IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

ST-RNet: A Time-aware Point-of-interest Recommendation Method based on Neural Network

Lu Gao, Yuhua Li*, Ruixuan Li, Zhenlong Zhu, Xiwu Gu, Olivier Habimana
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
* Corresponding Author Email: idcliyuhua@hust.edu.cn

***Abstract*—Point-of-interest (POI) recommendation is one of the most important services in the rapid growing location-based social networks (LBSNs). Good POI recommendation can help people explore the locations they haven't visited but are interested in, and help merchants find their target users. Time-aware POI recommendation aims to recommend unvisited POIs for a given user at a specified time in a day. However, previous methods, such as user-based collaborative filtering, lack the mining of the features of POIs and the learning of abstract spatio-temporal interactions. In this paper, we propose a novel time-aware POI recommendation method named ST-RNet (Spatio-Temporal Recommender Network) to address these shortages. ST-RNet works in the following fashion. Firstly, we analyze the crucial features in LBSNs to alleviate data sparsity problem and further measure the similarities between POIs. For subsequent network training, we then construct the embedding matrices with same dimension for users and POIs by POI-based Collaborative Filtering (PCF). Furthermore, the positive and negative check-in records are fed into a novel recommender neural network (RNet) to learn the embedding matrix of times and the abstract interactions between users, POIs and times. Finally, ST-RNet recommends the unvisited POIs most likely to be visited to a given user at a given time. The experimental results on Foursquare real-world dataset show that ST-RNet is effective on time-aware POI recommendation task and is capable of analyzing the hidden patterns behind spatio-temporal interactions.**

***Index Terms*—time-aware POI recommendation, spatio-temporal interactions, collaborative filtering, matrix factorization**

I. Introduction

With the rapid development of social networks and the presence of Global Position System (GPS), modern citizens tend to share their daily life and traveling experiences in social network platforms in the form of check-in. Thus, it makes the birth of location-based social networks (LBSNs), such as Foursquare, Yelp, Gowalla and so on. Fig. 1 illustrates a typical check-in example in LBSNs. Point-of-interest (POI), the most crucial part of LBSNs, is defined as an identified place (e.g. a restaurant or a cinema) that people may be interested to check in. The aim of POI recommendation is to recommend unvisited POIs to the specific user according to his/her interest, habit and several other context information in LBSNs.

To address the time regularity and check-in periodicity, [1] first proposed time-aware POI recommendation task, a significant case in POI recommendation field. According to our life experiences, users' check-in behaviors always vary with time in a day, and these behaviors are cyclical to some extent. For example, people may regularly stay at workplace at 14:00-18:00, but engage in entertainment activities at 20:00-23:00. Therefore, recommending unvisited POIs for a given user according to a certain time is necessary and valuable for better user experience.

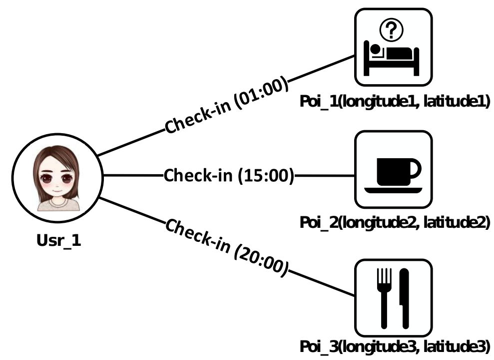


Fig. 1: A typical check-in example of a user in LBSNs

For the past few years, several techniques used in traditional POI recommender systems have been exploited for time-aware POI recommendation, where User-based Collaborative Filtering (UCF) is the most commonly used method [1], [2]. UCF firstly calculates the similarities between users pairwise and then recommends a given user these POIs visited by users similar to him/her at a given time. Good at learning latent factors of users and POIs, Matrix Factorization (MF) has achieved good results [3], [4], too. In addition, to create more effective model, recent works [5]–[10] gradually start to combine context information in LBSNs, such as social relationship and geographical coordinates.

However, the feature fusion models of most existing works are simple and lack the learning of abstract interactions between user features, POI features and time features. Besides, considering users' check-in records solely by UCF suffers severe data sparsity problem, which is one of the hardest problems in recommender systems. In order to address the shortages of existing methods and comprehend the check-in regularity in different time, we propose a novel time-aware POI recommendation method called Spatio-Temporal

978-1-7281-2009-6/$31.00 ©2019 IEEE

Recommender Network (ST-RNet), which is designed to learn the hidden and high-order patterns of user's check-in behaviors effectively.

