This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3002531, IEEE Transactions on Knowledge and Data Engineering

Personalized Long- and Short-term Preference Learning for Next POI Recommendation

Yuxia Wu, Ke Li, Guoshuai Zhao, Xueming Qian, *Member, IEEE*

Abstract—Next POI recommendation has been studied extensively in recent years. The goal is to recommend next POI for users at specific time given users' historical check-in data. Therefore, it is crucial to model both users' general taste and recent sequential behaviors. Moreover, different users show different dependencies on the two parts. However, most existing methods learn the same dependencies for different users. Besides, the locations and categories of POIs contain different information about users' preference. However, current researchers always treat them as the same factors or believe that categories determine where to go. To this end, we propose a novel method named Personalized Long- and Short-term Preference Learning (PLSPL) to learn the specific preference for each user. Specially, we combine the long- and short-term preference via user-based linear combination unit to learn the personalized weights on different parts for different users. Besides, the context information such as the category and check-in time is also essential to capture users' preference. Therefore, in long-term module, we consider the contextual features of POIs in users' history records and leverage attention mechanism to capture users' preference. In the short-term module, to better learn the different influences of locations and categories of POIs, we train two LSTM models for location- and category-based sequence, respectively. Then we evaluate the proposed model on two real-world datasets. The experiment results demonstrate that our method outperforms the state-of-art approaches for next POI recommendation.

Index Terms—Next POI recommendation, Attention mechanism, User preference, Personalization

1 Introduction

RECENT years have witnessed significant development of location-based social networks (LBSNs), such as Foursquare, Gowalla, Facebook Place, and Yelp, etc. Particularly, users can share their locations and experiences with their friends by checking-in points-of-interest (POIs). A check-in record usually contains the visited POI with its associated contexts that describe user movement, including the timestamp, GPS and semantics (e.g., categories, tags, or comments). The massive check-in data generated by millions of users in LBSNs provide an excellent opportunity to explore the intrinsic pattern of user check-in behavior [1-4]. For example, we can recommend POIs for users based on their check-in records, which not only help users to explore their interested places but also benefit for business to attract more potential customers [5, 6, 58].

The check-in sequences implicitly reflect users' preference on POIs and the daily activity patterns of users [7, 8]. Recently, next POI recommendation has received significant attention in research community [9-11, 58]. Excepted for users' general preference (long-term preference), next POI recommendation additionally considers the sequential patterns of users' check-in records (short-term preference).

Our work is motivated by the following inspirations:

(1) Users' long- and short-term preference on POIs co-determine where they will go next time. Therefore, it is necessary to consider the two factors together. In addition, different users show different dependencies on long- and short-term impact. Some users may rely more on long-term preference when making decisions, while others rely more on short-term preference. For example, one user may like outdoor entertainments from long-term preference aspect. But for some reason, he only goes outside several times during the most recent period. Then if he relies more on long-term preference, we will recommend some outdoor places for him. Otherwise, we will recommend him some indoor activities. Thus, it is crucial to learn specific weights on long- and short-term preference for different users to achieve personalized recommendation. However, current researchers always fail to consider users' personalized dependencies on long- and short-term preference.

(2) The check-in behaviors of users are autonomous and elusive, leading it difficult to capture users' long-term regularity. At different time and circumstances, users may prefer different POIs. Therefore, to better learn users' long-term preference for personalized recommendation, it is important to consider the context information of POIs. For example, users will go to restaurants at the time to have diner. Then after having diner over one hour, they will go to a pool for swimming or a park for relaxing.

(3) The activity purpose and check-in locations are inseparable. Excepted for the location-based sequences, the cate-

This work was supported in part by the NSFC under Grant 61732008 and 61772407

- *Yuxia Wu is with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: wuyuxia@stu.xjtu.edu.cn)*
- *Ke Li is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: likely@stu.xjtu.edu.cn)*
- *Guoshuai Zhao is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: guoshuai.zhao@xjtu.edu.cn)*
- *Xueming Qian (corresponding author) is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, the School of Information and Communication Engineering, and SMILES LAB, Xi'an Jiaotong University, Xi'an 710049, China. (e-mail: qianxm@mail.xjtu.edu.cn).*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3002531, IEEE Transactions on Knowledge and Data Engineering

gory-based sequences are also essential to exploit the category information when modeling users' behaviors. Users may prefer different categories at different time. We conduct some statistical analysis of the dataset and take some examples to observe the temporal regularity of the categories.

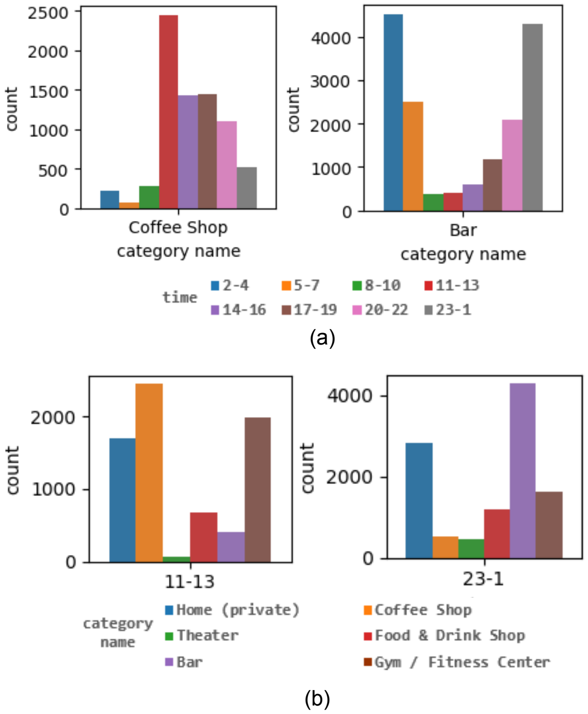

Fig.1. The statistical analysis of the dataset

Fig.1 (a) shows the check-in numbers of different categories at different time. We split the 24 hours into several fragments to show the statistical regularity. We can observe that the distributions of check-in time for different categories are different. For example, the most frequent time for users to check in at the coffee shop is 11: 00.am - 13: 00.pm. They may just finish the lunch and then take a coffee to refresh themselves. The most frequent time for bar is 23: 00.pm - 4: 00.am, which is also in line with people's daily behaviors to relax and drink. Fig.1 (b) shows the distributions of check in categories at different time slots. We can also observe the difference of the distributions of categories at different time. For example, at 11: 00.am - 13: 00.pm, users mostly check-in at coffee shop, home, and Gym/Fitness Center. At 23:00.pm-1:00.am, users usually go to the bar or at home. We can conclude that different categories have different distributions of time, and different time slots have different distributions of categories Thus, the temporal sequence of categories in users' check-in history is also essential to learn users' behaviors.

To this end, we propose a Personalized Long Short-term Preference Learning (PLSPL) model for next POI recommendation. Concretely, we integrate the long- and short-term preference together with user-based linear combination unit to capture users' personalized dependencies on the two parts. In long-term module, we learn contextual features of POIs in their check-in history and utilize attention mechanism to better capture users' long-term preference. In short-term module, we leverage LSTM to model the short-term sequential preference of users. Specially, we learn location-level and category-level preference by training two parallel LSTM models. Finally, we fuse the long-term and short-term together in a personalized way to obtain the final probabilities of candidate POIs.

The main contributions of this paper are summarized as follows:

1) We propose a unified model to learn the long-term and short-term preference of users. Specially, we consider personalized dependencies on long- and short-term preference for different users by user-based linear combination unit.
2) For long-term preference, we extract the contextual features of POIs in users' check-in history and utilize attention mechanism to further characterize the general taste of users.
3) For short-term preference, we integrate the location-level and category-level preference together by two parallel LSTM models to better capture users' sequential behaviors.

Comparing with our preliminary work in [58], we have made some improvements as follows:

1) In order to improve the personality of our recommendation, we introduce a new user-based linear combination unit in this paper. By learning personalized weights of the long-term and short-term modules, our model can better captured the specific preferences for different users. To explain the weights of the long and short term to the final decision, we also provide a user study to analysis the different influences of the long-term and short-term sequences.
2) More detailed steps and explanations of the consideration behind the designs of all the parts in our proposed method are provided. For example, we give more description in the embedding layer and attention mechanism in long-term module; the LSTM model and feature combination in short-term module; the fusion of each module in the output layer.
3) We conduct more comparison experiments and discussions compared with [58]. We compare our method with more baselines to demonstrate the advantage of our method and provide more comprehensive explanations about the results. In additions, more discussions about each part of our model are provided in this paper such as the impact of the factors in each module, the number of users' records and the dimensions of locations and categories.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the related work about next POI recommendation. Section 3 describes our task and some definitions briefly. The overview of the proposed model is introduced in Section 4. Then we give the experimental results and some discussions in Section 5. Finally, we make a conclusion on our study in Section 6.

