, times of random walks α, pseudo sequence length β
Output: Pseudo-sequence set for all users VS

```
Initialization set VS
for all p_i ∈ P do
    Initialization set VS_i
    for j = 0; j < β; j++ do
        Insert p_i into the pseudo sequence VS_i
        Extract user u_i in the set U_{u∈E_PU(p_i)} according to the
        normal distribution N()
        Randomly walk α times on the G_U = (U, E_UU) to get
        the user u_j
        Use the N() to extract the POI point p_j in the check-in
        set P_{p∈E_PU(u_{i+α})} of the user u_j, the geographic distance
        between p_i and p_j does not exceed the threshold θ
        Insert p_j into VS_i
    end for
    Insert VS_i into VS
end for
return VS
```

in the sequence can be repeated. The relevance of a POI is proportional to the number of its occurrences in the VS_i. By entering the pseudo-sequence VS_i into the Skip-gram model[29], we can obtain the maximization objective function as follows:

$$\operatorname{argmax} \text{ opt} = \frac{1}{\beta}\sum_{i=1}^{\beta}\sum_{j\neq i}^{\beta}\log\ p(\boldsymbol{x}_j|\boldsymbol{x}_i) \tag{6}$$

where β is the maximum length of the pseudo sequence VS_i, $p(\boldsymbol{x}_j|\boldsymbol{x}_i)$ is defined in the following Softmax function:

$$p(\boldsymbol{x}_j|\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_j)}{\sum\limits_{\beta}\exp(\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_\beta)} \tag{7}$$

where $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ are the embedding vectors of the POI, which represent the target and previous POIs, respectively. However, when β is too large, it makes the gradient calculation $\nabla(\boldsymbol{x}_j|\boldsymbol{x}_i)$ is hard to calculate. According to the Skip-gram negative sampling method proposed by Mikolov[30], Eq. (7) can be modified as follows:

$$p(\boldsymbol{x}_j|\boldsymbol{x}_i) = \sigma(\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_j)\prod_{k=1}^{E}\sigma(-\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_k) \tag{8}$$

where $\sigma(\,)$ is the sigmoid function, i.e., $\sigma(\,)=1/(1+\mathrm{e}^{-x})$, E is the negative sample for each positive sample, and C is a check-in set. The more times x_i appears in the check-in total C, the lower the probability is sampled, so we obtain the following final objective function:

$$\operatorname{argmax} \text{ opt} = \frac{1}{\beta}\sum_{i=1}^{\beta}\sum_{j\neq 1}^{\beta}\log\ p(\boldsymbol{x}_j|\boldsymbol{x}_i) =$$

$$\frac{1}{\beta}\Big(\sum_{i=1}^{\beta}\sum_{j\neq i}^{\beta}\log\sigma(\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_j) + \sum_{k=1}^{E}\log\sigma(-\boldsymbol{w}_i^{\mathrm{T}}\cdot\boldsymbol{v}_k)\Big) \tag{9}$$

In this way, we can obtain high-quality POI embedding representations that integrate user's social information and POI geographic location information via the DG. The significance of this model is that the social and geographic location information we obtained in the embedding representation is in a unified embedding space, so that the subsequent neural network can effectively measure and adjust the embeddings. Finally, for each user u_i, we can use the embedding matrix combined with the check-in record to generate a POI historical check-in embedding sequence $\boldsymbol{V}_u = \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$, where $\boldsymbol{v}_i$ represents the d-dimension embedding vector of $\boldsymbol{x}_i$. Figure 2 shows the Deepwalk on graph neural network.

3.4 User-POI interaction representation

We apply a stacking multi-layer structure in the model to capture the interaction representation between the user and the POI. Figure 3 shows the model used in this study. Next, we provide details of the structure of the proposed model.

Input layer: The input layer receives one-hot

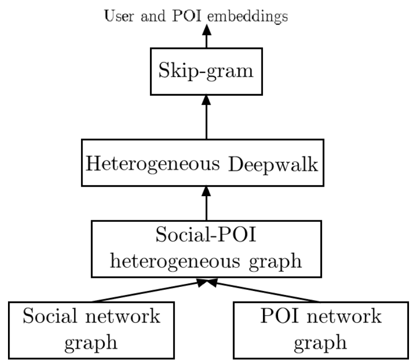

Fig. 2 Deepwalk on graph neural network.

