


Article

Long- and Short-Term Preference Modeling Based on Multi-Level Attention for Next POI Recommendation

Xueying Wang¹, Yanheng Liu^{1,2}, Xu Zhou^{2,3,*} , Zhaoqi Leng⁴ and Xican Wang⁴

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China; xueyingw20@mails.jlu.edu.cn (X.W.); yhliu@jlu.edu.cn (Y.L.)

² Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China

³ Center for Computer Fundamental Education, Jilin University, Changchun 130012, China

⁴ College of Software, Jilin University, Changchun 130012, China; lengzq20@mails.jlu.edu.cn (Z.L.); xcwang20@mails.jlu.edu.cn (X.W.)

* Correspondence: zhoux16@jlu.edu.cn

Abstract: The next point-of-interest (POI) recommendation is one of the most essential applications in location-based social networks (LBSNs). Its main goal is to research the sequential patterns of user check-in activities and then predict a user's next destination. However, most previous studies have failed to make full use of spatio-temporal information to analyze user check-in periodic regularity, and some studies omit the user's transition preference for the category at the POI semantic level. These are important for analyzing the user's preference for check-in behavior. Long- and short-term preference modeling based on multi-level attention (LSMA) is put forward to solve the above problem and enhance the accuracy of the next POI recommendation. This can capture the user's long-term and short-term preferences separately, and consider the multi-faceted utilization of spatio-temporal information. In particular, it can analyze the periodic hobbies contained in the user's check-in. Moreover, a multi-level attention mechanism is designed to study the multi-factor dynamic representation of user check-in behavior and non-linear dependence between user check-ins, which can multi-angle and comprehensively explore a user's check-in interest. We also study the user's category transition preference at a coarse-grained semantic level to help construct the user's long-term and short-term preferences. Finally, experiments were carried out on two real-world datasets; the findings showed that LSMA modeling outperformed state-of-the-art recommendation systems.

Keywords: next POI recommendation; recurrent neural network; LSTM; attentional mechanism; location-based social networks



Citation: Wang, X.; Liu, Y.; Zhou, X.; Leng, Z.; Wang, X. Long- and Short-Term Preference Modeling Based on Multi-Level Attention for Next POI Recommendation. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 323. <https://doi.org/10.3390/ijgi11060323>

Academic Editors: Wolfgang Kainz and Hartwig H. Hochmair

Received: 11 April 2022

Accepted: 25 May 2022

Published: 26 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the booming mobile internet has promoted the wide application of location-based social networks (LBSNs) with users sharing their location and life using check-in activities in LBSNs, such as Gowalla, Brightkite, Yelp, etc. [1]. The next point of interest (POI) recommendation, as one of the most significant services in LBSNs, is to recommend the next POI to a user based on his movement pattern. Different from the general POI recommendation, it takes into account the order of user check-ins, and is time-dependent and sequential. The next POI recommendation is of great significance to both the user and the merchant and can be applied for route plans, business advertising, and traffic prediction [2,3].

Early studies used the advantage of the Markov chain to predict process states to recommend the next POI [4]. A recurrent neural network (RNN) was then used in some studies to model check-in sequences for the next POI recommendation. As an improvement, long short-term memory (LSTM) based on spatio-temporal information has been proposed to model long trajectories and achieves good performance [5]. However, the user's check-ins are not only limited by time and location—there are also various pieces of contextual

information that are worth considering. As shown in Figure 1, user u has a check-in sequence with a list of activities on Sunday. The list contains many types of contextual information, such as time, geographical location, POI category, time difference Δt_{N-1} , distance Δd_{N-1} , etc. To solve the problem of data sparseness, spatial context information is exploited to capture the correlation between POIs for the POI recommendation [6]. Liu et al. proposed the ST-RNN model that takes spatio-temporal information into consideration on the basis of an RNN [7]. However, to be noticed, a user has a different pattern on every day of the week. So it is necessary to analyze the temporal information of check-ins and to study the daily check-in pattern of users in a week. That is to say, the temporal factor should be further used to study the periodicity of the user check-in, so as to mine the regularity of check-ins and improve the accuracy of prediction. In addition, the check-in trajectory shown in Figure 1 shows the user has a category transition preference at the category level. The user tends to shift from the category entertainment to the category residence. It is necessary to study the user's preference for the category of POI, and this can assist in the prediction of a specific POI. However, there is still room for improvement about how to utilize the POI category to enhance POI recommendation performance.

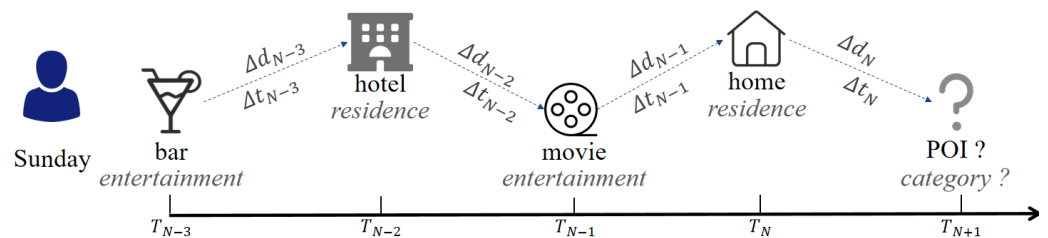


Figure 1. An example of the next POI recommendation.

Generally, a user's preference are complicated and change over time. The user's long-term preference can express their general interest, and the user's short-term preference can reflect their sudden interest. To analyze a user's long- and short-term performances, an LSTPM model based on spatial-temporal information is proposed by combining LSTM and RNN [8]. Moreover, the attention mechanism is introduced into the next POI recommendation to study the different influences of check-ins at different time steps on the next POI check-in [9]. However, it is necessary to fully utilize the attention mechanism to study the degree of influence of different factors in each user check-in. On the one hand, it is very important to mine non-linear dependence from the non-adjacent terms and find out the main intention of the user in the check-in sequence. On the other hand, the factor influencing the user's decision in each check-in activity is dynamic—it may be time, distance, or category for different users. It is also an urgent challenge to find out the decisive factors of each user check-in and obtain a multi-factorial dynamic representation of user check-ins.

