



Neu-PCM: Neural-based potential correlation mining for POI recommendation

Jun Zeng¹ · Haoran Tang¹ · Yizhu Zhao¹ · Junhao Wen¹

Accepted: 30 July 2022 / Published online: 23 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In the past few years, with the advent and developments of artificial intelligence, the location-based mobile services became prevalent, which produce large-scale location-based data that embeds abundant hints of user preferences on locations. Point of interest (POI) recommendation, as one of the significant mobile services, aims to recommend new satisfactory location to user according to their historical records. Existing classical POI recommendation models based on matrix factorization and collaborative filtering both face a significant challenge that they can not capture the users' preferences deeply and effectively. Hence, we propose a deep POI recommendation model (Neu-PCM), which is based on neural networks, to extract the potential correlation between user and location. First, we present a local learning and dimension-reduction network to obtain the key information. Second, we put forward a union network to mine the potential correlation effectively. Third, we build a deep matrix factorization to enhance the final prediction by correlation combination. The results of experiments on real-world datasets demonstrate our model outperforms the current popular recommendation algorithms.

Keywords Recommendation system · Point of interest · Neural network · Correlation mining · Deep learning

1 Introduction

In recent years, with the continuous innovation and technical improvement of artificial intelligence, location-based mobile applications, leading by Facebook, Instagram and Foursquare which link the virtual network and the real world are becoming pervasive in our daily lives [35]. They provide users with many mobile services by encouraging users to share locations and experiments [20]. Point of interest (POI) recommendation, as one of the most popular location-based mobile services, recommends users'

unexplored but potentially interested locations [22]. In particular, when users face massive amounts of information in an unfamiliar area, recommendation system attempt to recommend the most suitable locations [21]. It will facilitate the outdoor activities of users by producing a customized list of POIs [16]. Meanwhile, it could also bring huge commercial benefits to third-party businesses or advertisers [25], and may promote the development of city. POI recommendation uses the check-in data generated by users to understand user behavior patterns and preferences of locations [1]. Historical behavior has significant effect on future predicting for recommendation system [11]. Since POI recommendation is high value to both users and service providers, it has attracted much attention from academia and industry in recent years [33].

Current POI recommendation helps users explore locations which they have not visited before. As the life quality of modern individuals rises, the outdoor entertainment demands increase significantly as follow. Therefore, it is a necessity to technically improve the POI recommendation accuracy. Original technologies of machine learning are widely used in recommendation system, such as collaborative filtering and matrix factorization [2]. Subsequently, a good local minimum depends largely on the initialization of the latent feature vectors of users and items [18]. The first

✉ Jun Zeng
zengjun@cqu.edu.cn

Haoran Tang
tangaoran@cqu.edu.cn

Yizhu Zhao
zhaoyizhu@cqu.edu.cn

Junhao Wen
jhwen@cqu.edu.cn

¹ School of Big Data, Software Engineering, Chongqing University, Chongqing, China

challenge for POI recommendation is data sparseness which caused the failure of original matrix factorization. Practically, users always visit only a few locations, which makes it more difficult to capture personal preferences effectively. The second challenge is the problem of implicit preference which could not be mined in traditional way. The implicit correlation between users and POIs indicates complex pattern that hides under the original user-POI data. It is hard to characterize heterogeneous and complex data in the recommendation system [5]. Neural network in deep learning, as a powerful technology for mining the potential correlation, is able to solve those problems exactly [28]. The correlation represents the semantic relevance of the user vector and location vector, which shows the implicit preference deeply. By applying neural network in POI recommendation, it is possible to relieve the negative impact due to data sparseness and obtain the potential preferences of user on locations and then we could design a personal recommendation list to finish mobile service. Hence, in this paper, we propose a deep POI recommendation model that focus on mining the potential correlations between users and locations. Our model is called Neu-PCM that is neural-based potential correlation mining for POI recommendation, and its purpose is the predicted probability of user visiting a candidate location. For simplicity, POI is also called location.

The main contributions of this paper can be summarized as follows.

- First, in order to avoid the uncertainty caused by data sparsity and make neural network more concentrate on local features, we present local embedding by dividing the first feature extraction layer into multiple parts instead of fully connected layer. Local embedding could strengthen the ability of learning the original vector of users or locations.
- Second, considering the dimension of vector will affect the subsequent process of recommendation task, we build two different dimension-reduction neural networks for users and locations respectively. This will further capture the deep information of features and get a lower dimension which is suitable for recommendation calculation.
- Third, our core goal in this paper is the potential correlation mining between users and locations. Hence, after obtaining the new low-dimension vectors, we construct a union neural network to compute the potential correlation by concatenating vectors of users and locations. This helps us understand the implicit preferences of users and make full advantage of them.
- Finally, to improve the performance of our model, we define a deep matrix factorization and then combine the potential correlation with it. Neu-PCM can output the probability on visiting location. We conduct abundant

experiments and the results demonstrate our Neu-PCM outperforms some popular recommendation algorithms.

The rest of our paper is organized as follows. Section 2 introduces the related work of POI recommendation based on deep learning. Section 3 presents our deep model Neu-PCM. Section 4 discusses and analyzes the experimental results. Finally, Section 5 makes the conclusion of the paper.

2 Related work

POI recommendation aims to find new locations where users will be interested in according to their historical records. The improvements of POI recommendation contribute to the location exploration. As an important component of recommendation system, POI recommendation is widely studied by researchers.

Since Collaborative Filtering (CF) has been proposed for a long time, its idea still provides many researchers inspiration [13]. Matrix Factorization (MF) which fills unknown items in matrix, has the same contribution [23]. There are many models improving the performance of original CF or MF by some novel technologies. Liu et al. define a heterogeneous information network based POI recommendation model to model various heterogeneous context features [19]. Yang et al. present a general and principled POI recommendation framework to solve the sparsity problem by smoothing among users and locations [29]. In order to alleviate the error propagation produced by intermediate outputs, Wang proposes a low-rank and sparse matrix factorization with prior relations which predicts items through a sum of the learned low-rank matrix and sparse matrix [24]. Neural network in deep learning has been used in various research fields including recommendation system since its powerful non-linear ability [8]. Ebesu et al. design a collaborative memory networks based on deep architecture to unify the two classes of CF models capitalizing on the strengths of the global structure of latent factor model and local neighborhood-based structure in a non-linear fashion [6]. Xue et al. present a novel deep matrix factorization model based on neural network to learn common low-dimension feature space for both users and locations [27].

By replacing the inner product with a neural architecture, he proposes a general neural collaborative filtering which could express and generalize MF under its deep framework [9]. Zeng et al. mine the relationship between user movement and context and proposes a method based on a recurrent neural network and self-attention mechanism [31]. Deng et al. present a deep collaborative filtering to combine the strengths of representation learning-based CF method and matching function learning-based CF method [4]. Zhou

et al. initiate the first attempt to learn the distribution of user latent preference by proposing an adversarial POI recommendation model [34]. It is obvious that neural network actually embeds the input data into another new space by non-linear mapping, then to extract the core information hidden in feature for recommendation computing. Feng et al. believe that the dimension of embedding space will lose the high-level implicit information of POI and then designs a hyperbolic metric embedding approach to capture the behavior patterns of users [7]. Besides, Kim et al. propose an adaptive weighting scheme based on meta-learning that self-generates the meta-data via self-ensembling [14]. For modeling the interaction information, Wang et al. develop a neural graph-based collaborative filtering which exploits the user-item graph structure by propagating embedding on it [26]. In our previous work, we propose a POI model based on deep neural network and it also considers the geographical influence [32]. However, in this paper, we focus on how to apply deep learning for potential correlation mining between users and locations. Besides, Neu-PCM model we proposed also combines the deep matrix factorization. The related works mentioned above do not construct deep models like ours.