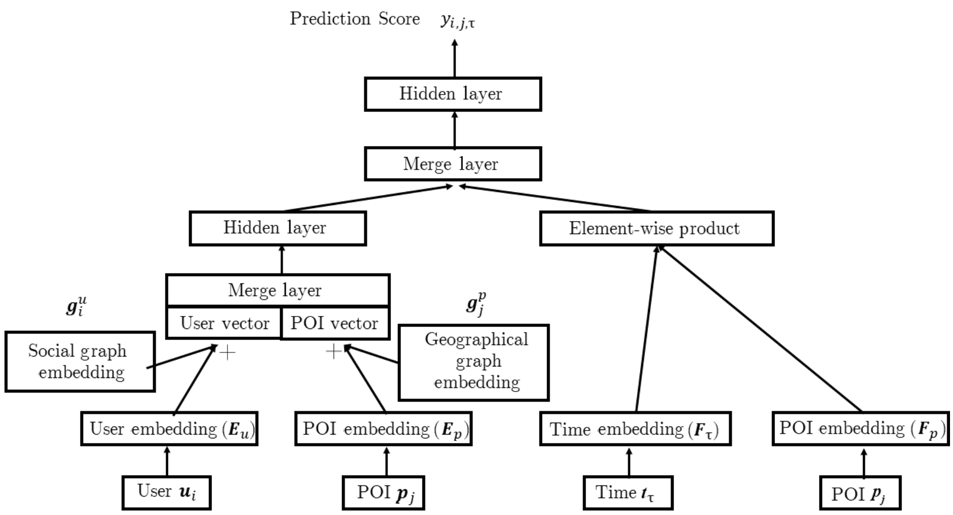

Fig. 3 SG-NeuRec model.

encoded user-POI item pairs $(\boldsymbol{u}_i, \boldsymbol{p}_j)$. We use the one-hot vector multiplied by the weight matrix to quickly find the embedding of the user or POI.

Embedding layer: The embedding layer can map the one-hot encoded user and POI vectors to the low-dimensional embedding representation vector. Through iterative adjustment of the model, we obtain two implicit vector embedding matrices $\boldsymbol{E}_u \in \mathbb{R}^{N\times D}$ and $\boldsymbol{E}_p \in \mathbb{R}^{M\times D}$ seperately, where N denotes the number of users, M denotes the number of POIs and D denotes dimension of the users' and POIs' embeddings. We use $\boldsymbol{u}_i$ and $\boldsymbol{p}_j$ to represent the one-hot encoding vectors of the users and POIs, respectively, and $\boldsymbol{E}_u^{\mathrm{T}}\boldsymbol{u}_i$ and $\boldsymbol{E}_p^{\mathrm{T}}\boldsymbol{p}_j$ represent the user's $\boldsymbol{u}_i$ and POI's $\boldsymbol{p}_j$ embedding vectors, respectively.

Merge layers: In the merge layers, we add $\boldsymbol{E}_u^{\mathrm{T}}\boldsymbol{u}_i$ to the social embedding $\boldsymbol{g}_i^u$, and add $\boldsymbol{E}_p^{\mathrm{T}}\boldsymbol{p}_j$ to the geographical embedding $\boldsymbol{g}_j^p$. The purpose of doing this is to generate a user vector and a POI vector seperately. Then we concatenate these two vectors, which serve as the input vector z of the subsequent layer. The formulation of merge layer is as follows:

$$z = [\boldsymbol{E}_u^{\mathrm{T}}\boldsymbol{u}_i + \boldsymbol{g}_i^u; \boldsymbol{E}_p^{\mathrm{T}}\boldsymbol{p}_j + \boldsymbol{g}_j^p] \tag{10}$$

Hidden layer: The hidden layer uses a stacking multi-layer model to learn the implicit interaction embeddings of the users and POIs. The output of the stacking multi-layer model of the K-th layer is expressed as follows:

$$\boldsymbol{h}^K(z) = a(\boldsymbol{W}^K\boldsymbol{h}^{K-1}(z) + \boldsymbol{b}^K) \tag{11}$$

where $\boldsymbol{W}^K$ and $\boldsymbol{b}^K$ are the weight matrix and bias vector of the stacking multi-layer model's K-th layer, respectively, and a is a nonlinear activation function. Here we apply ReLU, $\mathrm{ReLU}(x) = \max(0, x)$, as the activation function.