To take full advantage of the check-in information and take the user's long- and short-term preferences into account, we construct a long- and short-term preference learning model based on a multi-level attention mechanism (LSMA) for the next POI recommendation. Firstly, the multi-factor dynamic representation of the user check-in is considered and the weights of different attributes in each check-in for a user are obtained. Secondly, the non-linear dependence between user check-ins is considered using the accurate representation of a user check-in, and the influence of the check-in at different time steps on the next POI check-in is obtained. This greatly improves the precision of the check-in presentation and the accuracy of the recommendation. In addition to directly using the contextual information mentioned above at the fine-grained POI level to recommend the next POI to a user, we also study the user's coarse-grained category transition preference at the semantic level, which can enhance the user's preference degree for POI. Experimental results on two large real-world datasets of Foursquare show that the LSMA performed significantly better than the other seven baselines in terms of recall and MAP. The main contributions of this study are as follows:

- We analyze the long- and short-term preferences of the user, respectively, and combine these to form the final user preference. A top-k POIs recommendation list is generated based on the next POI access probability computed by user preferences and all candidate POIs through a POI filter we designed.
- We utilize a multi-level attention mechanism to study the multi-factor dynamic representation of a user check-in behavior and non-linear dependence between check-ins in their check-in trajectory. This can learn the weights of different attributes in each check-in of a user and the influence of check-in at different time steps on the next POI check-in.
- We study the user's category transition preference at the semantic level to build the user's check-in representation using a category module we constructed. Furthermore, we consider the periodicity of the user check-in and mine the user's sequential pattern based on spatiotemporal context information. Both of these greatly promote the formation of user preferences and enhance the performance recommendation.

The remainder of the paper is organized as follows. We review the POI recommendation methods in Section 2. In Section 3, some preliminary investigations are described. Section 4 details the proposed POI recommendation approach. Section 5 provides the experimental results and the corresponding parameter analysis. Finally, Section 6 presents the conclusions.

2. Related Work

2.1. POI Recommendation

POI recommendations have attracted a great deal of attention in academia as a topic of wide currency and significance in the real world. Collaborative filtering (CF) is used to study the user's preference for POI in the early recommendation methods. Zhang et al. [10] put forward a cross-region collaborative filtering method to recommend new POIs; it involves mainly mine-hidden topics from the user's check-in records. Considering the compatibility of social relations, Xu et al. [11] studied the collaborative filtering model (SCCF) based on the social communication influence space and the individual attribute influence space in order to solve the problem of cold start. Other model-based CF techniques have also been used for POI recommendations, such as matrix factorization(MF) and probability matrix factorization(PMF) [12]. Davtalab et al. [13] noticed that there was an implicit association between users and POIs, and proposed a social spatio-temporal probability matrix decomposition (SSTPMF) model using POI similarity and user similarity, to model the similarity of social space, geographic space, and POI category space. It uses the potential similarity factor for a POI recommended multivariate reasoning method. Unfortunately, traditional POI recommendation methods ignore the sequential dependence of user check-in track sequences, which reduces the accuracy of the POI recommendation.

2.2. Next POI Recommendation

To describe the time series and movement pattern formed by the user check-in, some researchers have utilized the hidden Markov chain to recommend the next POI [14]. The Markov chain has a great advantage in the state prediction of sequential processes. Cheng et al. [15] put forward the FPMC-LR model combining POI transformation and the distance constraint of the first-order Markov chain. However, these recommendation algorithms based on the Markov chain have difficulty capturing the long sequence context.

With the application of recurrent neural networks, the above problems can be solved. The time-LSTM model based on LSTM is proposed to study the influence of the time interval on check-in behavior for the next POI recommendation [16]. Many factors (such as time, geographical location, category, etc.) during check-in will affect the users' check-in selection. Therefore, integrating different contextual information into the recommendation model can effectively mine the users' mobility patterns. A multi-task learning framework based on the LSTM network named iMTL is present to comprehensively consider the category and temporal information in the trajectory sequence for the next POI recommendation [17].

Numerous studies have shown that contextual information is valuable for studying user check-in behavior.

The attention mechanism can capture the degree of influence of different components; it is helpful to obtain more accurate user preference details to improve the performance of the POI recommendation model [18]. ATST-LSTM is based on the attention mechanism and contains spatio-temporal information, but it does not consider category transition preference information [19]. Zheng et al. [20] studied the effect of a hierarchical attention network (MAHAN) on memory enhancement for next POI recommendation. Liu et al. [21] proposed an attention-based, category-aware GRU model for the next POI recommendation, mainly focusing on the user's POI category preference. Xia et al. [22] designed intra- and inter-trajectory attention mechanisms to tackle the sparsity problem. Feng et al. [23] proposed an attentional recurrent network to predict user mobility from sparse trajectories. The network selects the most relevant historical trajectory to capture the periodic nature of human movement. The above studies showed that good recommendation performance could be achieved through the attention mechanism. However, they only consider the influence of each check-in on the last check-in in the trajectory by studying the importance of each RNN cell to the last cell—the influence degree of different attributes in each check-in is ignored, which is also very important for reflecting the user's check-in behavior more accurately.