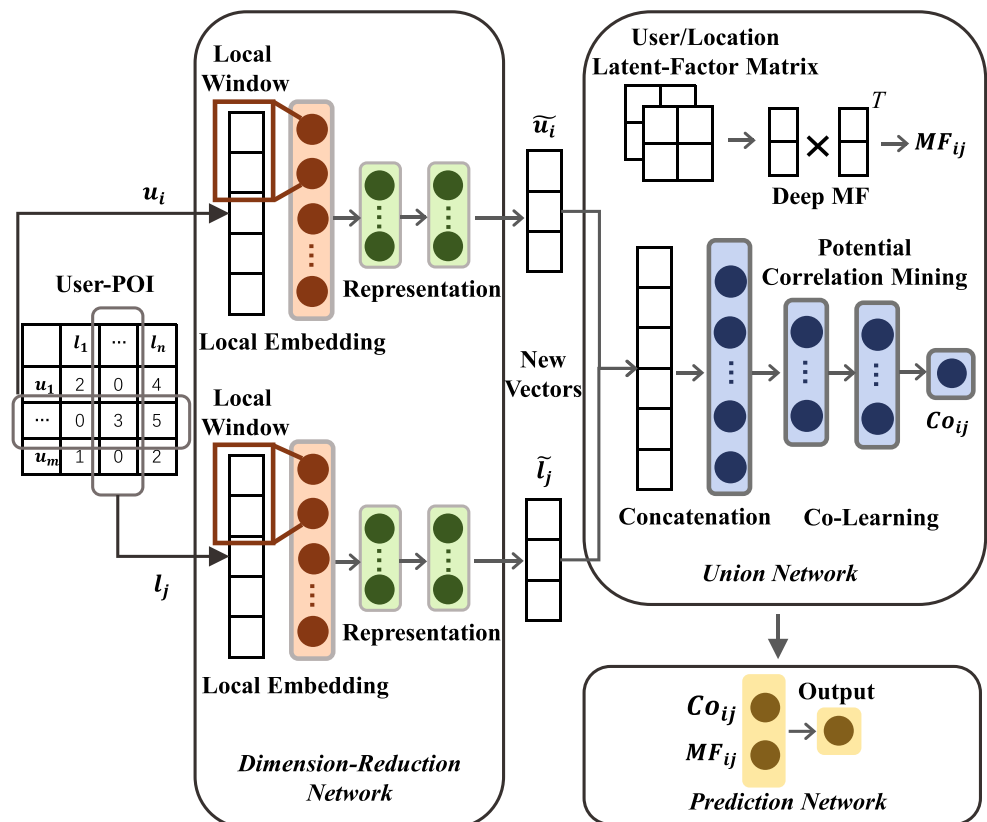
There are several obvious difference between our POI recommendation and sequence-based next location prediction. We focus on recommending a new specific

location that users has not visited, such as a nearby Chinese restaurants. Sequence-based next location prediction always explores the spatial-temporal information in terms of trajectory history, such as traffic-flow prediction [30] and tracking offender [10]. It is common that geo-aware sensors have been deployed in environmental detection which also adopt complex deep learning [17]. However, our deep model is designed for human activity and proposes deep matrix factorization as well as union network for mining the potential correlation, instead of capturing the geo-sequence pattern.

3 Neu-PCM: Neural-Based potential correlation mining for POI recommendation

In this section, we will introduce our deep neural-based potential correlation mining model for POI recommendation, which is also called Neu-PCM. The whole framework of this POI model is shown in Fig. 1. Firstly, we present local embedding and dimension-reduction networks to learn core information from original users vector and locations vector respectively. Secondly, we concatenate the new vectors to represent the unique pair of users and locations. Thirdly, we put it into a novel union network which is for mining potential correlation. Then, we build a deep matrix

Fig. 1 The Framework of Neu-PCM



factorization to further improve the performance by combining potential correlation with it. Finally, the output of our Neu-PCM model is the probability of user visiting location. The detail of each part will be presented in following subsections.

3.1 Problem formulation

POI recommendation aims to help users find new satisfactory locations according to their historical check-in data. In this paper, suppose that we have a set U of users $\{u_1, u_2, \dots, u_m\}$, a set L of locations $\{l_1, l_2, \dots, l_n\}$. All these users and locations form the User-POI matrix $\mathbf{R} : M \times N$ and \mathbf{R}_{ij} denotes the number of times that u_i has visited l_j . \mathbf{R} indicates the original explicit correlation between users and locations, which could be observed. We also could use 0 or 1 to replace the value in \mathbf{R} where it tells us whether users has visited the location or not.

The goal of POI recommendation model is to predict the probability of visiting an unobserved location from \mathbf{R} . If current users u_i has not visited the candidate location l_j before, the goal could be defined as follows:

$$\widehat{\mathbf{R}}_{ij} = Model(u_i, l_j | \theta) \tag{1}$$

$$Rec_{u_i} = \{l_j | sorted\ by\ \widehat{\mathbf{R}}_{ij}, K\} \tag{2}$$

where $\widehat{\mathbf{R}}_{ij}$ is the final prediction and θ is the set of all parameters in our model. Rec_{u_i} is the personal list of recommendations we produce by sorting predictions of all unvisited locations. K is the length of Rec_{u_i} . Hence, our task is to design a better POI model that will recommend satisfactory locations to users correctly.

3.2 Local embedding

The process of predicting unknown \mathbf{R}_{ij} based on deep learning is to mine the potential correlation from User-POI matrix. However, the original users vector or location vector is much sparser with many zeros, which will destroy the computing of recommendation task. Now, we present local embedding to capture core information on input in the first layer of our deep model.

Users input and locations input are denoted as \vec{u}_i and \vec{l}_j . In the first layer, we construct a local window instead of fully connected layer which is shown in Fig. 2. Local window will divide the input vector into different parts in terms of the size of the window. At the same time, the first layer will also be divided into different parts. Each part of input corresponds to a part of the first layer. In another word, there are many sub-networks in the first layer and each will concentrate more on local input to learn features sufficiently.

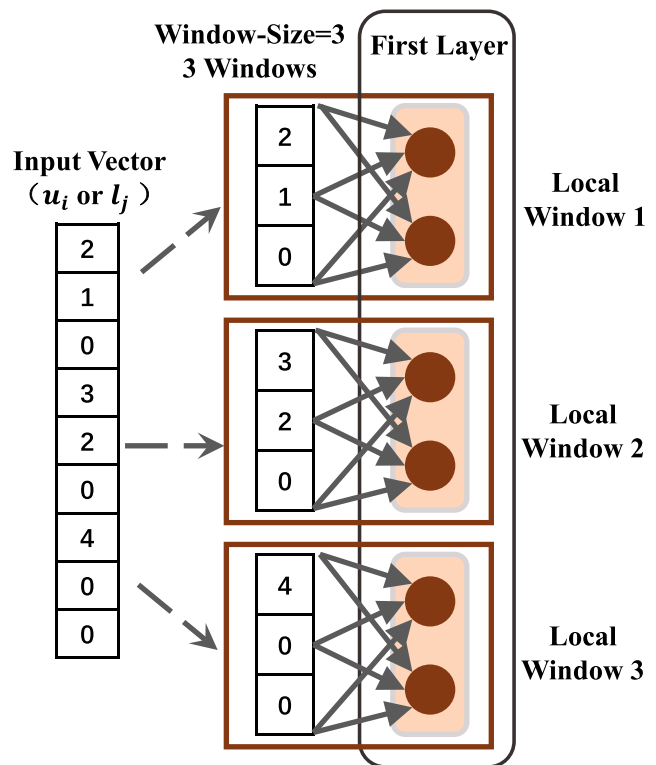


Fig. 2 Local Embedding

The input is the original users vector or location vector. The first local-embedding layer is shown as follows.