In ST-RNet, based on data analysis, we start to propose three hypotheses to understand the features of POIs and measure the similarities between POIs. Then two embedding matrices of users and POIs are constructed based on POI-based Collaborative Filtering (PCF). Moreover, we design a recommender network (RNet) which takes a check-in triple as input and the occurrence possibility of the input triple as output. RNet is capable of learning the spatio-temporal interactions in historical check-in records and recommending unvisited POIs for a given user at a given time.

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

1. Based on Deep Learning, ST-RNet is proposed to learn the abstract and high-order patterns behind check-in interactions between users, POIs and time. To the best of our knowledge, this is the first work to adopt neural network for time-aware POI recommendation task.

2. By taking multi-dimensional features of POIs into account, a PCF method is applied to embed users and POIs into a same latent space innovatively, while existing works usually employ UCF.

3. A novel recommender network (RNet) is designed for effective POI recommendation, by learning the cross features and the combined features of users, POIs and times together.

4. We conduct comprehensive experiments on the real-world dataset to evaluate the performance of ST-RNet. The results show that ST-RNet is effective on time-aware POI recommendation task.

The remainder of this paper is organized as follows. Section II reviews the related works. Section III defines the time-aware POI recommendation task and provides the details of proposed ST-RNet, and then Section IV analyzes the experimental results. Finally, Section V concludes this paper and presents future works.

II. Related Work

In this section, we introduce the related works of POI recommendation and time-aware POI recommendation, and then give the perspective that Deep Learning is applicative in time-aware POI recommendation.

A. POI recommendation

Compared with the traditional recommendation field, check-in records in LBSNs can not only bring implicit feedback, but also contain multiple context information. These context information can be mainly categorized into three types: temporal factor, spatial factor and social factor.

For users, Social-based Collaborative Filtering (SCF) [5] proved that social relationships online are effective in POI recommendation. However, according to the analysis in [7], relationships in LBSNs only have weak influence on check-in behaviors in real life, because friends online not always have similar range of activity or similar interest. Therefore, in recent years, some scholars are devoted to redefine relationships in LBSNs [6], [8].

For POIs, spatial coordinates are the most important context information. From daily life experience, we know that even if a user is highly interested in a POI which is far from his usual range of activity, he will not visit this POI and may prefer to choose a similar POI nearby as an alternative. Considering other factors without spatial factor may lead to a large deviation in the recommendation results. Therefore, in POI recommendation, spatial factor is always considered in large number of works [4]–[8], [10]. Power-law distribution [7], Gaussian distribution [6] and Kernel density estimation [10] are widely used to model POI's spatial feature. Moreover, many modified MF [4] were proposed by combining the impact of spatial feature and achieved good results.

B. Time-aware POI recommendation

As we known, users' interest and check-in behaviors are different at different time. Observing that the check-in behaviors of users are regular and cyclical due to the limitation of activity range, [11] further confirmed the influence of temporal factor on POI recommendation.

Time-aware POI recommendation was firstly proposed in [1]. [1] analyzed the similarities between time in a day, and employed UCF to recommend POIs to a given user at a specified time. Geographical-Temporal influences Aware Graph (GTAG) [12] was further proposed by constructing an undirected graph of check-ins, and mining the POIs most likely to be visited based on Breadth-first Search (BFS). Recently, UCF-based UPT [2] took the popularity of POIs into account to improve recommendation performance.

Matrix Factorization [3], [13] and Tensor Factorization [14] also achieved excellent results in time-aware POI recommendation. The Ranking-based Geographical Factorization Method (Rank-GeoFM) [3] considered spatial and temporal feature as the latent factor in MF to incorporate them into check-in interactions, and then predicted the POIs most likely to be visited based on ranking metric. Moreover, [14] proposed a fourth-order tensor factorization-based ranking methodology to recommend users the point-of-interest by considering their time-varying behavioral trends in long-term and short-term simultaneously. The Spatio–Temporal Distance Metric Embedding model (ST-DME) [15] was applied to time-specific POI recommendation to model users' time-specific preferences effectively.

However, most of these works fuse multiple features with a simple way, which causes the problem in learning the abstract interactions between users, POIs and times. Neural Networks have been proved to be effective of learning non-linear and abstract patterns in these years, especially in Natural Language Processing and Image Recognition fields. Therefore, in this paper, we aim to design an effective network method to address time-aware POI recommendation problem.

C. Deep Learning based POI recommendation

With the rapid development of Deep Learning, neural networks have gradually used in recommender systems [16], [17] and achieved satisfactory results. For example, [18], [19] both adopted Recurrent Neural Network (RNN) in session-based recommendation network, [17] employed the Long Short-Term Memory (LSTM) in film recommendation, and [20] addressed the music recommendation task with the heterogeneous information graph embedding. In addition, an universal Neural Collaborative Filtering (NCF) framework [21] directly inputted user and item characteristics into Deep Neural Network (DNN) and conducted non-linear modeling for interactions.