We take the last output of the stacking multi-layer model as the interaction embedding vector $\boldsymbol{h}_{u,p}$ between the user and the POI. The formulation of $\boldsymbol{h}_{u,p}$ is as follows:

$$\begin{aligned}\boldsymbol{h}_{u,p} =& \boldsymbol{h}^K(\boldsymbol{h}^{K-1}(\cdots \boldsymbol{h}^1(z))) = \\ & \boldsymbol{h}^K(\boldsymbol{h}^{K-1}(\cdots \boldsymbol{h}^1([\boldsymbol{E}_u^{\mathrm{T}}\boldsymbol{u}_i + \boldsymbol{g}_i^u; \boldsymbol{E}_p^{\mathrm{T}}\boldsymbol{p}_j + \boldsymbol{g}_j^p])))\end{aligned} \tag{12}$$

3.5 Time pattern-POI relevance representation

POI check-in records often have significant time interval features. Therefore, the use of a POI recommendation model based on time modeling is often conducive to improving the prediction accuracy of the model. In this study, we took 1 h as the time-sampling interval, namely, seven days a week, 24 h a day, for a total of 168 sample points. In this way, we divided a week into 168 periods as the time pattern of users' check-ins. Similar to the user-POI model, we used the same method to obtain an embedding representation of time. The time pattern of a user's check-in $\boldsymbol{t}_\tau$ and POI $\boldsymbol{p}_j$ are embedded, respectively, and we obtain the embedding vector of time pattern $\boldsymbol{F}_t^{\mathrm{T}}\boldsymbol{t}_\tau$ and the embedding vector of POI $\boldsymbol{F}_p^{\mathrm{T}}\boldsymbol{p}_j$, where $\boldsymbol{F}_t$ and $\boldsymbol{F}_p$ are the implicit vector matrices of the POIs and time patterns, respectively. We applied a DNN model of the relevance between the time pattern and POI, and took the Hadamard product of the two embedding matrices to obtain the relationship vector $\boldsymbol{h}_{t,p}$ between the time POI and pattern. The Hadamard product was chosen to obtain an equally long relationship vector $\boldsymbol{h}_{t,p}$ with embeddings ($\boldsymbol{t}_\tau$ and $\boldsymbol{p}_j$), the formulation for which is as follows:

$$\boldsymbol{h}_{t,p} = \boldsymbol{F}_t^{\mathrm{T}}\boldsymbol{t}_\tau \odot \boldsymbol{F}_p^{\mathrm{T}}\boldsymbol{p}_j \tag{13}$$

where $\odot$ denotes the Hadamard product of the vectors, i.e., the element-wise product.

3.6 Model training

Now, we have the interaction vector $\boldsymbol{h}_{u,p}$ between the POIs and users, and the relationship vector $\boldsymbol{h}_{t,p}$ between the POIs and time patterns. We then concatenate these two embeddings in the merge layer and put it into the hidden layer to obtain the implicit vector $\boldsymbol{h}_{u,p,t}$ among the users, POIs, and time patterns via a hidden layer, the formulation for which is as follows:

$$\boldsymbol{h}_{u,p,t} = \mathrm{ReLU}(\boldsymbol{W}\left[\boldsymbol{h}_{u,p}; \boldsymbol{h}_{t,p}\right] + \boldsymbol{b}) \tag{14}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and bias vector of the hidden layer, respectively. The final output layer generates the probability of the interaction prediction value of user $\boldsymbol{u}_i$ and POI $\boldsymbol{p}_j$ on time stamp $\boldsymbol{t}_\tau$, the formulation for which is as follows:

$$\hat{y}_{i,j,\tau} = \sigma(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}_{u,p,t}) \tag{15}$$

where $\boldsymbol{H}$ is the correlation weight of the prediction layer and σ is the sigmoid function, which is defined as $\sigma(x) = 1/(1 + \mathrm{e}^{-x})$.