$$In = \vec{u}_i \text{ or } \vec{l}_j \tag{3}$$

$$Layer_1 = Con \left(f \left(\mathbf{w}_1^1 In_1 + b_1^1 \right), f \left(\mathbf{w}_1^2 In_2 + b_1^2 \right), \dots, f \left(\mathbf{w}_1^{w_n} In_{w_n} + b_1^{w_n} \right) \right) \tag{4}$$

where w_n is the number of local windows in terms of windows-size. \mathbf{w}_1^1 and b_1^1 are the weight matrix and bias of the first local window. \mathbf{w}_1^2 and b_1^2 are the weight matrix and bias of the second local window, and so on. Function Con links results of all local windows, which is the output of the first layer. Activation function f is chosen as follows.

$$f(x) = max(0, x) \tag{5}$$

Since there are much useless information in original input, the first local-embedding layer plays a significant role in our whole model. The local embedding makes each sub-network to concentrate more on local input.

3.3 Dimension-reduction network

Even local embedding learns information from original input, the dimension of vector is so large that will make the computing process cumbersome. We want to extract more useful high-level features and make vector have an appropriate dimension. Therefore, we construct

two dimension-reduction networks for both users and locations. The dimension-reduction network including local embedding has M layers, which is defined as follows.

$$Layer_k = f(\mathbf{w}_k Layer_{k-1} + b_k), k = 2, 3, \dots, M \quad (6)$$

where \mathbf{w}_k and b_k are the weight matrix and bias of corresponding layer. $Layer_k$ is the output of last layer. When $k = 2$, the $Layer_{k-1}$ is the output of local embedding. f is still chosen as ReLu function. Note that each layer has a smaller number of neural units than former layer to reduce the dimension. Each layer is able to capture more useful information from the former one. Hence, the final output of this network is denser than original vector with less dimensions. To avoid parameter redundancy, we adopt less than 4 layers for this network part.

We have mentioned before that neural network transforms the data into another new feature space. So the original distribution of data will be changed. In order to follow a constant distribution of data and mine the potential correlation hidden in data correctly, we adopt batch normalization [12] to dimension-reduction networks. As shown as follows, it works before activation function.

$$Layer_k = f(BN_k(\mathbf{w}_k Layer_{k-1} + b_k)) \quad (7)$$

$$BN_k(Neu_{mid}) = \gamma \left(\frac{Neu_{mid} - \mu(Neu_{mid})}{\sqrt{\sigma(Neu_{mid})^2 + \epsilon}} \right) + \beta \quad (8)$$

$$Neu_{mid} = \mathbf{w}_k^{neu} Layer_{k-1}^{neu} + b_k^{neu} \quad (9)$$

where BN_k is the batch normalization that sets the distribution of data and alleviates over-fitting to a certain extent. Neu_{mid} is the middle-output of a neuron before activation function. Its weight vector and bias are \mathbf{w}_k^{neu} and

b_k^{neu} . Function μ and σ calculate the average value and standard deviation of one training batch on this neuron. ϵ is a minimum to prevent denominator from being zero. γ and β are parameters of batch normalization that is for complementing non-linear ability of neural network. More details about batch-normalization are beyond the scope of our paper.

The dimension-reduction networks are shown in Fig. 3. The upper half part is for user feature vector and the lower half part is for location vector. After reducing dimension, we capture more high-level features and the new vectors could be applied into our recommendation task. The new vectors of users and locations are defined as follows.

$$\tilde{u}_i = f(\dots f(BN_k^U(\mathbf{w}_k^U f(BN_{k-1}^U(\mathbf{w}_{k-1}^U f(\dots) + b_{k-1}^U)) + b_k^U)) \dots) \quad (10)$$

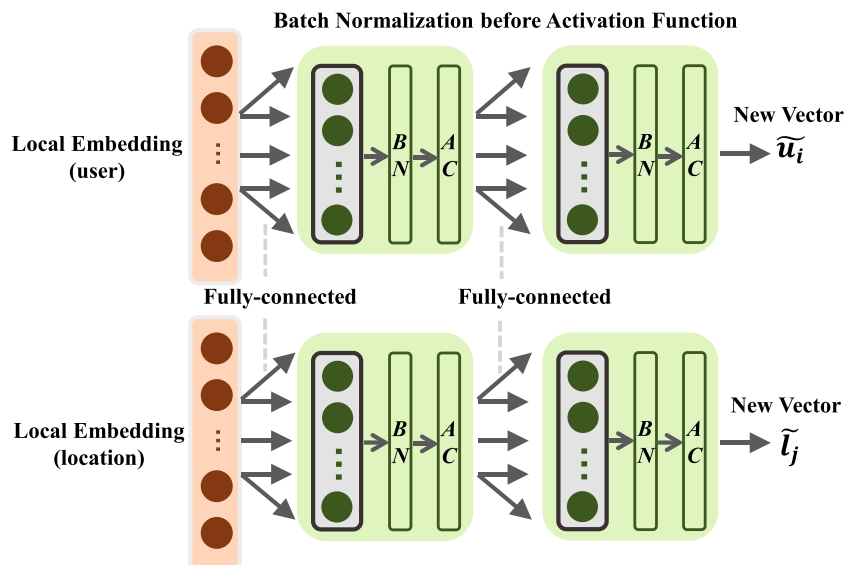
$$\tilde{l}_j = f(\dots f(BN_k^L(\mathbf{w}_k^L f(BN_{k-1}^L(\mathbf{w}_{k-1}^L f(\dots) + b_{k-1}^L)) + b_k^L)) \dots) \quad (11)$$

where BN_k^U , \mathbf{w}_k^U and b_k^U are the batch normalization, weight matrix and bias for dimension-reduction networks of users. BN_k^L , \mathbf{w}_k^L and b_k^L are batch normalization, weight matrix and bias for dimension-reduction networks of location. \tilde{u}_i and \tilde{l}_j are new vectors of users and location respectively and they indicate more core high-level information with lower-dimension features.

3.4 Union network for potential correlation mining

The purpose of this paper is to mine the potential correlation between users and locations for POI recommendation by neural network in deep learning. The original input of

Fig. 3 Dimension-Reduction Network



users and locations after reducing dimensions become \tilde{u}_i and \tilde{l}_j respectively. The new vectors indicate the core high-level information with less dimensions. Hence, they are suitable for next recommendation computing. Now we concatenate the vectors to represent the unique pair of users and locations, which is shown as follows.

$$\widetilde{ul}_{ij} = \tilde{u}_i \oplus \tilde{l}_j \tag{12}$$

where symbol \oplus is operation of concatenating. \widetilde{ul}_{ij} is the unique-pair vector and it will be applied into feature cross. Each user and each location will form only one unique vector that improves the accuracy of the correlation mining between users and locations. Based on \widetilde{ul}_{ij} , we build a union network, as the following shows.

$$Layer_k = f(\mathbf{w}_k Layer_{k-1} + b_k) \times \overrightarrow{D}_K, K = 1, 2, \dots, N \tag{13}$$

$$\overrightarrow{D}_K = drop(r), r \sim Bernoulli(p) \tag{14}$$

where the union network has N layers and \mathbf{w}_k and b_k are weight matrix and bias of it. Activation function is still chosen as ReLu. Since it is possible to face over-fitting when learning correlation, we adopt drop-out in our union network. \overrightarrow{D}_K is the drop-out vector that is filled with 0-1. $drop$ is the function to generate drop-out vector and each dimension depends on $Bernoulli(p)$ which outputs 1 according to the probability p . After union network, we obtain the potential correlation Co_{ij} . It reveals the implicit preference of users. The union network is shown in Fig. 4.