2 RELATED WORK

In this section, we give a brief review about next POI recommendation. Different from general location recommendation that mainly exploit users' preferences on POIs, next POI recommendation additionally considers the sequential information of users' check-in history. In this part, we firstly introduce the general location-based recommendation methods. Then we introduce the related works on next POI recommendation.

2.1 Location-based Recommendation

Location recommendation has been widely studied in location-based services. Generally, the most well-known approaches of personalized recommendation are Collaborative Filtering (CF) [12, 13], Matrix Factorization(MF) [14, 15]. Collaborative Filtering method firstly mines similar users from users' check-in history. Then recommend POIs according to similar users' check-in records. CF-based method has been demonstrated as an effective approach for recommender system. However, this method suffers the data sparsity problem leading it difficult to identify similar users. Matrix Factorization based methods have become the effective approaches to collaborative filtering. The basic idea of MF methods is to factorize the user-item matrix into two latent matrices which represent the characteristics of users and items.

Compared with other recommendation systems, location recommendation has richer contextual information such as temporal, spatial, textual, visual, social, sentimental information and so on. Zhao et al. [16] proposed a Geo-Temporal sequential embedding rank (Geo-Teaser) model for POI recommendation. In temporal embedding module, they captured the contextual check-in information and the temporal characteristics of POIs. In geographical module, they learn the geographical influence via a hierarchical pairwise preference ranking model. Except for the rich contextual information, the data scarcity problem also brings challenges to POI recommendation. To tackle the data scarcity and various context problem, Yang et al. [17] proposed a semi-supervised learning framework named Preference And Context Embedding (PACE) jointly learning the embeddings of users and POIs. In this model, they built two context graphs: user graph based on friendship and POI graph based on geographical distance among POIs. Then they addressed the data scarcity and various context problem by enforcing smoothness among neighboring users and POIs on the two context graphs. On the other hand, they leveraged neural networks to model nonlinear complex interactions between users and POIs. To tackle the extreme sparsity of user-location matrices when using traditional matrix factorization method, Lian et al. [18] proposed GeoMF ++ model. This model integrated geographical modeling and implicit feedback-based matrix factorization, so that geographical modeling can be incorporated into matrix factorization. Qian et al. [19] proposed a spatiotemporal context-aware and translation-based recommender framework. They leveraged knowledge graph embedding to learn the relationship among users, POIs, and spatiotemporal contexts.

The complex nature of user interest and the sparsity of check-in data bring significant challenges for POI recommendation. It is difficult to capture users' true interest, because the check-in records and the unobserved ones couldn't reflect whether the user really like the location. Therefore, Li et al. [20] proposed a unified model to learn users' general tastes by fusing intrinsic and extrinsic interests. In this way, this model could learn fine-grained and interpretable interest of users. In this model, they first define locations that user can reach as their activity areas. Then they utilized the locations in activity area to learn user's intrinsic interest with pairwise ranking method. Similarly, they utilized the locations outside activity area to capture users' extrinsic interest.

Except for check-in POI recommendation, some researchers also focus on travel recommendation which recommends POIs or travel route for users. Jiang et al. [5] proposed an author topic model-based collaborative filtering (ATCF) method to recommend POIs for users. This model learns users' travel preference topics extracted from the description information of photos. To utilize the visual information in photos for tour recommendation, Zhao et al. [21] proposed a Visual-enhanced Probabilistic Matrix Factorization model (VPMF), which integrates visual features into the collaborative filtering model to learn user interests. Jiang et al. [22] proposed a personalized travel sequence recommendation method to recommend travel route for users. They fuse many contextual information include tags, cost, visiting time and season to mine the topical package space of users and routes. Then they obtained the ranked list of routes according to the similarity between user package and route package. And the ranked routes were further optimized by the similarity among users' travel records.

2.2 Next POI Recommendation

The goal of next POI recommendation is to recommend POIs at next time based on the history records of users. It is crucial to take the sequential information into account. In the literature, effective methods have been widely applied for sequential data analysis and next item recommendation. Generally, the widely used approaches of next POI recommendation are Markov Chains [23, 24] , ranking-based methods [25, 26] and Recurrent Neural Networks (RNNs) based methods [27, 28].

2.2.1 Markov Chains-based Methods

Markov Chains-based methods model the sequential correlation between POIs based on users' check-in sequences. A transition matrix over POIs is estimated which gives the probability of the next POIs based on the recent POIs visited by user. Due to the sparse transition data, it is difficult to estimate the transition probability in Markov Chain. FPMC [29] is a state-of-art method which apply personalized Markov chains and matrix factorization to learn the transition matrix and the general taste of users, respectively. They applied Matrix Factorization method to learn the general taste of a user by factorizing the matrix over observed user-item preferences. Then MC method was used to model the short-term sequential behavior to predict the next action based on the recent actions of a user. However, the complex nature of user interest and the sparsity of check-in data present significant challenges to learn the long-term and short-term preference of users. Following this idea, Cheng et al. [9] combined personalized Markov Chain and localized region constraint, and proposed a novel Matrix Factorization method, namely FPMC-LR. However, in FPMC, each item was represented with two independent vectors, failing to model the latent relationship between them. Therefore, Feng et al. [11] proposed a personalized ranking metric embedding method (PRME) to effectively compute the location transition in Markov chain. This model leveraged metric embedding method which represents each POI as a single point in a latent space to embody the latent relations of POIs.

2.2.2 Ranking-based Methods

In terms of ranking-based methods, Bayesian Personalized Ranking (BPR) [30] is a widely studied method with promising performance. It is a pairwise approach, which takes the implicit feedback as the relative preference rather than absolute one. Zhao et al. [31] established a spatial-temporal latent ranking (STELLAR) model to capture the impact of time information on next POI recommendation. For each POI, three latent vectors were used to describe the POI-user, POI-time, and POI-POI interactions respectively. Then a ranking-based pairwise tensor factorization framework was used to learn these feature vectors and the ranking list of next check in possibilities. However, the aforementioned method still overlooked category-level transition patterns which reflect human daily activities. Therefore, He et.al [32] proposed List wise Bayesian Personalized Ranking (LBPR) method to predict users' next category. The candidate POIs were ranked based on the spatial influence and categorical influence. Besides, Jiao et al. [33] proposed a novel real-time next POI recommendation system. They integrated geographic and preference information to calculate a POI score to obtain the ranking list.

2.2.3 RNNs-based Methods

Recently, recurrent neural networks such as Long Short-term Memory (LSTM) [27, 28] have demonstrated groundbreaking performance on modeling sequential data [34-39]. RNNs have achieved much success in language modeling [40-42], machine translation [43, 44], speech recognition [45, 46], image caption [47, 48], visual question answering [49, 50] and recommendation [10, 51]. However, the original RNNs cannot well model the contextual information such as temporal, spatial information and user activity preference which play a key role in analyzing user behaviors. Therefore, existing studies focus on exploiting users' sequential preference on POIs by integrating various context information into RNNs framework.

For example, Zhu et al. [52] proposed a time-LSTM which equipped LSTM with time gates to model time intervals between users' actions. It was good at modeling the order information in sequential data. Besides, it can also model the interval information between locations. Wang et al. [55] proposed a Similarity-based POI Embedding and recurrent Neural network with Temporal influence (SPENT). They organized the POIs into a similarity tree based on the embedding vectors. Then LSTM added with temporal distance influence was used to learn users' transition behaviors. Excepted for the temporal information, the spatial information is also essential to model users' preference. For example, to model spatial and temporal information, Liu et al. [10] proposed Spatial Temporal Recurrent Neural Networks (ST-RNN) model. ST-RNN utilized RNN to capture the periodical temporal contexts with time-specific transition matrices. Meanwhile, this model incorporated distance-specific transition matrices to characterize dynamic properties of geographical properties of distances. To capture user intentions effectively by fusing various contextual information, Yao et al. [53] proposed a method named Semantics-Enriched Recurrent Model (SERM) which modeled spatio-temporal regularities, activity semantics, and user preferences in a unified way. Considering that users' activity and location preferences interplay with each other, Liao et al. [54] proposed Multi-task Context Aware Recurrent Neural Network (MCARNN) to leverage the spatial-activity topic for activity and location prediction. To integrate the context information and sequential pattern dynamically, the author proposed a novel Context Aware Recurrent Unit (CARU) as hidden layer unit.