In recent years, Deep Leaning also has some applications in the field of POI recommendation [9], [22]–[25]. To take full advantage of multi-dimensional information in LBSNs, PACE [23] incorporated check-in records learning and context learning into a semi-supervised recommender framework to alleviate the data sparsity in POI recommendation. Besides, [25] firstly decomposed user and POI features into the same latent space with Non-negative Matrix Factorization (NMF). Visual Content Enhanced POI recommendation (VPOI) [24] exploited CNN to extract features from figure comments and gave recommendations based on basic Probability Matrix Factorization (PMF).

However, most of above works with Deep Learning only address the traditional personalized POI recommendation or Next-POI recommendation task, without considering the periodicity of check-in behaviors specifically. Time-aware POI recommendation is valuable and crucial in recognition of human periodic behaviors. To deal with these shortages, the ST-RNet proposed in this paper focuses on temporal and spatial features in LBSNs, and explores the check-ins' periodicity in a day.

III. Proposed Method

In this section, we first define the problem of time-aware POI recommendation and list key notations used in this paper in Table I. Then we discuss the details of the proposed method Spatio-Temporal Recommender Network (ST-RNet), which includes three main components: (i) Measuring similarities between POIs based on multiple features in LBSNs, (ii) Constructing the embedding matrices of same dimension for users and POIs with PCF, (iii) Feeding positive and negative $\langle u_i, l_j, t_h \rangle$ triples into recommender network (RNet) for training and recommendation.

A. Problem Definition

The task of time-aware POI recommendation is formally defined as follows. Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users in LBSNs, $L = \{l_1, l_2, \ldots, l_n\}$ be the set of POIs, and $T = \{t_1, t_2, \ldots, t_s\}$ be the set of times. In this paper, time means one specified time slice divided by a day with concrete granularity. We use a triple $\langle u_i, l_j, t_h \rangle$ to represent a check-in record, which denotes that user i visited POI j at time h. The aim of time-aware POI recommendation is to recommend POIs most likely to be visited to a given user at a specified time, by learning the patterns behind historical check-ins in LBSNs.

TABLE I: Key notations used in this paper

Notation	Interpretation
L_u	The set of POIs that user u has visited
S, C, P	The Spatial, Co-visit and Popularity matrix of POIs in LBSNs
e_u, e_l, e_t	The embedding representation of a user, a POI and a time
E_u, E_l, E_t	The embedding matrix of users, POIs and times
$d(l, l^{'})$	The distance between l and $l^{'}$
η	The dimensionality of feature vector after embedding
δ	The number of considered neighbors for a POI
$N_\delta(l)$	The δ-nearest neighbors of POI l

B. POI Similarity Measurement

The aim of POI-based Collaborative Filtering (PCF) is recommending a user the POIs similar to these he/she checked in before at a specified time. Consequently, the most important problem for PCF is how to measure similarities between POIs properly and recognize which feature is pivotal for a POI to attract users. It is worth mentioning that the word "similarity" in this paper means the probability that two POIs may be accessed by one user at a specified time, but not the similarity on physical category of the two POIs. Based on previous researches and analysis of dataset, we use the following three intuitions as the starting point.

Intuition 1: Users tend to visit nearby POIs to their previous visited POIs, and their willingness to visit a POI decreases as the distance increases. This assumption was proposed and proved in [1]. Therefore, we can conclude that there exists a higher similarity between adjacent POIs.

Intuition 2: The check-in interest of a user is stable in the long term, so we can assume that if two POIs always appear in several users' check-in sets in pair, they are similar to each other. This is the basic assumption of User-based Collaborative Filtering (UCF) [26].

Intuition 3: The check-in habit and interest of a user will vary with time in a day. POIs always visited at one same time are similar to each other. For example, people may visit those recreational POIs during break time, such as a park or a shopping mall, so we can assume that parks and shopping malls are similar choices at break time for users.

To represent the **spatial** feature of POIs (**Intuition 1**), we construct a $|L| \times |L|$ spatial symmetric matrix S. Following the previous study [3], we set $S_{l,l'} = \left(0.5 + d\left(l, l^{'}\right)\right)^{-1}$ if $l^{'} \in N_\delta(l)$, and 0 otherwise. With this equation, we filter those POIs far away to reduce noise and unnecessary computation.

To represent the **co-visit** feature of POIs (**Intuition 2**), a $|L| \times |L|$ co-visit symmetric matrix C is constructed, too. In this matrix, $C_{l,l'} = n$ means n users have previously visited l and $l^{'}$ in pair, while $C_{l,l'} = 0$ means no one has visited l and $l^{'}$ pairwise.