Considering that matrix factorization could improve performance of recommendation model based on deep learning, we define a user latent-factor matrix U^{LF} and a location latent-factor matrix L^{LF} to realize deep matrix factorization. The two latent-factor matrices will be trained in our whole deep model. The unknown item using deep matrix factorization is shown as follows.

$$MF_{ij} = U_i^{LF} L_j^{LF^T} \tag{15}$$

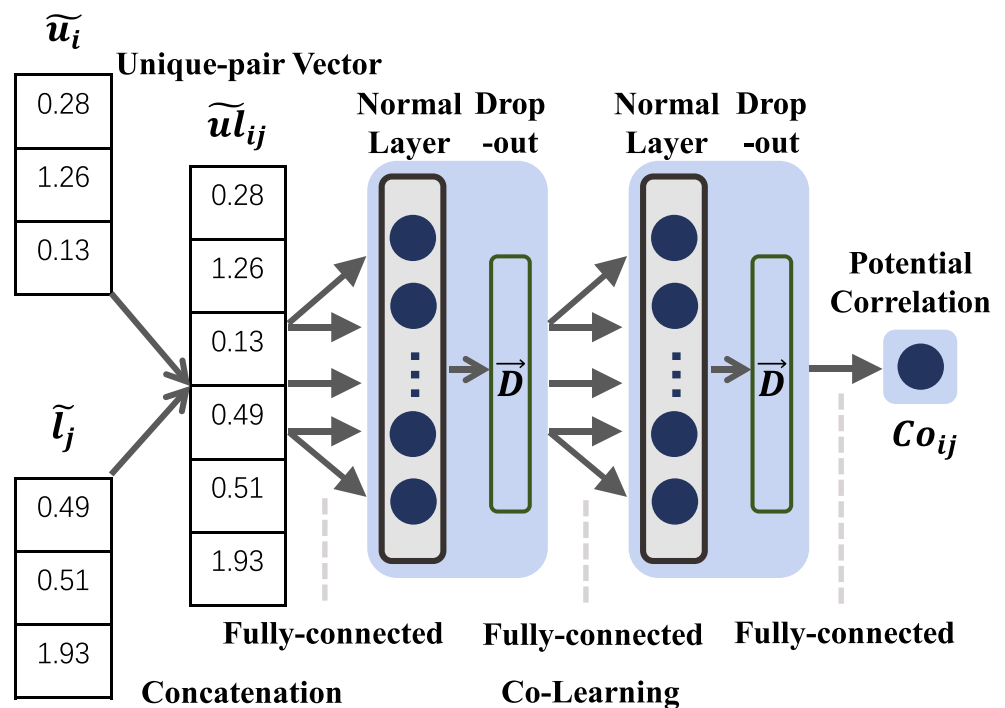
where U_i^{LF} and L_j^{LF} are corresponding row vectors of users and locations. Deep matrix factorization further enhances the result of potential correlation mining. The potential correlation Co_{ij} is a unique implicit preference and if we combine it with deep matrix factorization, its role will be displayed fully.

3.5 POI recommendation

Now, we have obtained Co_{ij} and MF_{ij} by potential correlation mining and deep matrix factorization. Co_{ij} tells us the implicit preference that is hidden under the original data, which is the core part of our deep model. MF_{ij} predicts the unknown item from matrix, which could be regarded as an improvement.

One measurement to realize deep matrix is to design a two-channel deep network which accepts original row and column from matrix and multiply result as one value, like our dimension-reduction network. However, in order to reduce parameters, we use an initial weight matrix to represent latent matrix, which is trained along with

Fig. 4 Union Network



our whole deep model. As shown in right-top part in Fig. 1, the two latent matrices will conduct the deep matrix factorization in our model and also be trained as part of loss.

In order to combine them and make final prediction of POI recommendation, as the following shows, we use a simple perceptron.

$$Pre_{ij} = f(w^{Co}Co_{ij} + w^{MF}MF_{ij} + b) \tag{16}$$

$$f(x) = 1 / (1 + e^{-x}) \tag{17}$$

where w^{Co} and w^{MF} are the weights of Co_{ij} and MF_{ij} respectively, b is the bias of the perceptron. Since we want to get the probability of users visiting location, the activation function is chosen as sigmoid. Actually, Pre_{ij} is the goal \hat{R}_{ij} in problem formulation. We sort all candidate locations in terms of their predictions and make a personal list of POI recommendation to complete our service. It is shown in Fig. 5.

A whole framework could be designed for pattern mining and supervised classification to form different applications of location predicting [3]. As for loss function which suits our POI recommendation framework, we adopt cross-entropy loss that is widely common in many deep learning models. It makes the core recommendation task become classification problem of whether users will visit the candidate location or not. The loss is defined as follows.

$$Loss = - \sum_{R_{ij} \in Batch} (I(R_{ij}) \log Pre_{ij} + (1 - I(R_{ij})) \log (1 - Pre_{ij})) \tag{18}$$

where $I(R_{ij})$ outputs 1 if l_j is a visited location, otherwise 0. R_{ij} belongs to one training batch which includes negative samples. We adopt mini-batch gradient descent for our training.

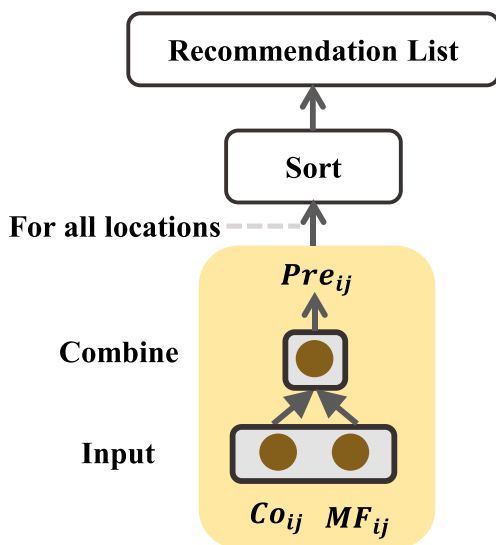


Fig. 5 POI Recommendation

Table 1 Description of Foursquare Datasets

City	Users	Locations	Check-ins	Average Check-in
Los Angeles	4747	7136	48461	10
Seattle	2381	6928	58052	24

4 Experiments

In this section, we will choose appropriate values for key parameters in our model, and then evaluate our model with some popular recommendation algorithms on two real-world datasets.

4.1 Datasets

We employ two real-world datasets¹ collected from two cities on Foursquare which is a location-based social application. One is Los Angeles, the other is Seattle. The descriptions of datasets are shown in Table 1. There are 48,461 check-ins made in Los Angeles and they are produced by 4,747 users on 7,136 locations. The average check-in of per user is 10. For Seattle dataset, there are 58,052 check-ins and they are made by 2,381 users on 6,928 locations. The average check-in of each user is 24. Obviously, the dataset of Los Angeles is sparser than that of Seattle. We randomly select 70% of the locations of each user as training data and the remaining 30% as test data. Moreover, for the sake of effectiveness of experiments, the users who have visited less than 5 locations and the locations that have been visited by less than 5 users are removed from datasets.