Recently, attention mechanism has been widely used in image caption, machine translation and recommendation. Ying et al. [56] proposed Sequential Hierarchical Attention Network(SHAN) which combined long-term and short-term preferences to recommend next item for users. But they failed to consider the sequential behavior of users. Feng et al. [57] proposed an attentional recurrent model named DeepMove to predict human mobility. Firstly, a multi-modal embedding module was designed to convert the sparse features (e.g., user, location, time of day) into dense representations. Then a historical attention module was used to capture the multi-level periodical nature of human mobility by jointly selecting the most related historical trajectories under the current mobility status.

3 Problem Description

Before describing our approach for next POI recommendation, we introduce the notations in this paper. Let $U=\{u_1,u_2,\cdots,u_M\}$ be a set of users, and $L=\{l_1,l_2,\cdots,l_N\}$ be a set of locations, where M and N are the total number of users and locations, respectively. In our work, the categories of locations are also considered. We denote $C=\{c_1,c_2,\cdots,c_K\}$ as the categories of all the locations, where K is the total number of categories. Obviously, different locations can belong to the same categories. Therefore, the number of categories is smaller than the locations. For each user, we define the check-in sequence as follows.

Definition 3.1 (check-in sequence). *The check-in sequence for a user $u\in U$ with n records is a time-ordered sequence $Q^u=\{q_1^u,q_2^u,\cdots,q_n^u\}$. Each record $q_i^u\in Q^u$ contains three attributes (l_i,c_i,t_i), where t_i is the timestamp; $l_i\in L$ is the location visited by user u at time t_i; $c_i\in C$ is the category of l_i.*

Definition 3.2 (long-term sequence). *In this paper, we utilize the data in training set to represent the long-term sequence for a user u, which is regarded as prior information of each user. We set the long-term sequence as $L^u=\{q_1^u,q_2^u,\cdots,q_L^u\}$*

Definition 3.3 (short-term sequence). *Given the raw sequence Q^u of user u, we split it into a set of sequences as short-term sequences. Suppose the length of short-term sequence is k, we set the short-term sequence as $S^u=\{q_1^u,q_2^u,\cdots,q_S^u\}$.*

Formally, given the historical check-in sequence $Q^u=\{q_1^u,q_2^u,\cdots,q_n^u\}$ and the next check-in time t_{n+1} of a user

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3002531, IEEE Transactions on Knowledge and Data Engineering

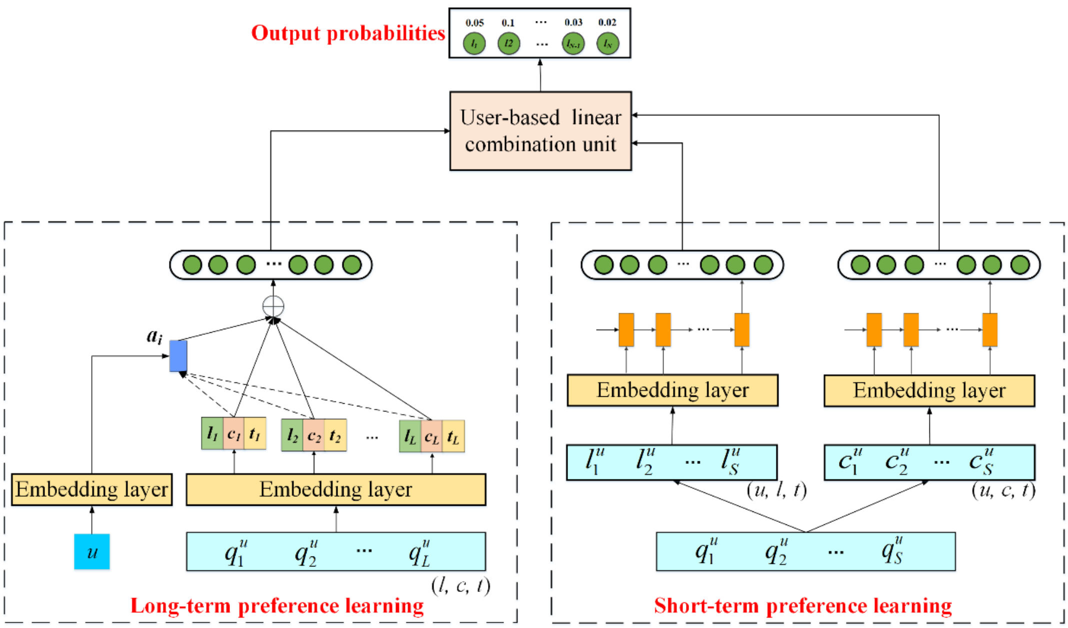

Fig 2. The overall architecture of PLSPL model

u, our goal is to recommend the next location l_{n+1} from the candidate location set L. In order to achieve our task, we learn the user's long-term preference from the long-term sequence $L^u=\{q_1^u,q_2^u,\cdots,q_L^u\}$ and the short-term preference from the short-term sequence $S^u=\{q_1^u,q_2^u,\cdots,q_S^u\}$. Then we fuse them together to capture the preference for next location.

4 Our Model

In this section, we introduce our PLSPL model. Our model characterizes the long-term and short-term preference of users and fuse them into a unified framework. We first present the overall architecture of our model. Then we describe each part in detail in the following sections. At last, we give the objective function and the training algorithm of our approach.

4.1 The Overall Architecture

The illustration of the overall framework is shown in Fig.2. The basic idea of our approach is to recommend a ranked list of POIs for users by jointly learning the long and short-term preferences. More specifically, we learn the long-term preference of user u from the long-term sequence $L^u=\{q_1^u,q_2^u,\cdots,q_L^u\}$. The check-in POIs reflect the general taste of users. And the same POIs may have different impacts for different users. Thus we use the attention mechanism to learn the long-term preference of users similar to [56]. Firstly, we learn the latent vectors for user u and the POI q_i (which contain the location l_i, category c_i and the timestamp t_i) in the embedding layer. Then we compute the importance a_i of each POI q_i in the long-term sequence. Finally, we integrate the embedding of POIs to represent the long-term preference of users. Then the preference vector is fed into a fully connected layer to calculate the probability of next POI.

Meanwhile, we utilize the short-term sequence $S^u=\{q_1^u,q_2^u,\cdots,q_S^u\}$ of user u to capture the short-term inference of users' activity patterns. In the short-term sequence, every factor (l_i,c_i,t_i) of each record is essential to infer users' intensions and preferences. Specially, the locations and categories have different influence on user's preferences at a certain time. Thus, we feed them into two models respectively to learn the location-level and the category-level preferences. Firstly, we learn the latent vectors for user u, locations l_i and categories c_i and timestamps t_i in the embedding layer. To better understand users' check-in behaviors, we separately feed the concatenated embeddings of (u,l_i,t_i) and (u,c_i,t_i) into two LSTM models. Then the fully connected layers are used to calculate the probability of next POI.

Finally, in the output layer, we combine the outputs of the long and short-term together to generate the final probabilities of candidate POIs in the location set L. Specially, to learn the personized preference, we learn the weighted vectors for every user to balance the importance of the long-term, location-level and category-level preferences.

4.2 The Long-term Preference Learning

In this section, we introduce the learning method for long-term preferences of users. The long-term sequence $L^u=\{q_1^u,q_2^u,\cdots,q_L^u\}$ of a user u reflects the general taste of the check-in behavior of user, thus we utilize it to learn the long-term preference. The main idea is to capture the different preferences of each POI in long-term sequence for every user. In this paper, we apply attention mechanism by similarity computation between the latent vectors of user u and POIs to learn the importance of each POI. To learn the latent vector of each POI $q_i^u \in L^u$, different with [56] which only consider the location ID, we also consider the contextual information, such as the location ID l_i, the category c_i and the check-in time t_i. Then the importance of each POI is calculated with attention mechanism. The long-term preference of user u is designed as the weighted summarization of the corresponding concatenated vectors of POIs in long-term sequence. The details of each part are shown as follows.