The model in this paper is a binary classification model, and the activation function of the output layer is a sigmoid function. Therefore, negative sampling is used to prevent the model from fitting. In this paper's sample set, Y is the set of POIs that is accessed by users in the dataset, and Y^- is the set of POIs not accessed by users in the dataset. This model's target likelihood function is as follows:

$$P(Y, Y^-|\Theta) = \prod_{(i,j,\tau)\in Y} \hat{y}_{i,j,\tau} \prod_{(i,j,\tau)\in Y^-} (1 - \hat{y}_{i,j,\tau}) \tag{16}$$

where Θ indicates the model parameters. Then we take the negative logarithm of the likelihood function. The final objective function is defined as follows:

$$L_{\text{predict}} = -\sum_{(i,j,\tau)\in Y} \log \hat{y}_{i,j,\tau} - \sum_{(i,j,\tau)\in Y^-} \log(1-\hat{y}_{i,j,\tau}) \tag{17}$$

We add the graph embedding representations which learning by our model to the loss function as a regular term. Users' embedding matrix and POIs' embedding matrix are denoted as $\boldsymbol{L_U}$ and $\boldsymbol{L_P}$, respectively. In this way, we uniformly optimize the model, and any gradient adjustment will affect all parts of the embedding layer. Finally, we can obtain the objective function as follows:

$$L = L_{\text{predict}} + \boldsymbol{\lambda}_U \cdot \boldsymbol{L}_U + \boldsymbol{\lambda}_P \cdot \boldsymbol{L}_P \tag{18}$$

where $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_P$ are hyperparameters, which are used to manually adjust the weight of the regular term.

4 Experiment and Analysis

4.1 Experimental settings

We evaluated the proposed model on Gowalla (http://snap.stanford.edu/data/loc-Gowalla.html) and Yelp (https://www.kaggle.com/yelp-dataset/yelp-dataset), two real-world datasets. Each of the check-in data in these datasets contains a user ID, a POI ID, a timestamp, and the latitude and longitude of the POI. Table 2 lists the statistics for these datasets, density is used to measure the sparseness of a dataset.

We used the Hit Ratio (HR) and Mean Reciprocal Rank (MRR) as measurement criteria for the model, and choose top-K recommended results as a comparison. HR embodies the most basic performance of the model, and POI recommendations often occur in the top one or top five scenarios, so the MRR can effectively measure whether the model could be the first to recommend suitable POIs for users. As the test set, we used the last check-in record in the user check-in sequence, and the rest of the sequence served as the training set. For each user, we generated 100 negative samples to make ensure that the model was not skewed. We obtained the HR and MRR values for each test user and calculated the average scores. We also tested the baseline model on the same dataset and analyzed the performances of both to identify the advantages of our model.

Table 2 Datasets statistics.

Dataset	Number of users	Number of POIs	Number of check-ins	Density (%)
Gowalla	43 071	46 234	1 720 082	0.0500
Yelp	30 887	18 995	860 888	0.1399

For the embedding of the geographic location of the POI, if the geographical distance of two POI points did not exceed 1 km, we set the value in the adjacent weight matrix to 1, otherwise we set it to 0. For the hyperparameter α_s of the regular term in the loss function, we set the default value to 0.1. We used the Adam optimization method as the gradient optimizer for the model and set the learning rate to 0.0001 and batch size to 256. We used a normal distribution to randomly initialize all the learn-able parameters of the model. α_s value was 4 and β value was 64, and we set the dropout ratio to 0.2. In the embedding of the timing feature, we set the embedding dimension to 16. For the embedding of the user and POI interactive feature, we set the embedding dimension to 32 and the depth of the hidden layer to 3, using a tower structure. Regarding the size of the hidden layer in the stacking multi-layer model, it is reduced by half with each layer, that is, its size was 64, 32, 16, respectively. In Section 4.3, we analyze the effect of the size and number of hidden layers on the performance of the proposed model through comparative experiments. For the objective function of the model, we also compared and analyzed the settings of the hyperparameters $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_P$ in the experimental analysis to select the optimal parameters of the model.

The baseline algorithms used in our comparative experiments include traditional recommendation methods based on collaborative filtering, neural network recommendation methods, and recommendation methods based on combined geographic and social factors.