4.2 Evaluation metrics

Like most recommendation models, we use Normalized Discounted Cumulative Gain (NDCG) and Recall as evaluation metrics. Since the goal of POI recommendation is to correctly predict the potentially interested locations, and the results of the test data is the best illustration. If our model captures the preferences of users effectively, the Recall will be high. Moreover, sorting is an important and indispensable step when making recommendation list and NDCG should be taken into account. They are defined as follows.

$$NDCG@K = 1/m \sum_u DCG_u@K / IDC G_u@K \tag{19}$$

$$Recall@K = 1/m \sum_u |Rec_u \cap Test_u| / |Test_u| \tag{20}$$

where Rec_u is the recommendation list for users u and $Test_u$ is the test data of users u . K is the length of

¹<https://github.com/lab413/datasets>

recommendation list, which is set to 10, 20 and 30. ‘@’ means that results of evaluation metrics will vary according to different values of K . NDCG requires DCG and IDCG. IDCG is the ideal sorting result and DCG is the real sorting result. They are shown as follows.

$$DCG_u@K = \sum_z \frac{1}{\log(z + 1)} \tag{21}$$

$$IDCG_u@K = \sum_q^{|Test_u|} \frac{1}{\log(q + 1)} \tag{22}$$

where z is the ranking index of test POI data in our recommendation list. q represents and indicates the real index it should be. Generally speaking, since all test POI data for one use should be placed in top part of recommendation list, it is allowed to apply any order on them.

4.3 Experimental settings

We adopt mini-batch gradient descent to train our deep model since its cost of training time will be decreased

with a good performance. The batch size is 512. The initial learning rate is 0.001. If the training data is only filled with observed item, the deep model we propose will just learn the positive preferences which could not make correct prediction when facing a location that users do not like. Hence, we take some negative samples for each observed data. The size of each sampling is 2, 3, 4, 5 and 6. The window-size of local embedding will also have influence on our deep model. We set the window-size to 100, 200, 300, 400 and 500. To comprehensively demonstrate the effectiveness of our model, we compare it with the following recommendation algorithms.

POP: A basic model that only recommends popular POIs.

NMF: A classical non-negative matrix factorization.

DMF [27]: The deep matrix factorization that is designed for recommendation system, which learns a common low-dimensional space for users and recommendations.

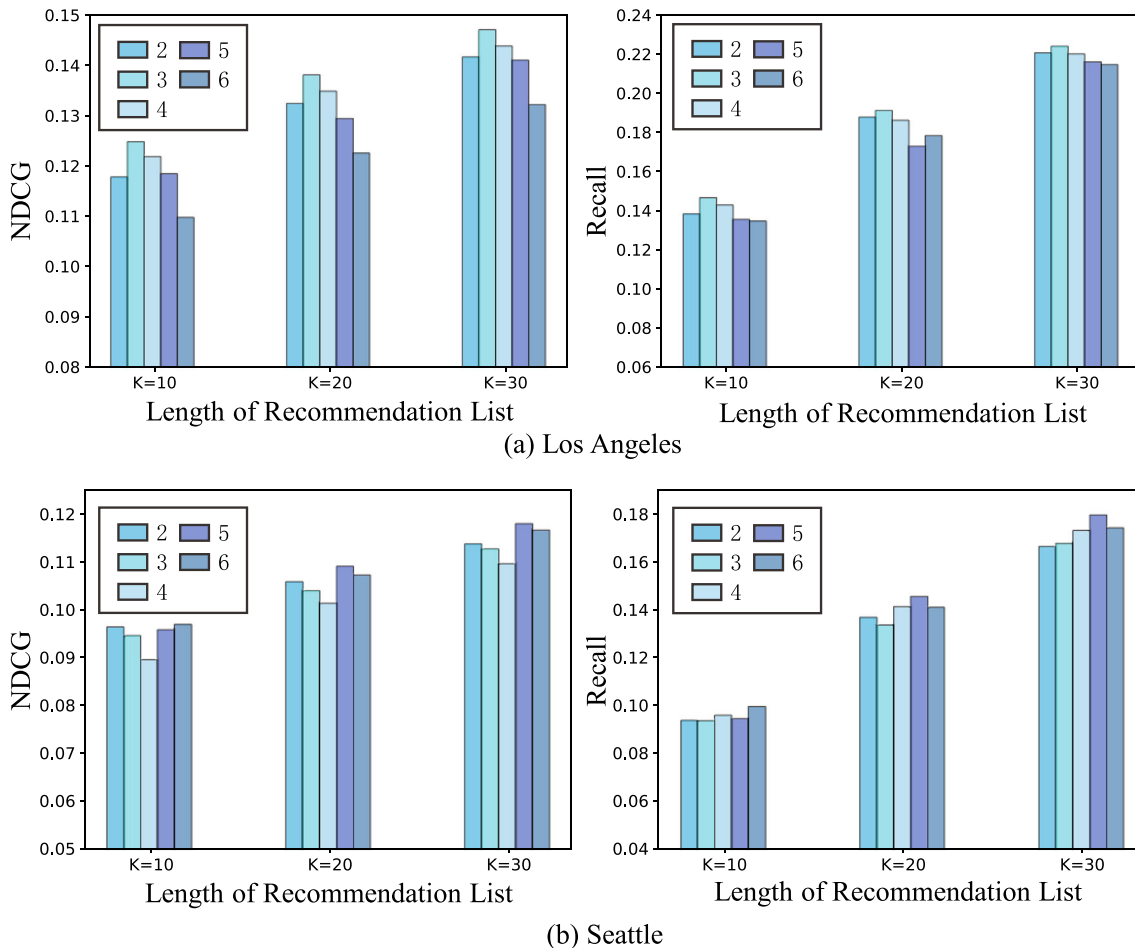


Fig. 6 Experiment on Negative-Sampling Size

NCF [9]: Neural collaborative filtering that is a general framework based on neural networks and could model the latent features of users and recommendation items.

FastHNBF [15]: This novel hierarchical negative binomial factorization models data dispersion via a hierarchical Bayesian structure, thus alleviating the effect of data overdispersion to help with performance gain for recommendation.

POP is a normal baseline in most recommendation system. It is still common in our daily mobile applications which have service of recommending. DMF and NCF are representative work of different types of recommendation based on deep neural network. Our deep model is inspired by DMF and NCF. Moreover, NCF is a highly-cited model which proposes a novel deep framework and is adopted in many papers. FastHNBF is a novel recently-published recommendation model based on hierarchical Bayesian structure.

In addition, since there are two components to achieve the recommendation goal, we add ablation experiment to

prove the effectiveness of them. One is potential correlation mining, which is the core in our model, and the other is deep matrix factorization that aims to enhance the former part. Meanwhile, the combination of them, conducting our final model, will also be compared with them.

4.4 Results

The experiment of negative sampling is shown in Fig. 6. For Los Angeles dataset, the optimal negative-sampling size is 3. For Seattle dataset, the optimal negative-sampling size is 5.