4.2.1 The Embedding Layer

For the long-term check-in sequence of user u $L^u=\{q_1^u,q_2^u,\cdots,q_L^u\}$, we learn the latent feature of user u and the contextual feature (l_i,c_i,t_i) of every record $q_i^u\in L^u$. For each timestamp t_i, the original information is continuous, which is difficult to embed. Therefore, we map the raw timestamps into discrete hours. Then each hour is represented as a one-hot 24-dimensional vector, where the non-zero entry denotes the index for the hour. Similarly, the user ID u, location l_i and category c_i are also represented as one-hot vectors, where the non-zero entry denotes the indexes. Intuitively, the sparsity increases with the number of the users, locations, categories and time, which will degrade the recommendation efficiency. Therefore, we transform them into D^u, D^l, D^c, D^t dimensional dense vectors, respectively.

To learn the high-level representations of the POIs in long-term sequence of each user, we utilize the nonlinear transformation to capture the latent vector for each POI. Different with [56] which only learn the latent vectors of POI ID, we further consider the context information such as the category of POI and the check-in time. The fused feature of each POI is calculated as follows:

$$h_i=\phi(W_l v_i^l+W_c v_i^c+W_t v_i^t+b). \tag{1}$$

where v_i^l, v_i^c and v_i^t represent the embedding vectors of the tuple (l_i,c_i,t_i) of every POI $q_i^u\in Q^u$ in the long-term sequence L^u. W_l, W_c, W_t and b are the weights and corresponding bias parameters. ϕ is the nonlinear activation function.

4.2.2 The Attention Mechanism

To learn the long-term preferences of users, we leverage the attention mechanism to calculate the summarization of the contextual features of POIs in long-term sequence. We use the embeddings of users learned by embedding layer to measure the similarity between users' preference and the latent vectors of check-in POIs. It is to calculate the importance of each POI for each user. In this way, we can learn users' long-term preference by fusing the latent vectors of POIs with different weights. Here, the importance of each POI is calculated as the normalized similarity between latent vector of the user u and the POI q_i:

$$a_i=\frac{\exp(u^T h_i)}{\sum_i \exp(u^T h_i)}, \tag{2}$$

$$u_{long}=\sum_i a_j[v_i^l;v_i^c;v_i^t]. \tag{3}$$

where $[v_i^l;v_i^c;v_i^t]$ represents the concatenation of the embedding vectors of the tuple (l_i,c_i,t_i) of each POI. a_i denotes the importance of each POI. u_{long} is the final representation of the long-term preference of user u. Then u_{long} is fed into a fully connected layer to calculate the probability of next POI P_L^i.

4.3 The Short-term Preference Learning

We leverage LSTM model to learn the short-term preferences of users. The input sequences contain user ID, location, category and time information. We first learn the latent embedding vectors of them before modeling sequential preference. Considering that the location and category have different influences on the decisions of users, we feed them into two LSTM models without weights sharing. The details of embedding and LSTM layer are introduced in following parts.

Firstly, for the check-in sequence of user u $S^u=\{q_1^u,q_2^u,\cdots,q_S^u\}$, the latent vectors of user u and the tuple (l_i,c_i,t_i) of every record $q_i^u\in S^u$ are represented in the same way as the section 4.2.1.

To better learn the short-term preference of different users, we combine the embeddings of users and time as context information for location-level and category-level sequence. With the context information, the latent vector of the same POI will be different and personalized for different users. Then the combined vectors of locations (u,l_i,t_i) and categories (u,c_i,t_i) are simultaneously fed into two LSTM models to learn the location-level and category-level preferences. By taking the location-level sequence as examples, we model user preference by the basic LSTM as follows:

$$\begin{aligned}
&x_t=[v^u;v^l;v^t]\\
&i_t=\sigma(W_i[h_{t-1},x_t]+b_i),\\
&f_t=\sigma(W_f[h_{t-1},x_t]+b_f),\\
&c_t=\tanh(W_c[h_{t-1},x_t]+b_c),\\
&c_t=f_t\odot c_{t-1}+i_t\odot c_t,\\
&o_t=\sigma(W_o[h_{t-1},x_t]+b_o),\\
&h_t=o_t\odot\tanh(c_t).
\end{aligned} \tag{4}$$

where x_t represents the input vector. $[v^u;v^l;v^t]$ denotes the concatenation of the embeddings of users, locations and time. i_t, f_t, o_t represent the input, forget and the output gate of step t, which deciding what information we're going to store, forget, and output, respectively. c_t denotes the new candidate state vector of step t. $\odot$ is element-wise product of two vectors. $f_t\odot c_{t-1}$ represents the retaining information obtained from the old state after forgetting the information of the old state c_{t-1} that we decide to forget. $i_t\odot c_t$ represents the adding new information obtained from the new state c_t that we decide to store.

c_t is the final state vector that combining the information of the old state c_{t-1} and new state c_t. h_t is the hidden output vector that represents the preferences of users. σ is a sigmoid layer which outputs a number between 0 and 1. W_i, W_f, W_o and W_c are the weights of gates. b_i, b_f, b_o and b_c are corresponding biases.

Then the output vectors of the two LSTM models are fed into a fully connected layer to calculate the probability of next POI P_l^i and P_c^i.

4.4 User-based Linear Combination Unit

In real life, when deciding where to go, different users show different dependencies on long- and short-term pref-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3002531, IEEE Transactions on Knowledge and Data Engineering

erences. However, many researchers in the literatures always neglect the important factor. To learn the different dependencies for different users, we integrate the results of long- and short-term preference learning modules with user-based linear combination unit in the output layer. Specifically, to learn the personized preferences for different users, we learn the personized weights over long- and short-term modules for different users. The user preferences here are different from the preferences in the long-term and short-term modules. They represent the personalized weights on the long- and short-term preferences. We compute the probabilities of next POI by linear combination of P_L^i, P_l^i and P_c^i as follows:

$$P_i = \alpha^u \cdot P_L^i + \beta^u \cdot P_l^i + \gamma^u \cdot P_c^i, \tag{5}$$

where P_L^i represents the output probability for next POI obtained from the long-term preference learning. P_l^i and P_c^i are the outputs of the location-based LSTM and category-based LSTM, respectively. $\alpha^u, \beta^u, \gamma^u$ are the specific weights for user u which will be learned by our model. The final output probability of POI i is determined as follows:

$$O_i = \frac{e^{p_i}}{\sum_{j=1}^{N} e^{p_j}}. \tag{6}$$

where N is the total number of candidate POIs. e is the exponential function.

4.5 Model optimization

So far, we have introduced our solutions to capture users' preferences in different level. Given a training set with $\Re$ samples, the loss function of the proposed model is defined as follows:

$$J = -\frac{1}{\Re}\sum_{i=1}^{\Re}\sum_{j=1}^{N} y_{ij} \cdot \log(O_{ij}) + \lambda \|\Theta\|_2. \tag{7}$$

where J is the cross-entropy loss between the recommendations of our model and the ground truth. $\Re$ and N represent the numbers of the training set and the candidate POIs, respectively. y_{ij} is an indicator variable representing whether the item is the ground truth . Its value is 1 when the POI j is the ground truth, otherwise it is 0. O_{ij} is the output probability for POI j computed by our model. $\|\Theta\|_2$ is the regularization term to avoid overfitting. λ controls the importance of regularization term. To minimize the object function, we use Stochastic Gradient Descent (SGD) and the Back Propagation Through Time (BPTT) algorithm to learn the parameters. The detailed learning algorithm is presented in Algorithm 1. The inputs of our model are the long-term sequence $L^u = \{q_1^u, q_2^u, \cdots, q_L^u\}$ and the short-term sequence of users $S^u = \{q_1^u, q_2^u, \cdots, q_S^u\}$. Firstly, we compute the latent vectors of user, location, category and timestamp via the embedding layer. Then the long-term preference is obtained according to equations (1) ~ (3) based on the long-term sequence L^u. Thus, the probability of next POI P_L^i is computed. After that, the short-term preference is calculated according to equation (4) based on the short-term sequence S_i^u. Then the probabilities of next POI P_l^i and P_c^i are computed by the location-level and category-level preference learning modules. At last, the final probability of POI i is obtained by fusing the outputs of long- and short-term modules according to (5) ~ (6).

Algorithm 1

Input: The long-term sequence $L^u = \{q_1^u, q_2^u, \cdots, q_L^u\}$ and the short-term sequence $S^u = \{q_1^u, q_2^u, \cdots, q_S^u\}$ of users
Output: Trained Model.
Shuffle all the sequences
Initialize the parameters Θ
Repeat
  for each input sequence **do**
    Compute long-term preference u_{long} according to equations (1) ~ (3)
    Compute the probability of next POI P_L^i
    Compute the short-term preference according to equation (4)
    Compute the probability of next POI P_l^i and P_c^i
    Compute the final output O_i according to (5)-(6)
    Update Θ with gradient descent according to (7)
Until convergence
Output trained model

5 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of our proposed model PLSPL on two real-world datasets from Foursquare check-in data. We briefly introduce the datasets firstly and then we evaluate the proposed model with the related methods mentioned before. Furthermore, we give some discussions about our proposed model. At last, we show the user study to explain the fusion weights of long- and short-term modules.