Similar to human memory as studied by existing psychological research, user interests can also be divided into long-term and short-term. Long-term interest expresses the user's long-term focus, and short-term interest can grasp the rapid changes in a user's interest. Some studies only consider users' short-term preferences, such as ST-RNN [7], but others have found it necessary to study both long- and short-term preferences. Jannach et al. [24] found that both the long- and short-term interests of the user have a significant impact on the performance recommendation. Then a long and short-term combination model, based on location and category information, is proposed for the next POI recommendation [25]. The LSTPM integrates the long- and short-term preferences based on two LSTM networks, but does not consider time and category information [8]. Liu et al. [26] proposed the RTPM model based on LSTM, which considered both long- and short-term preferences and studied the real-time preference of a user through public interest in the short-term preference module. Although RTPM filters some POIs in the recommendation stage using POI category information to reduce the recommendation space, it also ignores the category of check-in when constructing the user movement pattern.

Unlike previous studies that do not fully consider the category trajectory of the user and the spatio-temporal information of check-in, our research uses the user's category transition preference to build a check-in representation and then considers the user's long- and short-term preferences, respectively, based on a multi-level attention mechanism.

3. Preliminaries

3.1. Observations of User's Trajectory

Two interesting observations were highlighted by the user's check-in behavior analysis in LBSNs.

- Obs.1: (Category transition preference.) On the semantic level, the user's check-in behavior has a category correlation. Figure 2a shows the category transition probability of users' check-ins in the Charlotte dataset. The correlation between the user's check-ins changes over time and is sequential. Different from other studies using a category as one attribute of check-in for a POI recommendation, we construct a category module of the proposed model to consider the category trajectory of the user and study the user's category transition preference at a coarse-grained category level. The module influences the user's preference for a specific POI. Moreover, there are hundreds of POI category labels in the datasets, which makes the prediction space very large and is not conducive to assisting in predicting the specific POI of the check-in. Therefore, inspired by [27], we summarize twelve coarse-grained categories on the basis of

existing categories. Note that each user’s dependence on the category transition preference is different. For instance, the user in Figure 2b goes from “residence” to “office”, and vice versa.

- Obs.2: (Periodic preference.) A user may have a fixed-mobile pattern on weekdays and another check-in pattern on weekends (Saturday and Sunday). However, except for distinguishing the user check-in pattern roughly on working days and weekends to analyze user periodic preference, a user’s check-in behavior will have a relatively obvious change pattern every day of the week in real life. As shown in Figure 3, for example, a user would like to visit the gym after the restaurant on Wednesday and to visit the company before the bar on Thursday. It is not appropriate to only study the periodicity of the user check-in movement on weekdays and weekends, so we analyze the spatio-temporal information of check-ins and study the periodic check-in pattern of users for every day of the week.

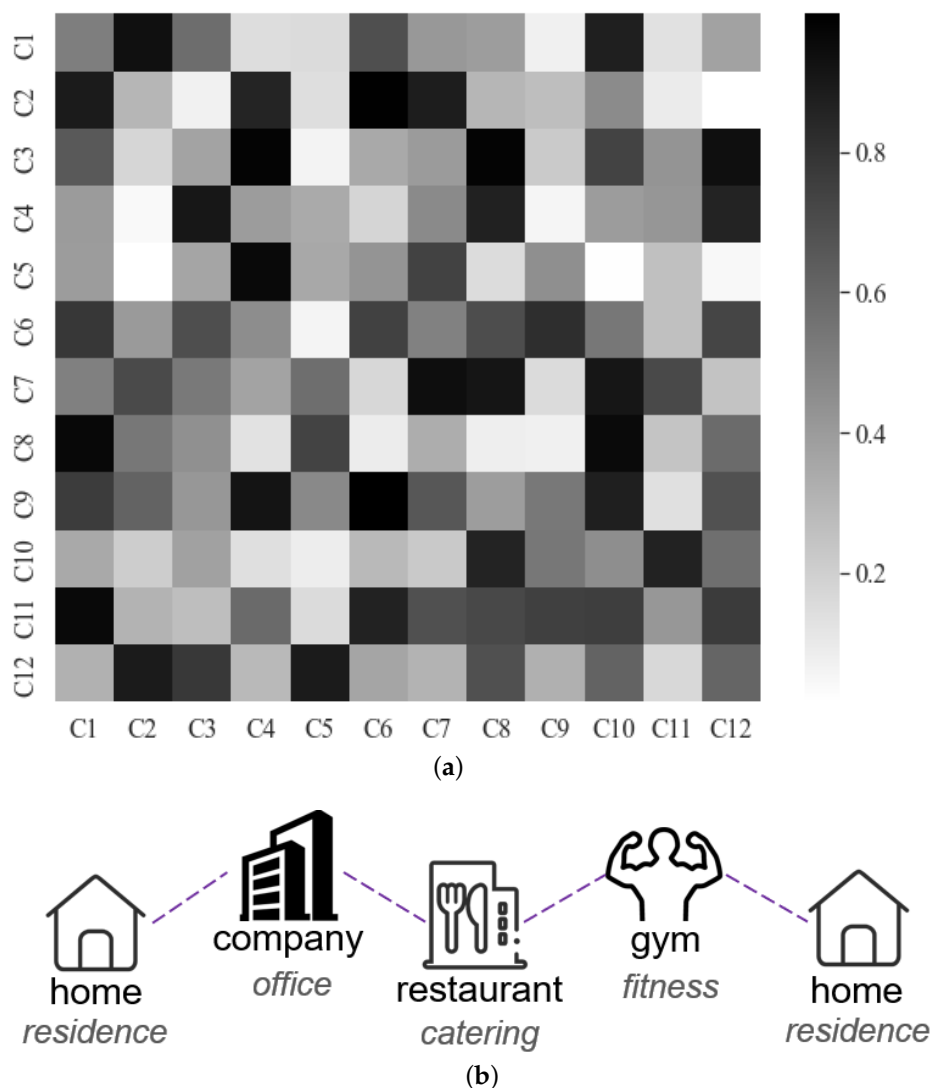


Figure 2. (a) Category transition probability of user’s check-ins in the Charlotte dataset (C1: Catering, C2: Entertainment, C3: Fitness, C4: Travel, C5: Office, C6: Residence, C7: Leisure, C8: Medical, C9: Store, C10: Transportation, C11: Livelihood, C12: Beauty); (b) An example of a user category trajectory sequence.