Generally speaking, with the increase of K , the evaluation matrices of both datasets will also increase. This is because we expand the range of recommending locations and it will bring more chance to hit the locations that users are interested in. In Los Angeles dataset, the NDCG of all different K begin to decline significantly after the negative-sampling size reaches 3. The Recall is the same. However, while $K = 20$, size 6 is a little better than size 5 and it is still not outperform size 3. Hence, it does necessary to set negative-sampling size to 3 for Los Angeles dataset since

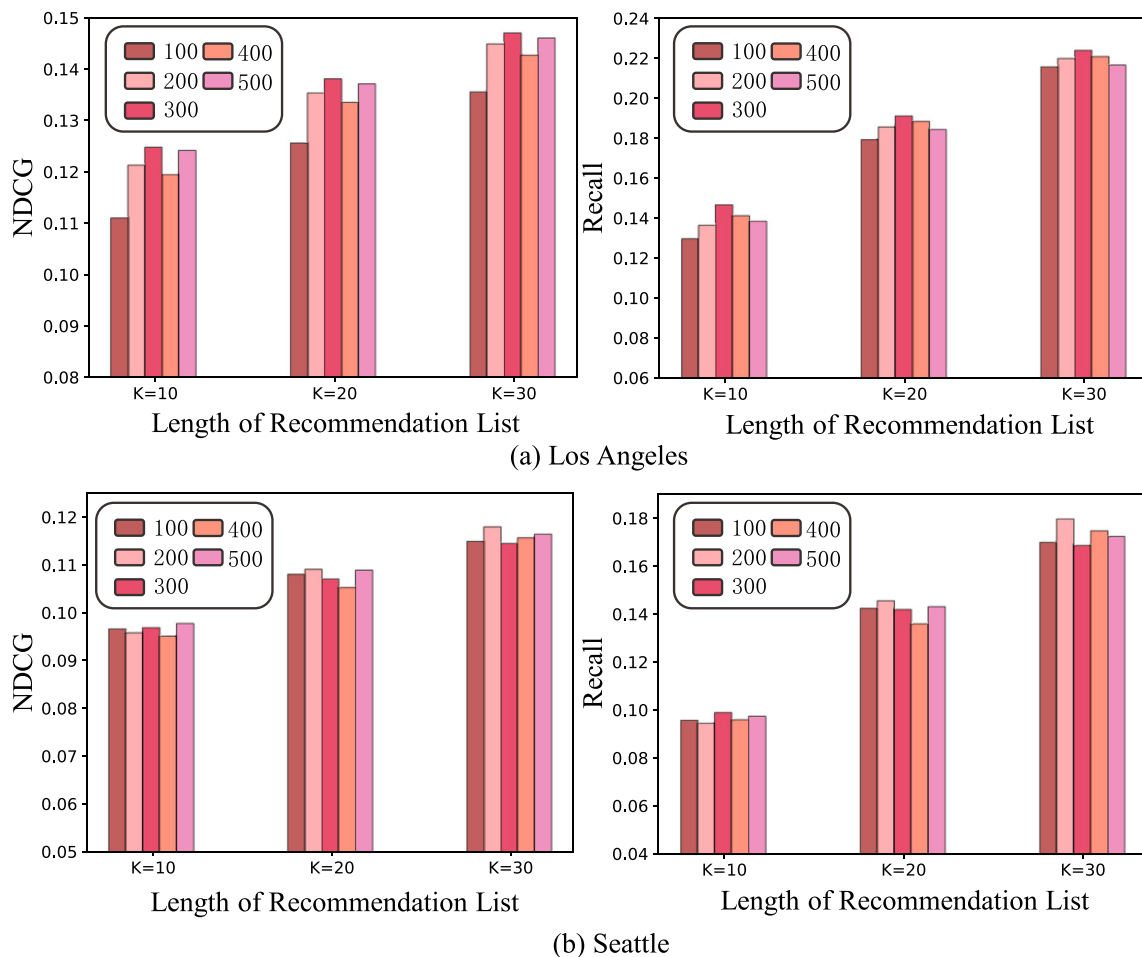


Fig. 7 Experiment on Window-Size

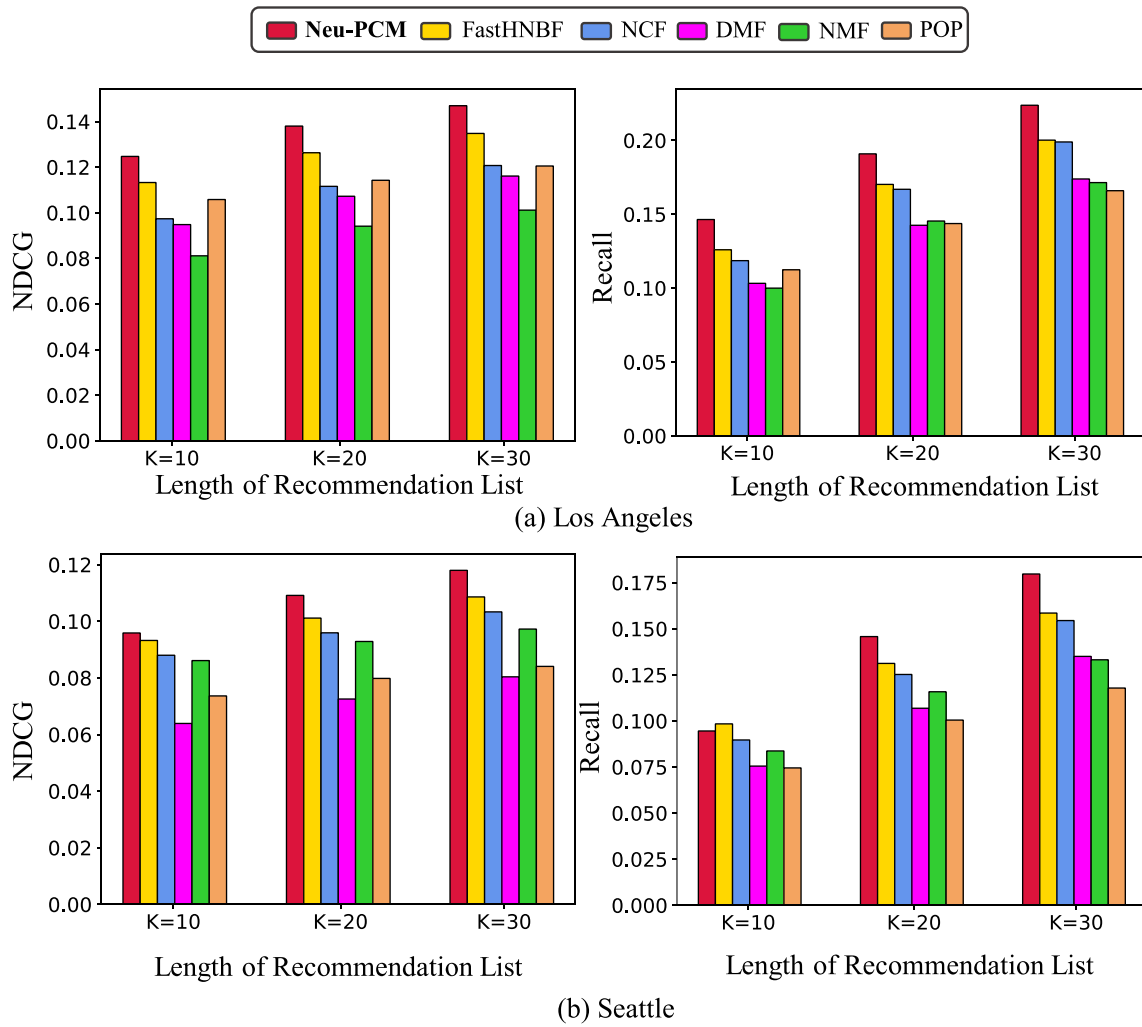


Fig. 8 Experiment on Performance Comparison

there is no need to capture more negative preferences on users of that city.