5.1 Datasets

We evaluate our model on public Foursquare check-in datasets collected from New York City (NYC) and Tokyo (TKY) [1], which have been widely used in many related research papers. The check-in records are collected from April 2012 to February 2013. Each record contains user ID, POI ID, category name, GPS and timestamp. In following experiments, for each user, we set the records in chronological order based on the timestamp of each record. We split the records into several sessions keeping each session as the same length. Then we take the first 80% check-ins as the training set, the latter 20% as the test set. After data preprocessing, the overall statistics is shown in Table 1.

TABLE 1

DATASETS STATISTICS

	#user	#location	#category	#session
NYC	1,083	38,333	398	11,415
TKY	2,293	61,858	385	28,727

5.2 Baselines

Several baselines and state-of-the-art methods on next POI recommendation are used for comparison.

MF [14] modeled the latent vectors of users and items by Matrix Factorization.

FPMC [29] modeled both general taste and sequential behavior by integrating Matrix Factorization and Markov Chain method.

ST-RNN[10] modeled temporal and spatial contexts in recurrent neural network with time-specific and distance-

TABLE 2

PERFORMANCE COMPARISON WITH BASELINES

Datasets	Methods	P@1	P@5	P@10	P@20	MAP@5	MAP@10	MAP@20
NYC	MF	0.0332	0.0859	0.1348	0.2013	0.0518	0.0571	0.0599
	FPMC	0.0892	0.2262	0.2943	0.3895	0.1363	0.1451	0.1483
	ST-RNN	0.1103	0.2171	0.2580	0.2882	0.1471	0.1614	0.1636
	LSTM	0.1147	0.2424	0.2916	0.3249	0.1629	0.1695	0.1718
	SHAN	0.1353	0.1779	0.1896	0.2019	0.1510	0.1526	0.1545
	DeepMove	0.1408	0.2946	0.3630	0.4052	0.1975	0.2071	0.2101
	MCARNN	0.1477	0.2909	0.3510	0.3894	0.2005	0.2088	0.2115
	PLSPL (ours)	**0.1559**	**0.3252**	**0.3953**	**0.4475**	**0.2172**	**0.2266**	**0.2302**
TKY	MF	0.0174	0.0550	0.0837	0.1439	0.0302	0.0335	0.0362
	FPMC	0.0655	0.1725	0.2385	0.2944	0.1057	0.1131	0.1128
	ST-RNN	0.1204	0.2437	0.2927	0.3421	0.1667	0.1733	0.1767
	LSTM	0.1339	0.2737	0.3295	0.3780	0.1868	0.1942	0.1975
	SHAN	0.1084	0.1527	0.1684	0.1813	0.1266	0.1287	0.1296
	DeepMove	0.1282	0.2488	0.2923	0.3289	0.1735	0.1794	0.1820
	MCARNN	0.1490	0.3128	0.3723	0.4292	0.2093	0.2174	0.2214
	PLSPL (ours)	**0.1571**	**0.3321**	**0.4020**	**0.4664**	**0.2212**	**0.2307**	**0.2352**

specific transition matrices.

LSTM [27] applied recurrent neural network to learn users' sequential behaviors based on the check-in location sequences.

SHAN [56] applied a nonlinear two-layer hierarchical attention network to capture users' dynamic preferences including long-term preference and short-term preference.

DeepMove [57] learned user preference using recurrent neural networks for historical sequence and current sequence. Specially, an attention mechanism is used to compute the similarity of current state and historical states.

MCARNN [54] learned the spatial-activity topics as the latent factor to capture both users' activity and location preferences. Besides, they proposed a novel Context Aware Recurrent Unit (CARU) to integrate the sequential dependency and temporal regularity of spatial activity topics.

5.3 Evaluation Metrics

In this paper, we use precision@k (P@k) and MAP@k to evaluate the performance of different methods. They are standard metrics for evaluating the quality of the ranked lists. The larger the value, the better the performance. For each user, P@k indicates that whether the ground truth POI appears in the top-k recommended POIs and MAP@k measures the order of our recommendation list. We use the two metrics because we want the recommended item to appear not only in the top K lists, but also at the top of the recommended list. We set k = 1, 5, 10, 20 in our experiments. Given a training set with $\Re$ samples, the functions of the two metrics are defined as follows:

$$\mathrm{P}@k = \frac{1}{\Re}\sum_{i\in\Re}\frac{\left|S_{rec}^{i}\cap S_{visited}^{i}\right|}{\left|S_{visited}^{i}\right|}, \tag{8}$$

$$\mathrm{MAP}@k = \frac{1}{\Re}\sum_{i\in\Re}\frac{\left|S_{rec}^{i}\cap S_{visited}^{i}\right|}{position}. \tag{9}$$

where S_{rec}^{i} indicates the top-k recommended POIs. $S_{visited}^{i}$ indicates the ground truth POI visited by users. In our next POI recommendation task, the number of $S_{visited}^{i}$ is 1. The position in MAP@k represents the position of the correctly recommended POI in the ranked list.

5.4 Parameter Setting

The key parameters in our model include: the embedding dimensions of latent vectors for users D^u, locations D^l, categories D^c and time D^t, the dimension of the hidden state and the batch size. Considering the vocabulary size of them on both datasets, we set the dimensions of locations and categories on NYC to be D^l =300, D^c =100 and D^l =350, D^c =120 respectively. We set the dimensions of users and time to be D^u = 50, and D^t =20 respectively. The batch size was set to be 32, and the learning rate is set to be 0.001. The length of the short-term sequence is 20.

5.5 Performance Comparison

In this sub-section, we compare the performance of our model with other methods. The performance of all methods evaluated by precision@k and MAP@k in NYC and TKY datasets is illustrated in Table 2. We can observe that:

(1) Our model PLSPL outperforms the compared methods under all the metrics on the two datasets. Concretely, for P@k on the NYC dataset, our method is almost 12%-26% higher than MF, 5%-10% higher than FPMC, 1%-24% higher than SHAN, 4%-12% higher than LSTM, 4%-15% higher than ST-RNN, 1.5%-4.2% higher than DeepMove and 0.8%-5.8% higher than MCARNN. For MAP@20, our method outperforms MF, FPMC, SHAN, LSTM, ST-RNN, DeepMove and MCARNN by 17.03%, 8.19%, 7.57%, 5.84%, 6.66%, 2.01% and 1.87% respectively. On the TKY dataset, our method is also higher than other methods under all metrics. This indicates that our model can better capture users' long- and short-term preferences. Meanwhile, it also demonstrates the effectiveness of considering the contextual information and personalized dependencies on different parts for different users.

(2) SHAN shows an increase compared with MF and FPMC under P@1 and all MAP@k on the NYC dataset. That's because FPMC combines the matrix factorization and Markov chain in linear way. However, SHAN utilizes nonlinear model to better learn the user-item interaction.
(3) LSTM shows better performance than FPMC, ST-RNN and SHAN on many metrics on both datasets. That's because LSTM can better model sequential data than Markov Chain and RNN. Meanwhile, it demonstrates that the sequential information plays a very important role in next POI recommendation task.
(4) DeepMove shows better performance on all the metrics than FPMC, ST-RNN, SHAN and LSTM on NYC dataset. That's because DeepMove applies LSTM to model both long- and short-term preferences. Moreover, it leverages attention mechanism to learn the attention weights between recent states and history states. However, it shows slightly poor performance than LSTM on TKY dataset. This phenomenon can be explained that the average number of check-in records in TKY dataset is larger than NYC. This will lead to a particularly long historical data, which is difficult for DeepMove to better capture history information.
(5) The performance of MCARNN is better than the other baselines on the two datasets. That's because MCARNN integrates the temporal and sequential contexts dynamically. This model learns the weights of the sequential and the temporal context to capture the effect of the timespan between the target check-in time and the last time. Besides, they learn users' activity and location preferences by multi-task learning. In this way, they can better capture the latent factor by the shared CARU layer. However, the MCARNN shows worse performance compared with our method. That's because the MCARNN ignores the long-term preference of users. And the category information is regarded as a side information. The latent vectors of the category, the location and user are combined linearly to learn the sequential pattern. Compared with MCARNN, our method considers the long-term and short-term preferences of users. For short-term module, we learn the location-level and category-level preferences by two parallel LSTM models to capture sequential behaviors. Specially, we also learn different weights for the long- and short-term modules for different users to better learn users' personized preferences.