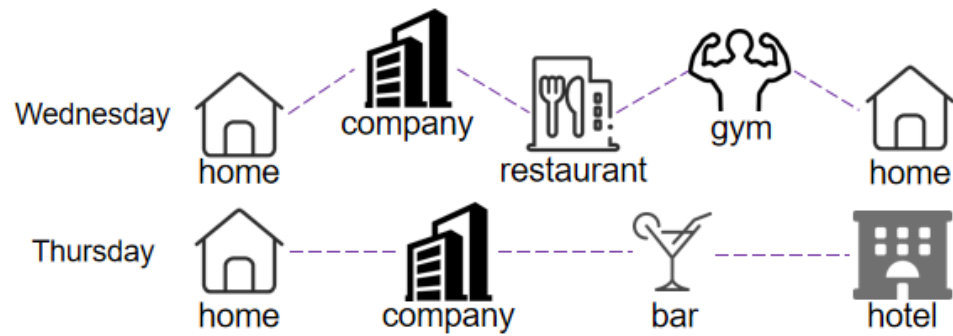


Figure 3. Example of a user check-in pattern on different day of the week.

3.2. Problem Statement

Let $U = \{u_1, u_2, \dots, u_{|U|}\}$ denote a set of users and $V = \{v_1, v_2, \dots, v_{|V|}\}$ denote a set of POIs, where $|U|$ represents the total number of users and $|V|$ represents the total number of POIs. Each POI $v \in V$ is a location associated with latitude, longitude and category information in LBSNs, such as a restaurant or bar.

A check-in activity of user $u \in U$ is a six-tuple $A_{t_k}^u = (u, v_{t_k}^u, l_v, c_v, t_k, w_{t_k})$, where $v_{t_k}^u$ represents user u who accesses the POI v at time t_k . The category of v is c_v , l_v is the geographical coordinate, w_{t_k} represents the time, and t_k is the w day of a week, such as Monday.

All the check-in activities of user u form their trajectory sequence $A_u = \{A_{t_1}^u, A_{t_2}^u, \dots, A_{t_N}^u\}$, where N is the total check-in number of user u . From the historical trajectory sequence A_u , we obtain the category sequence $C_u = \{C_{t_1}^u, C_{t_2}^u, \dots, C_{t_N}^u\}$ of u , where $C_{t_k}^u = (u, c_v)$. The short-term check-in sequence of user u can be extracted from A_u , denoted as $S_u = \{A_{t_{N-S+1}}^u, \dots, A_{t_m}^u, \dots, A_{t_N}^u\}$, where $A_{t_{N-S+1}}^u$ represents user u check-in for the first time in the short-term, and S is the total check-in number in the short-term.

Given A_u , C_u and S_u , the goal of the next POI recommendation is to predict a top- k list of POIs that user u is likely to visit at the next time t_{N+1} based on the two observations.

4. Proposed Method

The proposed model consists of four parts, as shown in Figure 4: (1) Category module: this captures the category transition preference of the user at the coarse-grained semantic level to assist the long- and short-term preference modules; (2) Long-term preference module: this obtains the user's long-term preference for POI based on the LSTM, and integrates the multi-level attention mechanism and the user's category transition preference; (3) Short-term preference module: this obtains the user's short-term preference for POI based on the RNN, and integrates the temporal attention mechanism and the user's category transition preference; (4) The output layer: the long and short-term preferences are combined to obtain the user's preference expression, and the final POI probability ranking list is formed with the calculation of user preference and candidate POIs based on a filter that we designed.

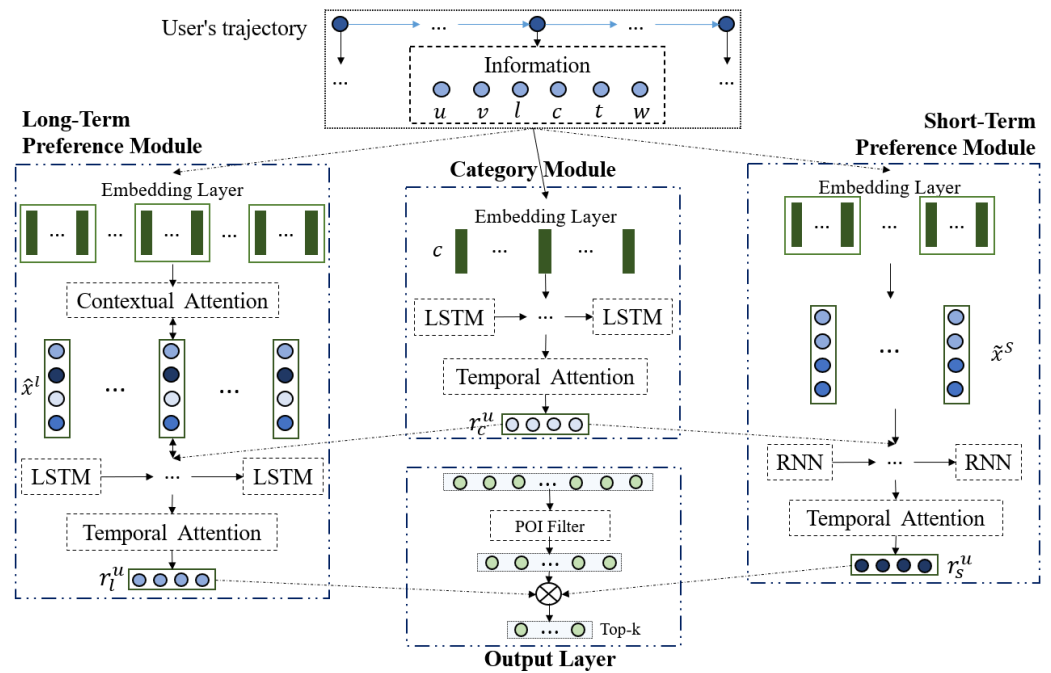


Figure 4. The proposed LSMA framework.