In Seattle dataset, when negative-sampling size increases from 2 to 4, the NDCG of all K shows downward trend. While size reaches 5, NDCG goes up obviously. Size 5 is the best situation of $K = 20$ and $K = 30$, which outperforms all other sizes. Moreover, for Recall, the best size is 6 when $K = 10$, and size 5 is most suitable for

other K . In a word, more negative-sampling size could help to improve the performance of our deep model on Seattle dataset. Therefore, we set its negative-sampling size to 5 that is optimal in most cases.

The experiment of window size is shown in Fig. 7. For Los Angeles, the optimal window-size of local embedding is 300. For Seattle dataset, the optimal window-size is 200.

Table 2 Contrast Experiments on Los Angeles Dataset

Model	NDCG@10	NDCG@20	NDCG@30	Recall@10	Recall@20	Recall@30
Neu-PCM	0.12470	0.13801	0.14695	0.14643	0.19090	0.22383
FastHNBF	0.113252	0.126309	0.134801	0.125944	0.170241	0.200182
NCF	0.09737	0.11159	0.12073	0.11863	0.16691	0.19899
DMF	0.09479	0.10724	0.11608	0.10324	0.14254	0.17390
NMF	0.08111	0.09410	0.10116	0.09998	0.14543	0.17146
POP	0.10581	0.11425	0.12052	0.11241	0.14372	0.16597

The bold entries are best performances in corresponding conditions

Table 3 Contrast Experiments on Seattle Dataset

Model	NDCG@10	NDCG@20	NDCG@30	Recall@10	Recall@20	Recall@30
Neu-PCM	0.09587	0.10914	0.11801	0.09458	0.14576	0.17974
FastHNBF	0.09322	0.10112	0.10862	0.09842	0.13119	0.15855
NCF	0.08801	0.09594	0.10333	0.08968	0.12521	0.15445
DMF	0.06391	0.07254	0.08038	0.07550	0.10689	0.13503
NMF	0.08613	0.09285	0.09724	0.08372	0.11583	0.13318
POP	0.07364	0.07981	0.08407	0.07451	0.10046	0.11784

The bold entries are best performances in corresponding conditions

In Los Angeles dataset, as the window-size increases from 100 to 300, the NDCG of all K goes up and reaches peak at size 300. Size 500 is better than size 400. For Recall, all situations under different K indicate the same pattern. They first go up and then go down significantly. Size 300 makes the best Recall for each situation. Hence, we set window-size to 300 for Los Angeles since it is appropriate and will not be too large to lead a wide-scale learning or too small to make embedding limited. In Seattle dataset, the NDCG and the Recall both have fluctuation under different K . When $K = 10$, size 500 makes NDCG best but when $K = 20$ or $K = 30$, size 200 is the optimal choice. Meanwhile, the Recall under size 300 reaches peak of $K = 10$. Size 200 makes Recall best of $K = 20$ or $K = 30$. Therefore, in terms of avoiding large-scale learning, we set window-size to 200 for Seattle dataset. There are many differences between datasets and the size 200 is enough.

Based on all optimal parameters of Los Angeles dataset and Seattle dataset, we conduct experiments to compare our model with some popular recommendation algorithms. Our deep model is called Neu-PCM. The result is shown in Fig. 8. In a word, Neu-PCM outperforms other models.

In Los Angeles dataset, POP has a better NDCG than NCF and DMF, which tells us sorting locations according to their popularity is suitable for this famous city. However, expect $K=10$, the Recall of POP is not relatively high and it may be much limited to recommend locations only based on popularity. DMF depends on the common space between users and locations, but it uses cosine similarity to calculate the final prediction. NMF is the worst model, which indicates original matrix factorization could not capture the preferences of users correctly. Except our model,

FastHNBF is superior to others, which demonstrates it has the ability to alleviate the effect of data overdispersion to help with performance for recommendation system, while it outperforms NCF slightly in terms of Recall.

In Seattle dataset, the NDCG of NMF is just inferior to NCF and better than other comparison models, which is most likely due to the differences between datasets. Meanwhile, DMF has the worst NDCG, which shows its ability based on deep learning to sort locations is weak. POP has the worst Recall and it could not hit the locations that users will be interested in. NCF that combines matrix factorization and collaborative filtering by neural networks has excellent performance on both datasets. On the other hand, the Recall of FastHNBF is only a little better than our model when $K=10$, which is an exception since our model outperforms it in terms of both NDCG and Recall for all other situations.

In short, The Neu-PCM we propose outperforms other popular recommendation algorithms on Los Angeles dataset and Seattle dataset. Hence, our Neu-PCM could be applied into POI recommendation system to complete location-based services. The Tables 2 and 3 show the real values of Recall and NDCG, which corresponds to the Fig. 8 and has been performed for statistical analysis.

The result of ablation experiments are shown in Tables 4 and 5. ‘No’ means ‘without’. Hence, No-PCM only includes the simple deep matrix factorization and No-MF only contains our core potential correlation mining. Full-Comb is our final model which consists of those two parts. In generally speaking, it is the PCM which has significant influence on both datasets. Deep MF enhances the recommendation result by its unique measurement.

Table 4 Ablation Experiments of Neu-PCM on Los Angeles Dataset

Model	NDCG@10	NDCG@20	NDCG@30	Recall@10	Recall@20	Recall@30
No-PCM	0.02056	0.02736	0.03164	0.03053	0.04987	0.06451
No-MF	0.10879	0.11904	0.12657	0.12166	0.15793	0.18559
Full-Comb	0.12470	0.13801	0.14695	0.14643	0.19090	0.22383

The bold entries are best performances in corresponding conditions

Table 5 Ablation Experiments of Neu-PCM on Seattle Dataset

Model	NDCG@10	NDCG@20	NDCG@30	Recall@10	Recall@20	Recall@30
No-PCM	0.02711	0.03311	0.04201	0.02700	0.04668	0.07557
No-MF	0.08112	0.08771	0.09371	0.08457	0.11592	0.13924
Full-Comb	0.09587	0.10914	0.11801	0.09458	0.14576	0.17974

The bold entries are best performances in corresponding conditions

However, the Recall and NDCG of No-PCM in Seattle dataset are quite larger than those in Los Angeles dataset, which indicates the deep MF are more suitable to improve our deep model on that dataset. Although core PCM cannot be replaced and deep MF could be regarded as an optional choice, the numbers of Recall and NDCG still increase to a certain extent. So, our final model Full-Comb combines them and then achieves better result. This demonstrates it is accurate to conduct those deep parts for our Neu-PCM, and the core PCM that contributes a lot is worthy to be applied to other fields.

5 Conclusion

Existing classical POI recommendation methods face the challenge where they could not capture the preference of users deeply and effectively. Meanwhile, they are also suffered from the problem of data sparsity. Hence, in this paper, we propose a deep neural-based potential correlation mining model for POI recommendation, which is called Neu-PCM. Firstly, we present local embedding and dimension-reduction network for core high-level information hidden under the original user-POI matrix. Secondly, we mine the potential correlation between users and locations for capturing the implicit preference of users. Thirdly, we construct a deep matrix factorization to combine potential correlation with it for further improving Neu-PCM. Finally, we compare our model with some popular recommendation algorithms and the results demonstrate Neu-PCM outperforms them. In the future, we will take some contexts of users or locations into Neu-PCM since we have already realized the POI model based on deep learning. There are many methods to transform original contexts into the inputs that our Neu-PCM could accept and we will find how different contexts influence the Neu-PCM. Meanwhile, the real-world check-in datasets we use in this paper do not occupy much storage and memory, we will use richer datasets and compare the computational cost of different hyper-parameters to further improve the model performance.