5.6 Discussions

In addition to the performance comparison of the proposed model with the existing MF, FPMC, ST-RNN, LSTM, SHAN, DeepMove and MCARNN, we also discuss some variant models to demonstrate the importance of each part of our model. We discuss six aspects in our experiments: (1) the impact of integration of long- and short-term modules; (2) the impact of factors in long-term module; (3) the impact of factors in short-term module; (4) the impact of the number of users' historical records; (5) the impact the number of users' current records; (6) the impact of the dimensions of locations and categories.

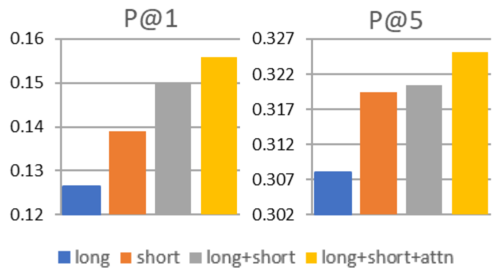

Fig.3. Discussions on the impact of integration of long- and short-term modules

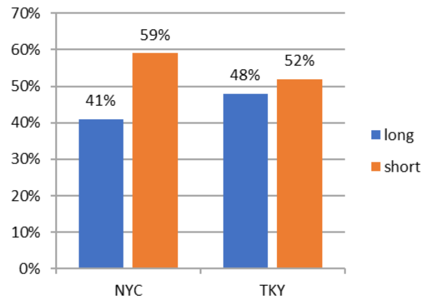

Fig.4. The proportion of different parts on two datasets

5.6.1 The Impact of Integration of Long- and Short-term Modules

To demonstrate the impact of integration of long- and short-term modules in our model, we perform experiments with variant models as follows:

1) **long:** variant model with only the long-term preference learning module.
2) **short:** variant model with only the short-term preference learning module.
3) **long+short:** variant model with long- and short-term preference learning module. Here we learn the same weights on the two parts for all users.
4) **long+short+attn (PLSPL):** our PLSPL model considering both long- and short-term modules with user-based linear combination unit.

Due to space limitation, we just investigate the performance under P@1 and P@5 on NYC dataset. We show the performance of different factors under P@1 and P@5 on the two datasets in Fig.3. We can observe that compared with long-term preference learning, the short-term behavior shows better performance. We suppose that the long-term preference reflects the inherent characters of users which are difficult to represent essentially. While the short-term preference can be learned by sequential information of recent behaviors. Besides, the integration of long- and short-term preference learning modules shows better performance than any single part under all metrics. It indicates that integrating users' general taste and recent interest is crucial to better learn and understand user's check-in behavior.

Specially, our PLSPL model **(long+short+attn)** outperforms all the variant models. It demonstrates the effectiveness of user-based linear combination unit learning different dependencies of long- and short-term modules. To intuitively interpret the user-based linear combination unit, we compare the proportion of the weights in long- and short-term preference for all the users in NYC and TKY datasets. As shown in Fig.4, in NYC datasets, 41% users depend more on long-term preference, and 59% users depend more on short-term preference. In TKY datasets, 48% users depend more on long-term preference, and 52% users depend more on short-term preference. We also give the user study in the following Section 5.7.

5.6.2 The Impact of Factors in Long-term Module

To better capture the long-term preference of users, we consider the category and check-in time of each location in long-term sequence. In this sub-section, we discuss the impacts of the two contextual factors in long-term module. We perform experiments for variant models with only long-term module as follows:

1) **loc:** variant model only considering the latent vectors of locations.
2) **loc+cate:** variant model further considering the categories of locations.
3) **loc+cate+time:** the variant model considering the locations, categories and the check-in time.

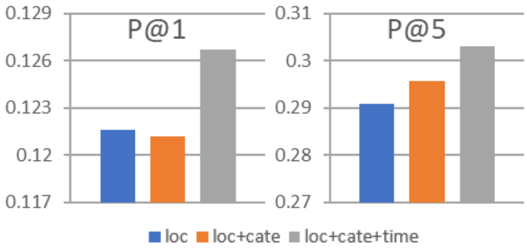

Fig.5. Discussions on the impact of contextual factors

The performance of different variant models is shown in Fig.5. We can see that the variant models with contextual factors perform better than the model with only the location information. Moreover, **loc+cate+time** model shows the best performance and has a more significant improvement than **loc+cate.** It indicates that contextual information is helpful to understand and capture users' long-term preferences.

5.6.3 The Impact of the Factors in Short-term Module

In short-term module, we apply two LSTM models to learn users' location- and category-level preferences. For each model, we also consider the latent vectors of users and check-in time to better learn users' sequential preferences. Besides, we leverage user-based linear combination unit to learn different dependencies on the two sub-modules. Here, we discuss the impact of different factors for variant models with only short-term module as follows:

1) **loc:** denotes the variant model considering the location-level preference and the latent vectors of locations.
2) **cate:** denotes the variant model considering the category-level preference and the latent vectors of categories.
3) **context:** denotes the variant model considering the latent vectors of users and check-in time.
4) **attn:** denotes the variant model considering the user-based linear combination unit.

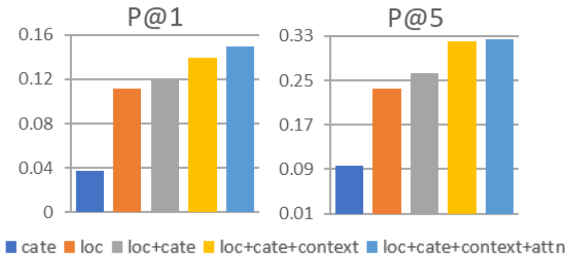

Fig.6. Discussions on the impact of factors in short-term module.

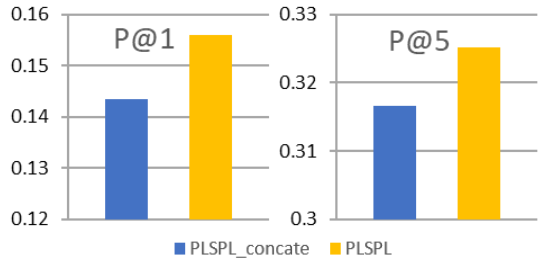

Fig.7. Discussions on the impact of location-level and category-level preference learning

5) **concate:** denotes the variant model with the concatenation of the location, category and the context information as the input of LSTM.
6) "+" denotes we take another factor into consideration. For example, **loc + cate** denotes the variant model considering the location-level and category-level preferences.

The performance of different factors is shown in Fig.6. We can see that the integration of location- and category-level preference shows better performance than any single one. Moreover, the model considering contextual information plays an important role in improving the performance. It also demonstrates the effectiveness of the contextual information. Besides, the model with user-based linear combination unit shows the best performance. It indicates that considering different dependencies on location- and category-level preferences is helpful for models with only short-term module.

In addition, to further demonstrate the effectiveness of location-level and category-level preference learning, we compare our PLSPL method with another variant model which sets the concatenation features of location and category as inputs in short-term module. From Fig.7 we can see that the performance of concatenation is worse than our proposed method. Once again, we demonstrate that the location-based sequence and the category-based sequence bring different information for understanding users' short-term preferences.

5.6.4 The Impact of the Number of Users' Records

In order to show the impact of the number of users' records, we divide the test dataset into four groups according to the number of check-in records on NYC dataset as shown in Table 3. Because the minimum number of users' records is 100, we divide the users starting from 100. The "100-150" means the number of records between 100 and 150. '500+' means the number of records more than 500. We can observe that the check-in number of most of the users is

TABLE 3

THE NUMBER OF USER'S RECORDS ON NYC DATESET

Numbers	100-150	151-200	201-300	300+
User count	518	243	154	157

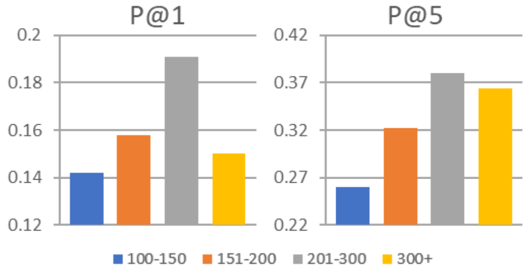

Fig.8. Discussions on the impact of users' records

100~300. The performance of P@1 and P@5 for each group is shown in Fig.8. We can see that when the number of check-in records is less than 300, the performance will be better along with the increase of the check-in number. However, when the number is larger than 300, the performance is decreased. This phenomenon can be interpreted that when the number of users' records is very large, the long-term sequences of users will also be longer. Thus, the general tastes of users will be more complicated and elusive, leading it difficult to better capture users' long-term preferences.