- **BPR-MF**[31]: It is a recommendation model based on Bayesian personalized ranking, which optimizes the matrix factorization model by using the pair-wise loss.
- **NeuMF**[21]: It is a recommendation model based on neural networks, which utilizes a multi-layer perceptron to model the latent interaction between a user and an item.
- **GE**[32]: It is a POI recommendation model based on graph embedding, which jointly embeds bipartite graphs of POIs, time, regions, and behaviors in the latent space.
- **SoRec**[33]: It is a social recommendation algorithm that models users and items by decomposing a social relationship matrix and a rating matrix, and sharing a users' latent feature matrix.
- **PACE**[22]: It is a POI recommendation model based on neural networks that integrates context factors and

a collaborative filtering method into a semi-supervised learning framework.

4.2 Performance comparison

Experiment results on the Gowalla and Yelp datasets are shown in Tables 3 and 4, respectively, where HR@K and MRR@K mean that we extract the first K samples of the model prediction as the basis for the calculation of HR and MRR. We can see that our proposed model achieved the best experimental results. Our model also showed sufficient robustness while improving the prediction accuracy. Otherwise, the GE model performed better than SoRec and BPR-MF due to geographical considerations. The performance of NeuMF was superior to those of BPR-MF, GE, and SoRec, which indicates that the DNN approach has significant advantages in modeling the potential interaction between POI and users. Lastly, compared with SG-NeuRec, the recommendation accuracy and ranking performance of our improved model DG-NeuRec were more effective to a certain extent.

4.3 Model analysis

4.3.1 Hide layer settings

By changing the number of hidden layers and the dimensional capacity of each hidden layer, we explored the impact of different hidden-layer designs on the recommendation performance. In the experiment, we changed the size of the embedding dimensions of users and POIs, and then changed the dimensional capacity of the hidden layers accordingly. For example, if the embedding dimensions of the users and POIs were 32, and the number of hidden layers was 3, the dimensions of the each hidden layer were 64, 32, and 16. As shown in Tables 5 and 6, when the embedding dimensions of the users and POIs were 32 and the number of hidden layers was 3, the MRR@20 of the recommendation model achieved maximum value. In Tables 5 and 6, D is the dimension of the embeddings and L is the number of hidden layers.

Table 3 Experiment results on the Gowalla dataset.

Method	HR@5	HR@10	HR@20	MRR@20
BPR-MF	0.3018	0.3364	0.4039	0.0986
SoRec	0.3760	0.4217	0.4836	0.1103
GE	0.4221	0.4782	0.5103	0.1409
NeuMF	0.5139	0.5787	0.6394	0.1749
PACE	0.5208	0.5914	0.6430	0.2030
SG-NeuRec	**0.5521**	**0.6203**	**0.7630**	**0.2541**
DG-NeuRec	**0.5583**	**0.6271**	**0.7636**	**0.2533**

Table 4 Experiment results on the Yelp dataset.

Method	HR@5	HR@10	HR@20	MRR@20
BPR-MF	0.2580	0.3116	0.3904	0.1030
SoRec	0.3350	0.4382	0.4701	0.1309
GE	0.3696	0.4501	0.5113	0.1607
NeuMF	0.4119	0.5118	0.6128	0.1617
PACE	0.4299	0.5210	0.6209	0.1810
SG-NeuRec	**0.5023**	**0.5890**	**0.7156**	**0.2159**
DG-NeuRec	**0.5047**	**0.5651**	**0.7203**	**0.2009**

4.3.2 SG-NeuRec versus DG-NeuRec

The main difference between the SG-NeuRec and DG-NeuRec methods is the learning methods used for the users and POI embeddings. SG-NeuRec utilizes an encoder-decoder framework to extract the relationships of the sequence elements. This method takes into account the user's friend relationships and the geographical proximity of the POI points. However, the SG-NeuRec method does not actively expand sparse datasets, so the robustness of the model is negatively affected when the sparsity of the data is particularly severe. For this reason, we proposed the DG-NeuRec method to improve our SG-NeuRec POI recommendation model. DG-NeuRec uses Deepwalk to obtain the pseudo-check-in sequence of the user. For POI embeddings, DG-NeuRec uses Deepwalk to obtain the pseudo-check-in sequence of the user, and then uses Skip-gram to obtain the POI embeddings. The methods used to obtain the user embeddings are similar.

From Figs. 4 and 5, we can see that when using SG-NeuRec, the POI embeddings in the T-distributed Stochastic Neighbor Embedding (T-SNE) view have a zonal distribution. Figure 4 suggests that the POI

Table 5 SG-NeuRec: MRR@20 results of different hidden layers on Gowalla dataset.