Acknowledgements This research is sponsored by Natural Science Foundation of Chongqing, China (No. cstc2020jcyj-msxmX0900) and

the Fundamental Research Funds for the Central Universities (Project No. 2020CDJ-LHZZ-040).

References

1. Chang B, Jang G, Kim S, Kang J (2020) Learning graph-based geographical latent representation for point-of-interest recommendation. In: Proceedings of the 29th ACM International conference on information & knowledge management, pp 135–144
2. Chen J, Lian D, Zheng K (2019) Improving one-class collaborative filtering via ranking-based implicit regularizer. In: Proceedings of the AAAI Conference on artificial intelligence, vol 33, pp 37–44
3. Comito C (2020) Next: a framework for next-place prediction on location based social networks. *Knowl Based Syst* 204:106205
4. Deng ZH, Huang L, Wang CD, Lai JH, Yu PS (2019) Deepcf: a unified framework of representation learning and matching function learning in recommender system. In: Proceedings of the AAAI Conference on artificial intelligence, vol 33, pp 61–68
5. Duan Z, Wang Y, Ye W, Fan Q, Li X (2022) Connecting latent relationships over heterogeneous attributed network for recommendation. *Appl Intell*, pp 1–19
6. Ebesu T, Shen B, Fang Y (2018) Collaborative memory network for recommendation systems. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp 515–524
7. Feng S, Tran LV, Cong G, Chen L, Li J, Li F (2020) Hme: a hyperbolic metric embedding approach for next-poi recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on research and development in information retrieval, pp 1429–1438
8. He X, He Z, Du X, Chua TS (2018) Adversarial personalized ranking for recommendation. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp 355–364
9. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th International conference on World Wide Web, pp 173–182
10. Huang C, Zhang C, Zhao J, Wu X, Yin D, Chawla N (2019) Mist: a multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In: The World Wide Web conference, pp 717–728
11. Hui B, Zhang L, Zhou X, Wen X, Nian Y (2022) Personalized recommendation system based on knowledge embedding and historical behavior. *Appl Intell* 52(1):954–966
12. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456. PMLR
13. Ji S, Feng Y, Ji R, Zhao X, Tang W, Gao Y (2020) Dual channel hypergraph collaborative filtering. In: Proceedings of the 26th

- ACM SIGKDD International conference on knowledge discovery & data mining, pp 2020–2029
14. Kim M, Song H, Kim D, Shin K, Lee JG (2021) Premere: Meta-reweighting via self-ensembling for point-of-interest recommendation. In: Proceedings of the AAAI Conference on artificial intelligence, vol 35, pp 4164–4171
 15. Kuo LY, Chen MS (2021) Hierarchical negative binomial factorization for recommender systems on implicit feedback. In: Proceedings of the AAAI Conference on artificial intelligence, vol 35, pp 4181–4188
 16. Li S, Zhou J, Xu T, Liu H, Lu X, Xiong H (2020) Competitive analysis for points of interest. In: Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining, pp 1265–1274
 17. Liang Y, Ke S, Zhang J, Yi X, Zheng Y (2018) Geoman: Multi-level attention networks for geo-sensory time series prediction. In: Proceedings of the Twenty-Seventh international joint conference on artificial intelligence, pp 3428–3434
 18. Liu T, He Z (2022) Dlr: A deep learning-based initialization recommendation algorithm for trust-aware recommendation. Appl Intell, pp 1–12
 19. Liu Y, Yang Z, Li T, Wu D (2022) A novel poi recommendation model based on joint spatiotemporal effects and four-way interaction. Appl Intell 52(5):5310–5324
 20. Shi H, Chen L, Xu Z, Lyu D (2019) Personalized location recommendation using mobile phone usage information. Appl Intell 49(10):3694–3707
 21. Si Y, Zhang F, Liu W (2019) An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features. Knowl Based Syst 163:267–282
 22. Sun K, Qian T, Chen T, Liang Y, Nguyen QVH, Yin H (2020) Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation. In: Proceedings of the AAAI Conference on artificial intelligence, vol 34, pp 214–221
 23. Tran T, Lee K, Liao Y, Lee D (2018) Regularizing matrix factorization with user and item embeddings for recommendation. In: Proceedings of the 27th ACM International conference on information and knowledge management, pp 687–696
 24. Wang J, Zhu L, Dai T, Xu Q, Gao T (2021) Low-rank and sparse matrix factorization with prior relations for recommender systems. Appl Intell 51(6):3435–3449
 25. Wang Q, Zhang M, Zhang Y, Zhong J, Sheng VS (2022) Location-based deep factorization machine model for service recommendation. Appl Intell, pp 1–20
 26. Wang X, He X, Wang M, Feng F, Chua TS (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on research and development in information retrieval, pp 165–174
 27. Xue HJ, Dai X, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: Twenty-sixth international joint conference on artificial intelligence, pp 3203–3209
 28. Yan C, Chen Y, Wan Y, Wang P (2021) Modeling low-and high-order feature interactions with fm and self-attention network. Appl Intell 51(6):3189–3201
 29. Yang C, Bai L, Zhang C, Yuan Q, Han J (2017) 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, pp 1245–1254
 30. Yao H, Liu Y, Wei Y, Tang X, Li Z (2019) Learning from multiple cities: a meta-learning approach for spatial-temporal prediction. In: The World Wide Web conference, pp 2181–2191
 31. Zeng J, He X, Tang H, Wen J (2021) Predicting the next location: a self-attention and recurrent neural network model with temporal context. Trans Emerg Telecommun Technol 32(6):e3898
 32. Zeng J, Tang H, Wen J (2020) Dpr-geo: a poi recommendation model using deep neural network and geographical influence. In: International conference on neural information processing, pp 420–431
 33. Zeng J, Tang H, Zhao Y, Gao M, Wen J (2021) Pr-rcuc: a poi recommendation model using region-based collaborative filtering and user-based mobile context. Mobile Networks and Applications, pp 1–11
 34. Zhou F, Yin R, Zhang K, Trajcevski G, Zhong T, Wu J (2019) Adversarial point-of-interest recommendation. In: The World Wide Web conference on World Wide Web (WWW), pp 3462–34618
 35. Zhou X, Mascolo C, Zhao Z (2019) Topic-enhanced memory networks for personalised point-of-interest recommendation. In: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, pp 3018–3028

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Jun Zeng received the BS and MS degree in software engineering from Chongqing University in 2007 and 2010. He received Ph.D. degree in advanced information technology of Kyushu University in 2013. Since 2013, he has been an associate professor in the school of big data and software engineering of Chongqing University. His research interest includes data mining, search engine and service computing.



Haoran Tang received the BS and MS degree in software engineering from Chongqing University in 2018 and 2021. He had worked in Microsoft (China) as a software engineer. Now, he is doing his Ph.D Programme from 2022 to 2025 in the Hong Kong Polytechnic University. His research interest includes recommendation system, data mining and artificial intelligence.



Yizhu Zhao received the BS degree in computer science and technology from Gansu Agricultural University in 2020. Currently, she is a Master Degree Candidate at the School of Big Data and Software Engineering, Chongqing University. Her research interest includes recommendation system, mobile user trajectory mining and data mining.



Junhao Wen received PhD degree in computer science from Chongqing University in 2005. He is a professor and dean at the School of Big Data and Software Engineering, Chongqing University. His research interests include recommendation systems, social computing and service computing.