5.6.5 The Impact of the Number of Users' Short-term Records

In our former experiments, we split users' records into many sub-sequences with 20 records as short-term sequence. To show the impact of the number of users' short-term records, we investigate the performance of test datasets under different lengths. Here we set the lengths of short-term sequences to be 1~19 with interval to be 1. To further demonstrate the effectiveness of our method, we compare the performance of our model with DeepMove. The results under P@1 and P@5 on NYC dataset are shown in Fig.9. We can observe that when the number of short-

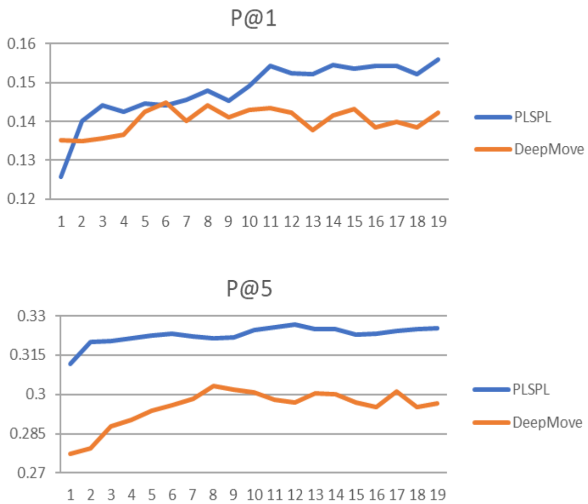

Fig.9. Discussion on the impact of the number of users' short-term records.

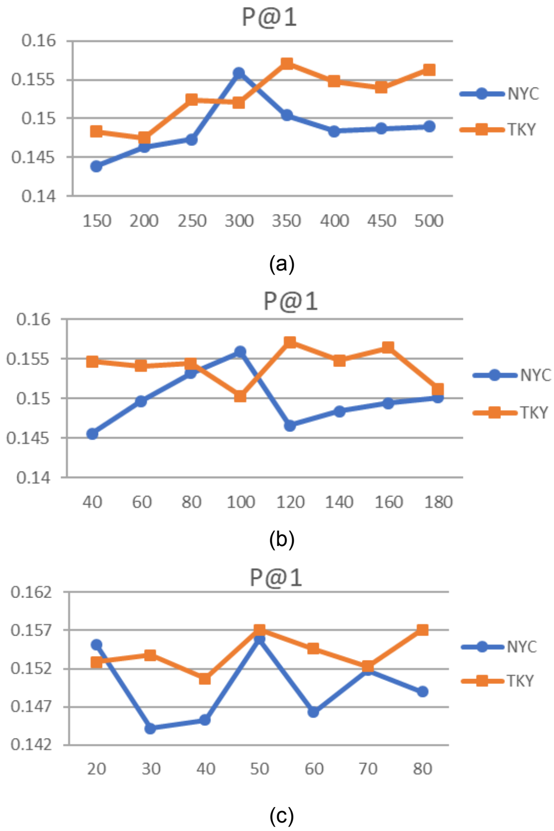

Fig.10. The P@1 under different dimensions of (a) locations (b) categories and (c) users

term records is small, our model can still achieve considerable results. Besides, as the number increases, the performance of our model under P@1 is significantly getting better. However, the improvement of DeepMove is not so obvious. For P@5, our method outperforms DeepMove on all numbers of users' short-term records.

5.6.6 The Impact of the Dimensions of Locations and Categories.

The dimensions of latent vectors for POIs, categories and users are the most important parameters in our model. Due to space limitation, we just investigate the performance with respect to P@1 on NYC and TKY datasets. For each parameter, we perform experiments with the others fixed on both datasets. Considering the total number of POIs, categories and users, we set the embedding dimensions of POIs and categories to be 150~500, 40~200 and 20~80, respectively.

The results are shown in Fig.10. We can see that when the embedding size is too large or too small, the performance is not so good. The model performs best when the embedding size of POIs is 300 and 350 for the two datasets respectively. That's because the total number of POIs are not the same for the two datasets. And we can observe that high dimensions perform better that low dimensions, that's because high dimensions can better capture the characters of POIs. For the embedding size of categories, the best one is 100 and 120 for the two datasets. For the embedding size of users, the best one is 50 for the two datasets.

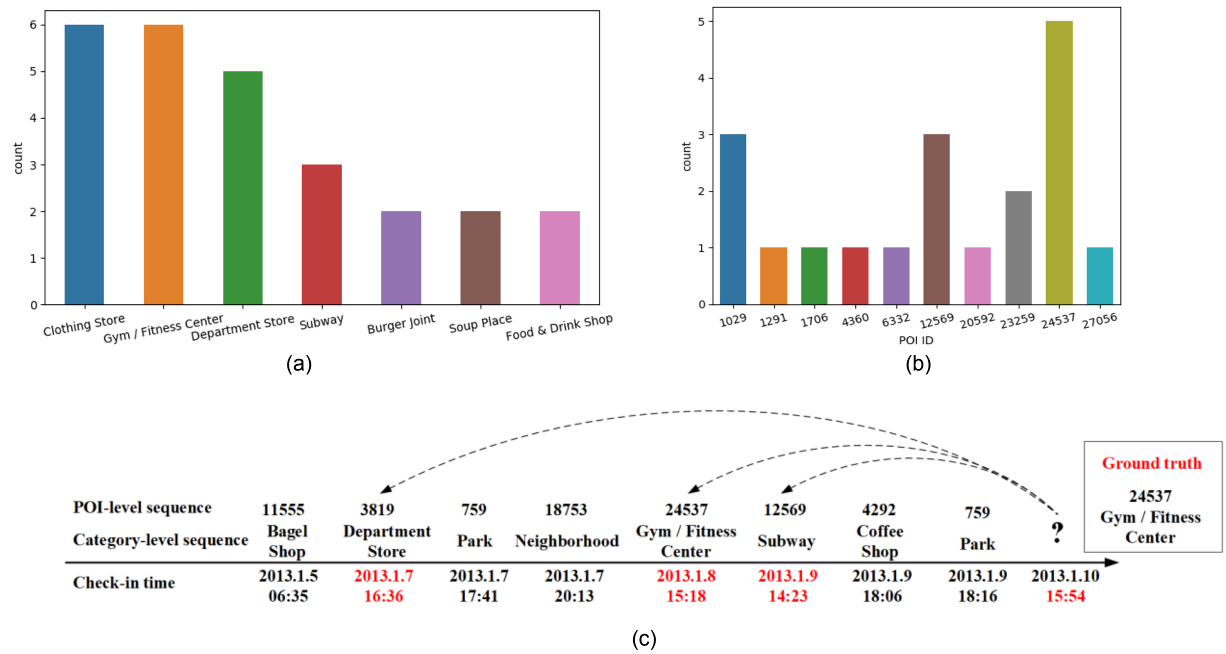

Fig.11 An example of the intuitive explanation of the weights for long- and short-term module. (a)The frequency of category name in long-term sequence; (b)The frequency of POI ID in long-term sequence. (c)The short-term sequence. The black dashed arrows represent the possible neighbor items of the target time. And we use red fonts to highlight the time information.

5.7 User Study

To explain the weights of the long and short term preference to the final decision, we randomly select one user from our dataset. The user ID of the selected user is "2". We recommend the next POI at 15:54 on Jan10, 2013. Our model integrates the long- and short-term preferences of the user. And the weights computed by our model are 0.63 and 0.37 for long- and short-term module. It indicates that the user shows more dependence on the long-term check-in histories. To intuitively interpret the meaning of the output weights, we select one sequence from his/her testing data as short-term sequence. As shown in Fig.11(c), the target time of next POI is 15: 54.pm. Considering the time shift, we find the possible POIs From the short-term sequence at 14:00~16:00. From the black dashed arrows, we can observe that the user may check-in at 3819 (Department Store), 24537 (Gym/Fitness Center), 12569 (Subway). However, from the analysis of the short-term sequence, we are not sure which one he/she will visit at next time. From the frequency analysis of the long-term items in Fig.11 (a) and (b), we can observe that the user mainly visits the Clothing Store, Gym/Fitness Center and Department Store at 14:00~16:00. And the most frequent POI is 24537, which is belonging to the Gym/Fitness Center. Besides, the weights for long- and short-term modules computed by our model indicate that the user shows more dependence on the long-term module. Therefore, our model recommend the 24537 (Gym/Fitness Center) for the next POI, which is in line with the ground truth.