Furthermore, we construct a $|L| \times |T|$ popularity matrix P to follow the change of POIs' **popularity** feature varying with time(**Intuition 3**). A day is divided into $|T|$ time slices and every entry of matrix P represents the POI's absolute popularity at a specified time slice. For example, $P_{l,t} = n$ means the POI l has been visited n times at time slice t.

C. PCF for Feature Embedding

The dimensionality of spatial matrix S and co-visit matrix C will explode as the number of POIs increases. To address this issue, we need to apply appropriate embedding method to construct embedding matrices of users and POIs. The final embedding matrices should reserve the crucial information of above three different feature matrices. Therefore, the similarities between pairwise users and pairwise POIs can be described in their embedding representations. The result comparison of different embedding methods is presented in Section IV.

After the embedding processing on S and C, we get the decomposed feature matrices S_F and C_F respectively. It is worth mentioning that P is directly applied as the popularity feature matrix P_F because the feature dimension is no more than 24. Hence, we use the concatenation of these three feature matrices as the embedding matrix of POIs E_l, and the embedding of POI l can be represented as follow:

$$e_l = [e_l^s, e_l^c, e_l^p]^\top \tag{1}$$

where e_l^s, e_l^c and e_l^p are the spatial feature vector, co-visit feature vector and popularity feature vector of POI l.

As for users' embedding representations, the direct intuition is simple one-hot vectors. However, it is improper in recommendation field, especially in time-aware POI recommendation because of the data sparsity problem. Here is the solution of user embeddings that exploiting the mathematical expectation of visited POI embeddings to represent a given user, just as follows:

$$e_u = \frac{\sum_{l \in L_u} e_l}{|L_u|} \tag{2}$$

where L_u is the POI set that user u has visited.

We don't represent a user in different ways at different times because of two reasons. First, too delicate processing for users will lead to worse results because of the severe data sparsity. Second, our embedding matrix can further be trained in subsequent network. The general rule of initialization is advantageous to the training of recommender network.

With above operations, the embedding matrices with same dimension of users and POIs, E_u and E_l can be well employed in further neural network.

Obviously, human activities are regular and periodic throughout a day. For example, people tend to visit workplace during afternoon and tend to visit restaurant during dinner hours. How to incorporate temporal regularity on check-in activities is the most important problem in time-ware POI recommendation. Fig. 2 shows the similarity curves for 24 hours over Foursquare dataset, where each curve represents a time, and each point on the curve of time t_h corresponds to the similarity of t_h to a certain time. From the diagram, we can see that users' check-in behaviors of a time are different

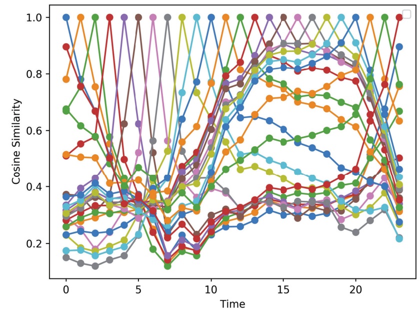


Fig. 2: Similarity Curves of check-in behaviors for 24 hours in a day

with that of other times, but similar to that of its closer time to some extent.

Because of the data sparsity and the consideration for network training, we don't do any initialization to construct the embedding matrix of times. The experimental results also show that no pre-initialized rules on representations of times can get better results.

D. Neural Network for Interaction Learning

Inspired by the network architecture of NCF [21], the RNet illustrated in Fig 3 is designed to learn abstract spatio-temporal check-in interactions.

Input. We use check-in triples like $\langle u_i, l_j, t_h \rangle$, containing the one-hot representations of user, POI and time as RNet's input. In network training, every check-in record that appears in the training set is regarded as a positive sample. For each positive sample, we fix its user and time, and randomly sample a POI unvisited by this user at this time to generate one corresponding negative sample. The negative sample will be fed into the network together with the positive sample.

Network Architecture. As shown in Fig. 3, RNet first adopts an input layer to merge the one-hot representations of user, POI and time, and adopts a trainable embedding layer contains pre-initialized E_u and E_l and randomly initialized matrix E_t. Then we do two operations on these three vectors e_u, e_l and e_t obtained from embedding layer. Multiplying the values of every corresponding position of each two vectors is used to get three new cross feature vectors u_p, u_t and p_t with the same dimension, since the cross features may reflect some certain relationships between embeddings and it was proved to be effective in [27]. Concatenating e_u, e_l and e_t is applied to obtain a new combined feature vector upt, enlightened by NCF [21]. Finally, RNet uses a merging layer to concatenate above four vectors, several full-connected hidden layers to learn abstract features and an output layer with one neuron to show result.