4.1. Category Module

We design the category module to infer the user's category transition preference, which obtains the category transition pattern when users visit the POI and participates in the POI recommendation as an auxiliary function. Due to the long sequence, the LSTM network is adopted to ensure recommendation accuracy.

We learn the user's category transition preference \mathbf{r}_c^u from the category sequence $C_u = \{C_{t_1}^u, C_{t_2}^u, \dots, C_{t_N}^u\}$, where each element of C_u is denoted as $C_{t_k}^u = (u, c_v)$. This indicates that the user u visits a POI v of category c_v at time t_k . The latent vector of the category module is defined as follows.

$$\mathbf{x}_{t_k}^c = \mathbf{W}^C \mathbf{c}_v + \mathbf{b}^C \quad (1)$$

where $\mathbf{W}^C \in \mathbb{R}^{d \times d}$ is the weight matrix, d is the dimension of the hidden vector, $\mathbf{b}^C \in \mathbb{R}^d$ is bias, and $\mathbf{c}_v \in \mathbb{R}^{D_c}$ indicates the embedding vector of the POI category c_v . Then, $\mathbf{x}_{t_k}^c$ is input into the LSTM network to infer the hidden state $\mathbf{h}_{t_k}^c$ of user u at time t_k .

$$\mathbf{h}_{t_k}^c = LSTM(\mathbf{x}_{t_k}^c, \mathbf{h}_{t_{k-1}}^c) \quad (2)$$

where $LSTM(\cdot)$ captures the sequential correlation of categories, and $\mathbf{h}_{t_{k-1}}^c$ indicates the check-in category up to t_{k-1} . Note that we treat the last hidden vector $\mathbf{h}_{t_N}^c$ as the representation of user u 's category transition preference.

$$\mathbf{r}_c^u = \mathbf{h}_{t_N}^c \quad (3)$$

4.2. Long-Term Preference Module

The long-term preference module obtains the user's long-term POI preference according to the contextual information of check-in activities and the multi-level attention mechanism.

4.2.1. Network Input

The historical check-in sequence of user u consists of all their check-ins. Each check-in is a six-tuple $A_{t_k}^u = (u, v_{t_k}^u, l_{t_k}, c_{t_k}, t_k, w_{t_k})$, which reflects the long-term preference of the user

to access the POI, so we utilize it to learn the user's preference at the POI level. Because the user's check-in is usually affected by the distance between the current location and the next location, as well as the time difference between the last check-in and the current check-in, the embedding layer of the long-term preference module should consider the impact of spatio-temporal contextual information on the check-in on the basis of the check-in location and time. The modeling of continuous check-in activities combined with the day of the week is more conducive to the study of the user's check-in regularity. In conclusion, the latent vector of the embedding layer of the long-term preference module is defined as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{t_k}^l = & \mathbf{W}_v \mathbf{v}_{t_k}^u + \mathbf{W}_l \mathbf{l}_v + \mathbf{W}_c \mathbf{c}_v + \mathbf{W}_t \mathbf{t}_k \\ & + \mathbf{W}_w \mathbf{w}_{t_k} + \mathbf{W}_d \mathbf{d}_{t_k} + \mathbf{W}_{td} \mathbf{td}_{t_k} + \mathbf{b} \end{aligned} \quad (4)$$

where \mathbf{W} is the weight matrix, \mathbf{b} is the bias term, $\mathbf{v}_{t_k}^u \in \mathbb{R}^{D_v}$ is the embedding of the POI number, $\mathbf{l}_v \in \mathbb{R}^{D_l}$ is the embedding of the POI location, $\mathbf{t}_k \in \mathbb{R}^{D_t}$ is the embedding of the access timestamp, $\mathbf{w}_{t_k} \in \mathbb{R}^{D_w}$ is the embedding of w_{t_k} , $\mathbf{d}_{t_k} \in \mathbb{R}^{D_d}$ is embedding based on the distance d_{t_k} between $l_{t_k}^u$ and $l_{t_{k-1}}^u$ of $v_{t_k}^u$ and $v_{t_{k-1}}^u$, and $\mathbf{td}_{t_k} \in \mathbb{R}^{D_{td}}$ is embedding based on the time difference td_{t_k} between t_k and t_{k-1} of $v_{t_k}^u$ and $v_{t_{k-1}}^u$.

The embedding layer of the long-term preference module has a total of seven input features, each of which marks an attribute of the current check-in. The influence degrees of these attributes on the current check-in are different. For example, a user is more likely to visit a POI close to their last check-in at a given time or is more likely to go to the "catering" category. The proportion of different attributes in the current check-in is researched by the contextual attention mechanism.