D	$L=0$	$L=1$	$L=2$	$L=3$	$L=4$
8	0.211	0.2219	–	–	–
16	0.2127	0.2368	0.2513	–	–
32	0.2183	0.2424	0.2522	**0.2541**	–
64	0.229	0.2437	0.2528	0.2532	0.2528

Table 6 SG-NeuRec: MRR@20 results of different hidden layers on Yelp dataset.

D	$L=0$	$L=1$	$L=2$	$L=3$	$L=4$
8	0.1823	0.2076	–	–	–
16	0.1964	0.2083	0.212	–	–
32	0.1973	0.2089	0.2136	**0.2159**	–
64	0.1987	0.2102	0.2117	0.2142	0.2137

Fig. 4 T-SNE view of the POI embeddings of the Gowalla dataset obtained by SG-NeuRec.

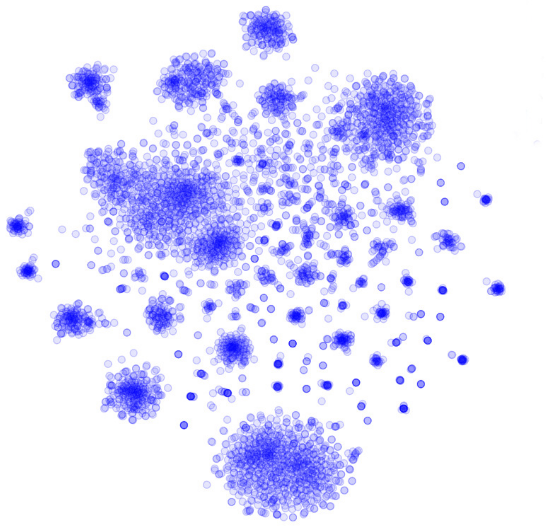

Fig. 5 T-SNE view of the POI embeddings of the Gowalla dataset obtained by DG-NeuRec.

embeddings learned through SG-NeuRec have weak relevance. Specifically, the POI points are only adjacent to a few POI points in the embedding space, and the distribution of different POI points is very discrete. In the contrast, POI embeddings obtained by DG-NeuRec exhibit a cluster distribution in the T-SNE view. This distribution is more conducive for the inference of subsequent neural networks regarding the relevance of different POIs, so these the neural networks can make predictions by referring to users' POI check-in records.

5 Conclusion and Future Work

In this paper, we proposed a neural-network POI recommendation model called SG-NeuRec and then improved the SG-NeuRec method, which we called DG-NeuRec. Our model integrates user's social relationship information with the geographic information of POIs. Using the framework of a DNN model, a multi-layer perceptron is used to learn the potential representation of the interaction between a user and a POI. In subsequent improvements, we enabled our model to learn embedding representations directly on the LBSN graph. We also designed a shallow network to explore the relationship between users and POIs. We combined the embeddings of user-POI interactions with time pattern embeddings under the DNN framework to improve the model's performance with respect to POI recommendations. Finally, we conducted experiments on two real-world datasets and demonstrated the validity of the proposed model.

In future work, we will use the knowledge graph to further explore the potential relationship between users and POIs.

References

[1] X. Zheng, Z. Cai, and Y. Li, Data linkage in smart internet of things systems: A consideration from a privacy perspective, *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[2] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems, *Computer Networks*, vol. 135, pp. 32–43, 2018.

[3] Y. Xu, L. Qi, W. Dou, and J. Yu, Privacy-preserving and scalable service recommendation based on SimHash in a distributed cloud environment, *Complexity*, vol. 2017, p. 3437854, 2017.

[4] Z. Cai and Z. He, Trading private range counting over big IoT data, in *Proc. of the 39th IEEE International Conference on Distributed Computing Systems*, Dalas, TX, USA, 2019, pp. 144–153.

[5] Y. Liang, Z. Cai, Q. Han, and Y. Li, Location privacy leakage through sensory data, *Security and Communication Networks*, vol. 2017, p. 7576307, 2017.

[6] Y. Huang, Z. Cai, and A. G. Bourgeois, Search locations safely and accurately: A location privacy protection algorithm with accurate service, *Journal of Network and Computer Applications*, vol. 103, pp. 146–156, 2018.