Moreover, in this paper, the Batch Normalization [28] is adopted to normalize data distribution and accelerate the convergence. The Rectified Linear Unit (ReLU) is applied as the activation function of the hidden layers to alleviate the vanishing gradient problem in Deep Neural Network, while the

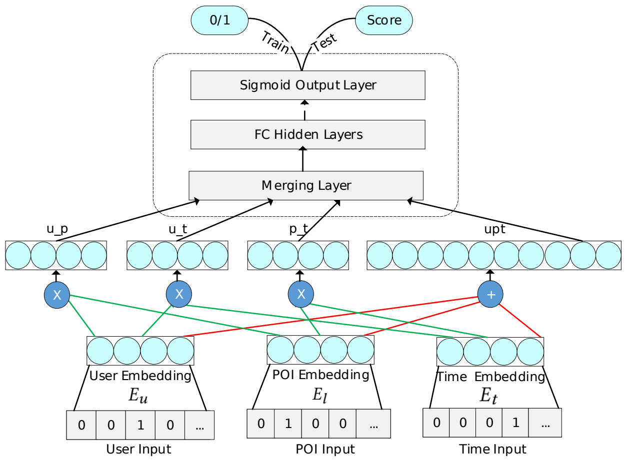


Fig. 3: Network Architecture of RNet. The $\times$ operation is multiplying the values of every corresponding position of two embedding vectors, and getting a new vector with the same dimension. The $+$ operation is getting a new vector by concatenating embedding vectors.

Sigmoid is exploited as the activation function of the output layer to restrict the outputs in range (0, 1).

Output. The output for training step is 1/0 to identify whether the input is a positive sample or a negative sample, which can be seen as a binary classification network. For predicting step, RNet's output is a score in (0, 1) to predict the occurrence possibility of the input triple.

Objective. We adopt the binary cross entropy loss as the loss function shown in Eq.(3), and the objective is to minimize the loss with the effective Adam optimizer [29].

$$loss = -\sum_{x}(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) + \lambda\|W\|_2 \quad (3)$$

where x is the input triple, y is the true label of x, $\hat{y}$ is the output result for x, λ is the parameter of L2 regular term, and W represents all parameters in RNet.

IV. Experiments

We conduct comprehensive experiments to compare ST-RNet with corresponding classical methods on the real-world dataset, and evaluate the performance of ST-RNet with different length of timeslot.

A. Dataset and Metrics

The dataset used in our experiments is the Foursquare public dataset, which was pre-processed in the baseline work [1]. In this dataset, users who have visited fewer than 5 POIs and POIs which have fewer than 5 visitors are filtered out. Finally, it contains 194108 check-ins made by 2321 users at 5596 POIs. For each user, we randomly mark off 62.5% of his/her visited POIs as training data, and the others as testing data to evaluate the effectiveness of recommendation methods.

To evaluate the performance of ST-RNet, we use precision and recall, two general metrics in Top K recommendation scenario. Precision, signed as $Pre@K$, measures how many recommended POIs correspond to the true POIs in testing data for a given user at a given time. Recall, signed as $Rec@K$, measures how many true POIs in testing data have been recommended for a given user at a given time.

For a given user u and a given time t, $R_{u,t}$ is the recommendation POI set, $T_{u,t}$ is the true POI set in testing data and U_t is the user set who have actual check-ins at time t. K is the number of recommended POIs, so the precision and the recall for a time t can be calculated as follows:

$$precision(t) = \frac{\sum_{u\in U_t}|R_{u,t}\cap T_{u,t}|}{K\cdot|U_t|} \quad (4)$$

$$recall(t) = \frac{\sum_{u\in U_t}|R_{u,t}\cap T_{u,t}|}{\sum_{u\in U_t}|T_{u,t}|\cdot|U_t|} \quad (5)$$

Then the overall precision and recall can be calculated as the mathematic expectation of the precision and the recall for all times:

$$Pre@K = \frac{\sum_{t\in T} precision(t)}{|T|} \quad (6)$$

$$Rec@K = \frac{\sum_{t\in T} recall(t)}{|T|} \quad (7)$$

B. Baseline Methods

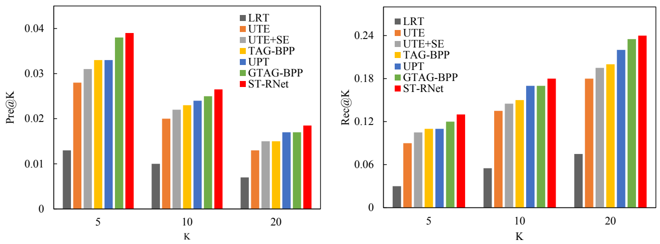
We compare the performance of proposed ST-RNet with following excellent time-aware POI recommendation algorithms:

- **LRT** [11]: a Matrix Factorization(MF) model considering time feature, which based on the phenomenon that user's check-in behaviors vary with time.
- **UTE** [1]: the first work in time-aware POI recommendation field. UTE well captures the periodicity of user's check-in behaviors utilizing UCF with temporal influence.
- **UTE+SE** [1]: the developed UTE algorithm by exploiting both temporal and spatial influence.
- **TAG-BPP** [12]: a graph-based preference propagation algorithm using check-in records of each time to construct a Temporal Aware Graph.
- **GTAG-BPP** [12]: the developed version of TAG-BPP by incorporating geographical influence to construct a Geographical-Temporal Aware Graph.
- **UPT** [2]: an integrated time-aware POI recommendation algorithm analyzing user's interest, location's popularity and temporal features based on UCF.

C. Recommendation performance

Following the previous studies, we select three different K(5, 10, 20) to evaluate the performance of ST-RNet compared with baseline methods in terms of $Pre@K$ and $Rec@K$ roundly.

There are several observations made from the results: (i) The precision value and recall value of **ST-RNet** are always the highest at Top K recommendation on Foursquare. It means that our method based on PCF and Neural Network well learns

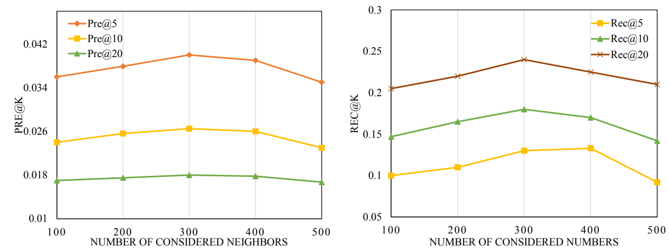


(a) precision comparison (b) recall comparison

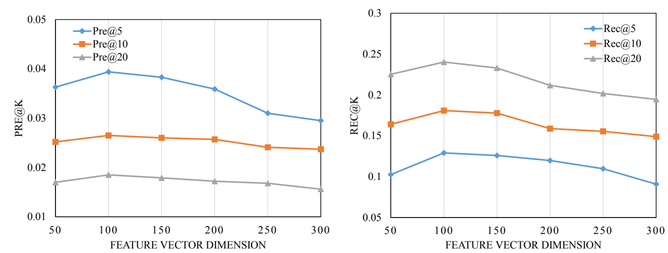
Fig. 4: Recommendation Performance Comparison

TABLE II: Performance comparison on different embedding methods

Performance	SVD	LINE	NMF
Pre@5	0.0256	0.0332	**0.0394**
Pre@10	0.0189	0.0240	**0.0265**
Pre@20	0.0145	0.0168	**0.0185**
Rec@5	0.0756	0.0982	**0.1291**
Rec@10	0.1107	0.1408	**0.1809**
Rec@20	0.1700	0.1970	**0.2403**



(a) The precision of different number of considered neighbors (b) The recall of different number of considered neighbors



(c) The precision of different dimension of feature vector (d) The recall of different dimension of feature vector

Fig. 5: Performance comparison on different hyper parameters

the abstract interactions between users, POIs and time. (ii) **LRT** always performs worst. The main reason is that raw MF without exploiting spatial feature or other features in LBSNs is not applicative for time-aware POI recommendation task due to the severe data sparsity problem. (iii) By comparing the results of **UTE** and **UTE+SE**, we can infer that combining the spatial influence into the recommendation model is effective, because the check-in behaviors of users heavily rely on the spatial feature of POI. Moreover, we can get the same inference from the comparison between **TAG-BPP** and **GTAG-BPP**. (iv) **UPT** performs better than **UTE**, probably because it takes the POI popularity into account. **ST-RNet** further distinguishes POI's different popularity at different times, which is proved to be effective from the results. (v) **UTE** and **UPT** are based on UCF, the mainstream method in time-aware POI recommendation. The results achieved by UTE, UPT and **ST-RNet** show that feature extraction based on PCF can achieve excellent performance, too.

D. Parameter tuning analysis

In this subsection, we conduct some experiments to analyze the impact of parameters, including the embedding method, the number of considered neighbors δ, the dimensionality of feature vectors η and the architecture of RNet.

1) Embedding Methods Tuning: For feature embedding, our purpose is to construct lower dimensional feature matrices from spatial matrix S and co-visit matrix C without losing crucial information. Following are the embedding methods for comparison: (i) **SVD** (Singular Value Decomposition) is an important method of MF in Linear Algebra, where the singular value can imply the potential features in the matrix. The bigger the singular value is, the greater the importance of the corresponding feature. (ii) **NMF** (Nonnegative Matrix Factorization) [30] is a traditional MF method to generate feature matrix without negative values. (iii) **LINE** (Large-scale Information Network Embedding) [31] is an effective network embedding method which suits arbitrary types of information networks. Table II shows that NMF is more effective for our task, possibly because MF is a great feature compression method, and negative values are not significant for the recommender systems.