We use $\tilde{\mathbf{x}}(i, t_k)$ to represent the i -th feature of the k -th history check-in. For example, $\tilde{\mathbf{x}}(1, t_2)$ represents the POI number information of user u in the second check-in, and $\rho(i, t_k)$ represents the weight of the i -th attribute in the k -th check-in, and the softmax function is used for normalization.

$$\tilde{\rho}(i, t_k) = \tanh\left(\mathbf{W}_i \left[\mathbf{h}_{t_{k-1}}^l; \mathbf{c}_{t_{k-1}}^l\right] + \mathbf{W}_i^{\tilde{\mathbf{x}}} \tilde{\mathbf{x}}(i, t_k) + \mathbf{b}_i\right) \quad (5)$$

$$\rho(i, t_k) = \frac{\exp(\tilde{\rho}(i, t_k))}{\sum_{i=1}^I \exp(\tilde{\rho}(i, t_k))}, 1 \leq i \leq I \quad (6)$$

where I is the number of attributes, $\mathbf{W}_i \in \mathbb{R}^{d \times 2d}$, $\mathbf{W}_i^{\tilde{\mathbf{x}}} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_i \in \mathbb{R}^d$ are the parameters to be learned, and $\mathbf{c}_{t_{k-1}}^l$ is the cell state of the LSTM network at time t_{k-1} . Then, $\tilde{\mathbf{x}}(i, t_k)$ is multiplied by $\rho(i, t_k)$ to obtain the multi-factor dynamic representation of the check-in at time t_k under the contextual attention mechanism, and then the updated attribute embedding vector is connected to obtain the aggregation $\hat{\mathbf{x}}_{t_k}^l$ of the embedding layer based on the contextual attention mechanism.

$$\hat{\mathbf{x}}(i, t_k) = \tilde{\mathbf{x}}(i, t_k) \times \rho(i, t_k) \quad (7)$$

$$\hat{\mathbf{x}}_{t_k}^l = \sum_{i=1}^I \mathbf{W}(i) \hat{\mathbf{x}}(i, t_k) + \mathbf{b} \quad (8)$$

where $\mathbf{W}(i) \in \mathbb{R}^{d \times d}$ is the weight parameter to be learned corresponding to the i -th attribute, and $\mathbf{b} \in \mathbb{R}^d$ is the bias vector to learn.

To take into account the user's transition preference at the semantic level, we add the user's category preference on the basis of the embedded layer based on contextual attention and obtain the expression of the end-user's long-term check-in behavior as shown below.

$$\mathbf{x}_{t_k}^l = \hat{\mathbf{x}}_{t_k}^l + \lambda_c^l \mathbf{r}_c^u \quad (9)$$

where $\lambda_c^l \in \mathbb{R}^{d \times d}$ is the weight of user u 's category transition preference in the current check-in representation of the long-term preference module, and $\mathbf{x}_{t_k}^l$ is the final potential vector sent to the LSTM network to infer the hidden state at t_k .

$$\mathbf{h}_{t_k}^l = LSTM(\mathbf{x}_{t_k}^l, \mathbf{h}_{t_{k-1}}^l) \quad (10)$$

4.2.2. Temporal Attention

Check-in activities in the user's trajectory sequence are not all linearly correlated. However, the LSTM cannot solve this problem because it cannot obtain non-linear dependencies between check-ins. To compensate for this, we study the different effects of different check-ins on user preference; that is, the weights of different time steps of the check-in sequence should be learned to distinguish the important degree of each check-in in the historical check-ins. Therefore, we utilize the temporal attention mechanism to adaptively select relevant historical check-in activities to achieve a better recommendation of the next POI.

Let $\mathbf{H}^l \in \mathbb{R}^{d \times N}$ be a matrix composed of all hidden vectors $\{\mathbf{h}_{t_1}^l, \mathbf{h}_{t_2}^l, \dots, \mathbf{h}_{t_N}^l\}$ of the long-term preference module, where N is the length of the historical check-in sequence. The weight vector μ^l of the historical check-in is generated through the temporal attention mechanism, and the influence degree of the k -th historical check-in on the next check-in is measured with the weight μ_k^l corresponding to each $\mathbf{h}_{t_k}^l$.

$$\mu_k^l = \frac{\exp(g(\mathbf{h}_{t_k}^l, \mathbf{q}_u^l))}{\sum_{i=1}^N \exp(g(\mathbf{h}_{t_i}^l, \mathbf{q}_u^l))} \quad (11)$$

the attention function $g(\mathbf{h}_{t_k}^l, \mathbf{q}_u^l)$ is as follows.

$$g(\mathbf{h}_{t_k}^l, \mathbf{q}_u^l) = \frac{\mathbf{h}_{t_k}^l (\mathbf{q}_u^l)^T}{\sqrt{d}} \quad (12)$$

where \mathbf{q}_u^l is the query information of the long-term check-in sequence in the temporal attention mechanism, that is, the POI that the user checks-in next. The dot product attention is used as the attention function, since the d is small and the dot product attention is superior to the additive attention. The embedded representation of the query "next POI check-in" for all the historical check-ins is obtained. Then multiply the resulting weight vector μ^l by \mathbf{H}^l to obtain the user u 's preference representation of the long-term.

$$\mathbf{r}_t^u = \sum_{k=1}^N \mu_k^l \mathbf{h}_{t_k}^l \quad (13)$$

4.3. Short-Term Preference Module

The user's next POI check-in is influenced by the short-term preference represented by the user's recent check-in behavior in addition to the long-term preference represented by the historical check-in. Using the last S check-in as the user's short-term check-in sequence, it is represented as $S_u = \{A_{t_{N-S+1}}^u, \dots, A_{t_m}^u, \dots, A_{t_N}^u\}$, where $A_{t_{N-S+1}}^u$ represents the first check-in in the short-term, and one of them is set as $A_{t_m}^u$.