6 Conclusion And Future Work

In this paper, we propose a unified model jointly learning users' long- and short-term preferences for next POI recommendation problem. And we specially learn personalized weights over different parts. In long-term module, we characterize contextual features of POIs and capture the long-term preference via attention mechanism. In short-term module, we learn the location-level preference and category-level preference by two parallel LSTM models. From the experiments, we observe that our model outperforms the state-of-the-art methods on real-world datasets in terms of precision and MAP. Besides, we demonstrate the importance of each part of our model according to the variant models. In future work, we will incorporate more context information such as the social network and spatial information into the model to further improve the next POI recommendation performance.

References

[1] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2014.

[2] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proc. 6th AAAI*, 2012.

[3] C. Song, T. Koren, P. Wang, and A.-L. Barabasi, "Modelling the scaling properties of human mobility," *Nature Physics*, vol. 6, no. 10, p. 818, 2010.

[4] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proc. 12th ICDM*, 2012, pp. 1038–1043.

[5] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized POI recommendations," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 907–918, 2015.

[6] P. Lou, G. Zhao, X. Qian, H. Wang, and X. Hou, "Schedule a rich sentimental travel via sentimental POI mining and recommendation," in *Proc. BigMM*, 2016, pp. 33–40.

[7] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Pro. 17th ACM SIGKDD*, 2011, pp. 1082–1090.

[8] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in foursquare," in *Proc. 5th AAAI*, 2011.

[9] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. 23rd IJCAI*, 2013.

[10] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 20th AAAI*, 2016.

[11] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *24th International Joint Conference on Artificial Intelligence*, 2015.

[12] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[13] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th ACM SIGIR*, 2011, pp.325–334.

[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[15] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. 26th AAAI,* 2012.

[16] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *Pro. 26th WWW,* 2017, pp. 153–162.

[17] 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 *Pro. ACM SIGKDD*, 2017, pp. 1245–1254.

[18] D. Lian, K. Zheng, Y. Ge, L. Cao, E. Chen, and X. Xie, "Geomf++: Scalable location recommendation via joint geographical modeling and matrix factorization," *ACM Transactions on Information Systems (TOIS)*, vol. 36, no. 3, p. 33, 2018.

[19] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based POI recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, p. 18, 2019.

[20] H. Li, Y. Ge, D. Lian, and H. Liu, "Learning user's intrinsic and extrinsic interests for point-of-interest recommendation: A unified approach." in *Proc. IJCAI*, 2017, pp. 2117–2123.

[21] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 43–56, 2016.

[22] P. Zhao, X. Xu, Y. Liu, V. S. Sheng, K. Zheng, and H. Xiong, "Photo2Trip: Exploiting visual contents in geo-tagged photos for personalized tour recommendation," in *Pro. 25th ACM MM*, 2017, pp. 916– 924.

[23] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden markov models," in *Pro. ACM UbiComp*, 2012, pp. 911–918.

[24] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "Gmove: Group-level mobility modeling using geo-tagged social media," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 1305–1314.

[25] X. Li, G. Cong, X. Li, T. N. Pham, and S. Krishnaswamy, "Rank-geofm: A ranking based geographical factorization method for point of interest recommendation," in *Proc. 38th ACM SIGIR*, 2015, pp. 433–442.

[26] H. Ying, L. Chen, Y. Xiong, and J. Wu, "PGRank: Personalized geographical ranking for point-of-interest recommendation," in *Proc. 25th WWW*, 2016, pp. 137–138.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[29] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proc. 19th WWW*, 2010, pp. 811–820.

[30] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th UAI*, 2009, pp. 452–461.

[31] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation," in *Proc. 30th AAAI*, 2016.

[32] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise Bayesian personalized ranking." in *Proc. IJCAI*, 2017, pp. 1837–1843.

[33] X. Jiao, Y. Xiao, W. Zheng, H. Wang, and Y. Jin, "R2SIGTP: a novel real-time recommendation system with integration of geography and temporal preference for next point-of-interest," in *Proc. WWW*, 2019, pp. 3560–3563.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. CVPR*, 2016, pp. 21–29.

[37] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question image co-attention for visual question answering," in *Advances in Neural Information Processing Systems*, 2016, pp. 289–297.

[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[39] X. Chen and C. Lawrence Zitnick, "Mind's eye: A recurrent visual representation for image caption generation," in *Proc. CVPR*, 2015, pp. 2422–2431.

[40] M. Sundermeyer, H. Ney, and R. Schluter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[41] S. Li, C. Wu, H. Li, B. Li, Y. Wang, and Q. Qiu, "FPGA acceleration of recurrent neural network-based language model," in *Proc. 23rd FCCM*, 2015, pp. 111–118.

[42] X. Chen, X. Liu, Y. Qian, M. J. Gales, and P. C. Woodland, "CUED-RNNLM: An open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *Proc. ICASSP*, 2016, pp. 6000–6004.

[43] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[44] F. Meng, Z. Lu, Z. Tu, H. Li, and Q. Liu, "Neural transformation machine: A new architecture for sequence-to-sequence learning," *arXiv preprint arXiv:1506.06442*, 2015.

[45] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. ICASSP*, 2013, pp. 8599–8603.

[46] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[47] A. Karpathy and F.Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. CVPR*, 2015, pp. 3128– 3137.

[48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. CVPR*, 2015, pp. 3156–3164.

[49] Q. Wu, P. Wang, C. Shen, A. Dick, and A. Hengel, "Ask me anything: Free-form visual question answering based on knowledge from external sources," in *Proc. CVPR*, 2016, pp. 4622–4630.

[50] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. ICCV*, 2015, pp. 2425–2433.

[51] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[52] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to do next: Modeling user behaviors by time-LSTM." in *Proc. IJCAI*, 2017, pp. 3602–3608.

[53] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. CIKM*, 2017, pp. 2411–2414.

[54] D. Liao, W. Liu, Y. Zhong, J. Li, and G. Wang, "Predicting activity and location with multi-task context aware recurrent neural network." in Proc. *IJCAI*, 2018, pp. 3435–3441.

[55] M.-F. Wang, Y.-S. Lu, and J.-L. Huang, "Spent: A successive POI recommendation method using similarity-based POI embedding and recurrent neural network with temporal influence," in *Proc. BigCom*, 2019, pp. 1–8.

[56] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *Proc. 27th IJCAI*, 2018.

[57] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc.WWW*, 2018, pp. 1459–1468.

[58] Y. Wu, K. Li, G. Zhao, X. Qian, "Long- and short-term preference learning for next POI recommendation", in *Proc. CIKM*, 2019, pp. 2301-2304.

Yuxia Wu, received the B.S. degree from Zhengzhou University, Henan, China, in 2014, the M.S. degree from the Fourth Military Medical University, Xi'an, China, in 2017, and is currently working toward the Ph.D. degree at Xi'an Jiaotong University, Xi'an, China. She is mainly engaged in the research of multimedia mining and recommender systems.

Ke Li, received the B.S. degree from China University of Mining and Technology, Xuzhou, China, in 2017. He is currently pursuing the M.S. degree with the School of Software Engineering, Xi'an Jiaotong University, Xi'an, China.

Guoshuai Zhao, received the B.S. degree from Heilongjiang University, Harbin, China, in 2012, the M.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2015, and the Ph.D. degree at Xi'an Jiaotong University, Xi'an, China in 2019. He is now an Assistant Professor at the School of Software Engineering, Xi'an Jiaotong University. He is mainly engaged in the research of social media big data analysis and recommender systems.

Xueming Qian (M'10), received the B.S. and M.S. degrees from Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2008. He was awarded the Microsoft Fellowship in 2006. From 1999 to 2001, he was an Assistant Engineer at Shannxi Daily. Since 2008, he has been an Associate Professor in the School of Electronics and Information Engineering, Xi'an Jiaotong University. Now, he is an Associate Professor in the School of Electronics and Information Engineering, Xi'an Jiaotong University. He is the Director of SMILES LAB. He was a Visiting Scholar at Microsoft Research Asia from August 2010 to March 2011. His research interests include social media big data mining and search. He is a member of the IEEE, ACM, and Senior Member of CCF.