2) Hyper Parameters Tuning: We tune the **number of neighbors** δ from 100 to 500 step with 100. Fig. 5(a) and Fig. 5(b) show that when δ is 300, the performance will be best. Considering less neighbors may ignore some important POIs and lose the accuracy, while considering too many neighbors will weaken the influence of spatial feature. So in our experiments, we set δ as 300 for the best performance.

Observing Fig. 5(c) and Fig. 5(d), with gradually tuning η from 50 to 300 step with 50, the recommendation performance will reach a peak when the **feature vector dimensionality** η is 100. The principle behind is that lower dimensionality may lose significant information, while higher dimensionality may lead to more parameter training and calculations in the RNet. Hence, in this paper η is set as 100 to achieve a trade-off between recommendation performance and experimental cost.

3) Network Architecture Tuning: The traditional Control Variate Technique is used in the adjustment of hyper-parameters in RNet, as shown in Table III. We firstly try all combinations of the variables presented to acquire the RNet architecture with best performance, which is **Cross features + Combined features, Hidden layers (512-256-128-64-1, Relu activation, Normal initializer, Dropout)**, and then change one variable and keep the others same for evaluation based on the best architecture.

TABLE III: Hyper-parameters adjustment for RNet

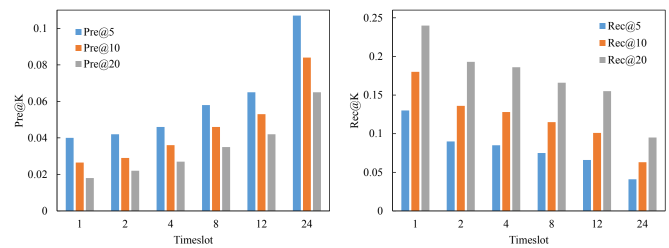
Hyper-parameters of hidden layers		**Pre@5**	**Pre@10**	**Pre@20**	**Rec@5**	**Rec@10**	**Rec@20**
Activation Function	tanh	0.0241	0.0200	0.0154	0.0691	0.1109	0.1813
	relu	0.0394	0.0265	0.0185	0.1291	0.1809	0.2403
Initializer	uniform	0.0343	0.0247	0.0169	0.1033	0.1488	0.2085
	normal	0.0394	0.0265	0.0185	0.1291	0.1809	0.2403
Number of layers and neurons	2048-1024-512-256-1	0.0389	0.0262	0.0174	0.1130	0.1787	0.2184
	1024-512-256-128-1	0.0395	0.0256	0.0182	0.1109	0.1684	0.2266
	512-256-128-64-1	0.0394	0.0265	0.0185	0.1291	0.1809	0.2403
Regular Term	L2reg(0.00001)	0.0388	0.0250	0.0167	0.1042	0.1721	0.2333
	dropout+L2reg	0.0295	0.0215	0.0149	0.0934	0.1275	0.2330
	dropout(0.5)	0.0394	0.0265	0.0185	0.1291	0.1809	0.2403
Network Structure	only cross features	0.0334	0.0246	0.0179	0.0987	0.1444	0.1975
	only combined features	0.0310	0.0234	0.0168	0.0867	0.1373	0.1779
	cross features+combined features	0.0394	0.0265	0.0185	0.1291	0.1809	0.2403

From the results, we can infer three conclusions. (i) The number of neural units and hidden layers are not the crucial factors affecting recommendation performance at the time of convergence, and less hidden layers and neural units can avoid over-fitting. (ii) Activation and initializer play the most important role in the RNet, because they are used to adjust data distribution fed into each layer. Proper data distribution makes each neural unit having chance to be activated and lets neural network achieving its most good further. (iii) Dropout and L2 regular term are the strategies proved to be effective on avoiding over-fitting. However, in our experiments, both of them can improve the performance solely but the combination of them will degrade the performance, maybe because too many regular terms will lead to under-fitting on the contrary. (iv) Using cross features and combined features together will lead to better performance, which shows that both low-order combined features and high-order combined features are vital to the final results.

E. Timeslot length effect analysis

In time-aware POI recommendation task, timeslot is one of the most important parameters to impact the performance. In reality, dividing a day into 24 hours is too detailed and complicated, since human behaviors always last few hours. For example, during the period from 20:00 to 23:00, the user might be watching a movie, so the check-in location can be stable during the whole time period. Larger length of timeslot means less sensitive to time effect, while a smaller one means the opposite. Specifically, the case of timeslot=24 is the normal POI recommendation without considering temporal influence. In this section, we only present the experimental results on ST-RNet as an example, since other methods all achieve the same observations.