As with the long-term model, seven check-in features are extracted from the check-in activity tuple $A_{t_m}^u = (u, v_{t_m}^u, l_{t_m}, f_{t_m}, t_{t_m}, w_{t_m})$ of the short-term as the user check-in attributes

to be learned in the short-term preference module. The latent vector of the embedding layer of the short-term preference module is defined as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{t_m}^s = & \mathbf{W}_v^S \mathbf{v}_{t_m}^u + \mathbf{W}_l^S \mathbf{l}_v + \mathbf{W}_c^S \mathbf{c}_v + \mathbf{W}_t^S \mathbf{t}_m \\ & + \mathbf{W}_w^S \mathbf{w}_{t_m} + \mathbf{W}_d^S \mathbf{d}_{t_m} + \mathbf{W}_{td}^S \mathbf{td}_{t_m} + \mathbf{b}^S \end{aligned} \quad (14)$$

where \mathbf{W}^S is the weight matrix, and \mathbf{b}^S is the bias term.

Similarly, the users' category transition preference is also considered in the short-term preference module of this study, so short-term preference learning aggregation is defined as follows:

$$\mathbf{x}_{t_m}^s = \tilde{\mathbf{x}}_{t_m}^s + \lambda_c^s r_c^u \quad (15)$$

where $\lambda_c^s \in \mathbb{R}^{d \times d}$ is the weight of user u 's category transition preference in the current check-in representation of the short-term preference module. $\mathbf{x}_{t_m}^s$ enters the RNN as a potential vector for the user's check-in for the short-term.

Note that the RNN has the disadvantage of gradient disappearance. To avoid the problem of inaccurate recommendation results, we introduce the temporal attention mechanism to aggregate the hidden states generated by the RNN.

Let $\mathbf{H}^s \in \mathbb{R}^{d \times N}$ be a matrix composed of all hidden vectors $\{\mathbf{h}_{t_1}^s, \dots, \mathbf{h}_{t_m}^s, \dots, \mathbf{h}_{t_S}^s\}$ of the short-term preference module, where S is the length of the short-term check-in sequence. The weight vector μ^s of the short-term check-in is generated through the temporal attention mechanism of the short-term preference module, and the influence degree of the m -th short-term check-in on the next check-in is measured with the weight μ_m^s corresponding to each $\mathbf{h}_{t_m}^s$.

$$\mu_m^s = \frac{\exp(g(\mathbf{h}_{t_m}^s, \mathbf{q}_u^s))}{\sum_{i=1}^N \exp(g(\mathbf{h}_{t_i}^s, \mathbf{q}_u^s))} \quad (16)$$

where \mathbf{q}_u^s is the query information of the temporal attention mechanism, that is, the embedded representation of the query "next POI check-in" for all short-term check-ins. Then we multiply that weight vector μ^s times \mathbf{H}^s to obtain the short-term preference representation for user u .

$$\mathbf{r}_s^u = \sum_{m=1}^S \mu_m^s \mathbf{h}_{t_m}^s \quad (17)$$

4.4. Output Layer

4.4.1. POI Filter

Traditional recommendation systems usually recommend POI directly with all POI candidates as the POI, which undoubtedly increases the recommended computation time and space, and reduces the recommended accuracy. Different from other interest recommendations, such as music and movies, users' check-in is restricted by geographical location, so the next check-in of users will not be too far away from the current location. In addition, considering the time and transportation cost of each check-in, users are actually more inclined to access POIs that have been checked in before. However, users are also influenced by other users to access popular POIs. Therefore, these three factors must be considered at the same time when making recommendations for users.

Considering the above reasons, we designed a filter to sift some POIs as candidate POIs from all POIs. The POI filter has three filtering rules: (1) the POI that the user u has visited, (2) ten of the nearest POIs to the user's current location, (3) five of the most popular POIs among all users. The specific parameters are discussed in Section 5.

4.4.2. Recommend Top-k POIs

In order to comprehensively and dynamically study user preference, we use a weighted fusion of the user's long-term preference obtained by the long-term preference module and

their short-term preference obtained by the short-term preference module to compute the final user preference.

$$\mathbf{r}^u = \alpha \mathbf{r}_l^u + \beta \mathbf{r}_s^u \quad (18)$$

where α and β are the weights to learn. The next POI access probability normalized by the softmax function is defined as follows:

$$o_{t_{N+1}, v_k}^u = \frac{\exp(\mathbf{r}^u \mathbf{v}_k)}{\sum_{k=1}^P \exp(\mathbf{r}^u \mathbf{v}_k)} \quad (19)$$

where \mathbf{v}_k is an embedded representation of candidate POI v_k , and P is the total number of candidate POIs passed by a filter. Then, we can obtain the next visit probability of all candidate POIs in the output layer and the ranked POI list and recommend the top-k POIs for user u (Algorithm 1).

Algorithm 1 Training of LSMA

Input: User set U , historical check-in sequences set A^U , $iter_max$, parameter set Θ

Output: LSMA model R_u

```

1: //construct training instances
2: Initialize  $C^U = \phi$ ,  $S^U = \phi$ 
3: for each  $u \in U$  do
4:    $C_u \leftarrow \left\{ (u, c_{t_1}^u), (u, c_{t_2}^u), \dots, (u, c_{t_N}^u) \right\}$ 
5:    $S_u \leftarrow \left\{ A_{t_{N-S+1}}^u, \dots, A_{t_m}^u, \dots, A_{t_N}^u \right\}$ 
6: end for
7: for each  $u \in U$  do
8:   for each  $C_u, S_u$  and  $A_u^i \in A_u$  do
9:     Get the negative samples  $\{c'\}$ ,  $\{v'_s\}$  and  $\{v'\}$ 
10:   end for
11: end for
12: //parameter updating
13: for  $iter = 1; iter \leq iter\_max; iter++$  do
14:   for each  $u \in U$  do
15:     Select a random batch of instances
16:     for each  $\theta \in \Theta$  do
17:        $\theta \leftarrow \theta - \eta * \frac{\partial l}{\partial \theta}$ 
18:     end for
19:   end for
20: end for

```

4.5. Network Training

To effectively improve recommendation performance, we employ Bayesian personalized ranking (BPR) to define the loss function for training the network in the category, long- and short-term preference modules [28]. The data used for the modules consist of a set of triplets sampled from the original data, each triplet containing the user u and a pair of positive and negative samples. In the category module, the positive sample is the category that user u is currently accessing, the negative sample is all of the other categories. In the long- and short-term preference modules, the positive sample is the POI that user u is currently accessing; the negative sample is the POI close to the current check-in location considering the influence of geographical coordinates on the user's check-in.