[7] Z. Cai, X. Zheng, and J. Yu, A differential-private framework for urban traffic flows estimation via taxi companies, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[8] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, A trust-based probabilistic recommendation model for social networks, *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.

[9] L. Qi, X. Xu, X. Zhang, W. Dou, C. Hu, Y. Zhou, and J. Yu, Structural balance theory-based e-commerce recommendation over big rating data, *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 301–312, 2018.

[10] X. Xia, J. Yu, S. Zhang, and S. Wu, Trusted service scheduling and optimization strategy design of service recommendation, *Security and Communication Networks*, vol. 2017, p. 9192084, 2017.

[11] Y. Wang, Z. Cai, Z. Zhan, Y. Gong, and X. Tong, An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing, *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.

[12] Z. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, doi: 10.1109/TNSE.2018.2830307.

[13] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, A survey of matrix completion methods for recommendation systems, *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 308–323, 2018.

[14] Y. Liu, S. Wang, M. S. Khan, and J. He, A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering, *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 211–221, 2018.

[15] L. Ma, L. Liao, D. Song, and J. Wang, A latent entity-document class mixture of experts model for cumulative citation recommendation, *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 660–670, 2018.

[16] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, Personalized real-time movie recommendation system: Practical prototype and evaluation, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 180–191, 2019.

[17] X. Zheng, Z. Cai, J. Li, and H. Gao, Location-privacy-aware review publication mechanism for local business service systems, in *Proc. of IEEE Conference on Computer Communications*, Atlanta, GA, USA, 2017, pp. 1–9.

[18] J.-D. Zhang and C.-Y. Chow, Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, 2015, pp. 443–452.

[19] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2014, pp. 831–840.

[20] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, Time-aware point-of-interest recommendation, in *Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 2013, pp. 363–372.

[21] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, Neural collaborative filtering, in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 173–182.

[22] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, Canada, 2017, pp. 1245–1254.

[23] Q. Yuan, G. Cong, and A. Sun, Graph-based point-of-interest recommendation with geographical and temporal influences, in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, Shanghai, China, 2014, pp. 659–668.

[24] R. Baral and T. Li, Exploiting the roles of aspects in personalized POI recommender systems, *Data Mining and Knowledge Discovery*, vol. 32, no. 2, pp. 320–343, 2018.

[25] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu, and B. Wang, Deepdirect: Learning directions of social ties with edge-based network embedding, *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2277–2291, 2018.

[26] G. Christoforidis, P. Kefalas, A. N. Papadopoulos, and Y. Manolopoulos, Reline: Point-of-interest recommendations using multiple network embeddings, arXiv preprint arXiv:1902.00773, 2019.

[27] J. L. Z. Cai, M. Yan, and Y. Li, Using crowdsourced data in location-based social networks to explore influence maximization, in *Proc. of the 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, 2016, pp. 1–9.

[28] D. Wang, P. Cui, and W. Zhu, Structural deep network embedding, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 1225–1234.

[29] X. Rong, Word2vec parameter learning explained, arXiv preprint arXiv:1411.2738, 2014.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, in *Proc. of Advances in Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, 2009, pp. 452–461.

[32] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, Learning graph-based POI embedding for location-based recommendation, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 15–24.

[33] H. Ma, H. Yang, M. R. Lyu, and I. King, Sorec: Social recommendation using probabilistic matrix factorization, in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, Napa Valley, CA, USA, 2008, pp. 931–940.

Kang Yang received the BS degree from Shandong University of Finance and Economics, Jinan, China in 2018. He is currently pursuing the MS degree in Heilongjiang University. His research interests include AI, recommender system, and social networks analysis. He is a recipient of many scholarships.

Jinghua Zhu received the BS degree from Heilongjiang University, Harbin, China in 1999, and the MS and PhD degrees from Harbin Institue of Technology in 2002 and 2009, respectively. She has been a professor at the School of Computer Science and Technology, Heilongjiang University, China since 2016. She has published many high quality conference and journal research papers. Her research interests include social networks, data mining, uncertain databases, and wireless sensor networks.

Xu Guo received the BS degree from Anhui University of Science and Technology in 2017. He is now a master student in Heilongjiang University. His research interests include AI, recommender system, and social networks analysis.