With the increase of timeslot length, the recommendation precision and recall are getting better, as shown in Fig. 6. Here are some reasons. (i) People always behave similar over a period of time, so decreasing the timeslot length will ignore this phenomenon and construct a more complicate but non-universal model. (ii) Time-aware POI recommendation suffers more severe data sparsity problem than traditional POI recommendation because of less check-in records and more combinations of user-POI-time. Increasing timeslot length can alleviate the data sparsity problem and achieve better recommendation performance.



(a) The precision of different length of timeslot (b) The recall of different length of timeslot

Fig. 6: Performance of varying length of timeslot

V. Conclusion and Future Work

In this paper, we propose a novel and effective time-aware POI recommendation method named ST-RNet to combine PCF and neural network.

The ST-RNet starts with three hypotheses to measure the similarities between POIs, and then two embedding matrices of users and POIs are constructed, in order to embed multiple features of POIs by PCF. Moreover, above two matrices and a randomly initialized matrix of times constitute the embedding layer of subsequent neural network together. Lastly, user-POI-time check-in records are fed into RNet for training. The RNet gets a score to predict the occurrence possibility of the input check-in triple finally. The experimental results on the Foursquare real-world dataset demonstrate that ST-RNet is well to learn the hidden patterns behind the spatio-temporal interactions, and effective on time-aware POI recommendation task.

For future work, we will conduct more experiments on the temporal periodicity with different granularity, for example, within a month, a season or a year. Moreover, with combining other sequential data of check-ins in LBSNs, we will further explore the temporal trend of check-in behaviors.

ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Program of China under grants 2016YFB0800402 and 2016QY01W0202, National Natural Science Foundation of China under grants U1836204, 61572221, 61433006, U1401258 and 61502185, Major Projects of the National Social Science Foundation under grant 16ZDA092, Guangxi High level innovation Team in Higher Education Institutions-Innovation Team of ASEAN Digital Cloud Big Data Security and Mining Technology and Shenzhen Science and Technology Planning Project under grants JCYJ20170307154749425.

REFERENCES

[1] 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*. ACM, 2013, pp. 363–372.

[2] Y.-L. SI, F.-Z. ZHANG, and W.-Y. LIU, "A time-aware poi recommendation method exploiting user-based collaborative filtering and location popularity," *DEStech Transactions on Computer Science and Engineering (cimns)*, 2017.

[3] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geofm: A ranking based geographical factorization method for point of interest recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 433–442.

[4] 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*. ACM, 2014, pp. 831–840.

[5] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2010, pp. 458–461.

[6] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Aaai*, vol. 12, 2012, pp. 17–23.

[7] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 325–334.

[8] H. Li, Y. Ge, R. Hong, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 975–984.

[9] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2537–2551, 2017.

[10] 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*. ACM, 2015, pp. 443–452.

[11] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 93–100.

[12] 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 Conference on Information and Knowledge Management*. ACM, 2014, pp. 659–668.

[13] L. Cai, J. Xu, J. Liu, and T. Pei, "Integrating spatial and temporal contexts into a factorization model for poi recommendation," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 524–546, 2018.

[14] X. Li, M. Jiang, H. Hong, and L. Liao, "A time-aware personalized point-of-interest recommendation via high-order tensor factorization," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, p. 31, 2017.

[15] R. Ding, Z. Chen, and X. Li, "Spatial-temporal distance metric embedding for time-specific poi recommendation," *IEEE Access*, vol. 6, pp. 67 035–67 045, 2018.

[16] S. Zhang, L. Yao, and A. Sun, "Deep learning based recommender system: A survey and new perspectives," *arXiv preprint arXiv:1707.07435*, 2017.

[17] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the tenth ACM international conference on web search and data mining*. ACM, 2017, pp. 495–503.

[18] Y. Sun, P. Zhao, and H. Zhang, "Ta4rec: Recurrent neural networks with time attention factors for session-based recommendations," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[19] Y. Gui and Z. Xu, "Training recurrent neural network on distributed representation space for session-based recommendation," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[20] D. Wang, G. Xu, and S. Deng, "Music recommendation via heterogeneous information graph embedding," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 596–603.

[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*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[22] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *Proceedings of the 26th international conference on world wide web companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 153–162.

[23] 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*. ACM, 2017, pp. 1245–1254.

[24] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 391–400.

[25] R. Ding and Z. Chen, "Recnet: a deep neural network for personalized poi recommendation in location-based social networks," *International Journal of Geographical Information Science*, pp. 1–18, 2018.

[26] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 501–508.

[27] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, pp. 106–117.

[31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.