The loss function of the category module is:

$$l_c = \sum_{(c > c') \in \Omega_c} \ln \left(1 + e^{-(o_i^c - o_i^{c'})} \right) \quad (20)$$

where c' is the negative category of c , Ω_c is the training example consisting of u, c, c' in the category module, o_t^c is the predicted probability of user u visiting the POI of category c at time t , and $o_t^{c'}$ is the predicted probability of user u visiting the POI of category c' at time t .

The loss function of the long-term preference module is:

$$l_l = \sum_{(v>v') \in \Omega_l} \ln \left(1 + e^{-\left(o_t^v - o_t^{v'}\right)} \right) \quad (21)$$

where v' is a negative sample of v in the long-term preference module, and Ω_l is the training example consisting of u, v, v' .

The loss function of the short-term preference module is:

$$l_s = \sum_{(v_s>v'_s) \in \Omega_s} \ln \left(1 + e^{-\left(o_t^{v_s} - o_t^{v'_s}\right)} \right) \quad (22)$$

where v'_s is a negative sample of v_s in the short-term preference module, and Ω_s is the training example consisting of u, v_s, v'_s .

To sum up, we design the total loss function by integrating the loss functions and regularization terms of the three modules, shown as follows:

$$l = l_c + l_l + l_s + \frac{\varepsilon}{2} \|\Theta\|^2 \quad (23)$$

where ε is the regularization coefficient, and $\Theta = (\mathbf{W}^C, \mathbf{b}^C, \mathbf{W}, \mathbf{b}, \mathbf{W}^S, \mathbf{b}^S, \mu)$ is the set of model parameters to learn. AdaGrad has been applied in large-scale learning tasks; thus, AdaGrad was employed to optimize the network parameters.

4.6. Complexity Analysis

In the LSMA training process, the computational complexity of all the category, long- and short-term preference modules is $O(d^2)$, where d is the embedding size. The training instance Ω_c with an average category sequence length of \bar{N} , training instance Ω_l with an average history sequence length of \bar{N} , and training instance Ω_s with an average short-term sequence length of \bar{S} are given, respectively. Each iteration trains overall complexity to $O((|\Omega_c| + |\Omega_l|) \cdot \bar{N} + |\Omega_s| \cdot \bar{S}) \cdot d^2$. That is, the complexity of LSMA has a quadratic relationship with the size of the embedding vector d .

5. Experiments

To verify the proposed method, we compared it with seven baselines on two public real-world check-in datasets named Charlotte (CHA) [17] and New York (NYC) [29] from Foursquare. All the algorithms were coded in Python 3.8 and the framework was TensorFlow 2.3.1. The experiments were conducted on the computer with CPU AMD Ryzen 5 3500U, Radeon Vega Mobile Gfx 2.10 GHz, and 16G RAM.

5.1. Datasets

The check-in data of the CHA were collected from January 2012 to December 2013 and the check-in data of NYC were collected from April 2012 to February 2013. The CHA dataset included 1580 users, 1791 POIs and 20,939 check-in records, and the NYC dataset included 1083 users, 38,336 POIs and 227,428 check-in records. In this study, each check-in record consisted of user, POI, the geographical coordinates of the POI, the timestamp of the check-in, the category of the POI, and the day of the check-in within the week. Similar to the work of Zhang et al. [17], we deleted the same POI that was accessed consecutively on the same day and deleted the inactive users who checked in less than eight times. For example, in the trajectory sequence $A \rightarrow A \rightarrow B \rightarrow A \rightarrow C$ on Sunday, the processed

sequence was $A \rightarrow B \rightarrow A \rightarrow C$. The 90% of check-ins of each user were used as the training set and the last 10% as the test set.

5.2. Methods for Comparison

We demonstrated the effectiveness of the LSMA method compared to the following seven baseline methods:

- PMF [12]: a recommendation algorithm designed based on the conventional probability matrix decomposition on the user-POI matrix.
- ST-RNN [7]: a next POI recommendation algorithm based on RNN, which integrates the spatio-temporal information into the latent vector.
- Time-LSTM [16]: equips the LSTM with a time gate to model continuous user actions in order to predict the next check-in POI.
- ATST-LSTM [19]: adds an attention mechanism on the basis of the LSTM network, and comprehensively considers spatio-temporal contextual information to improve the effectiveness of the next POI prediction.
- LSPL [25]: learns users' long- and short-term preferences by considering sequential information and the geographical location and category of the POI.
- iMTL [17]: a new interactive multi-task learning framework composed of a time-aware activity encoder, spatially aware position preference encoder, and task-specific decoder, mainly considering the next POI recommendation under uncertain check-in conditions.
- RTPM [26]: combines long- and short-term preferences and introduces public interest into the short-term preference to study the user's real-time interest.

5.3. Evaluation Metrics

All the methods discussed in this study calculate the dot product between the user representation and the POI representation to obtain the probability of the user accessing the POI the next time. In fact, the difference of each method lies in the different modeling of the user representation. To demonstrate the effectiveness of the methods, the recall rate ($Rec@k$) and mean average precision ($MAP@k$) were defined as follows:

$$Rec@k = \frac{1}{N} \sum_{u=1}^N Rec_u@k = \frac{1}{N} \sum_{u=1}^N \frac{|P_u(k) \cap V_u|}{|V_u|} \quad (24)$$

$$MAP@k = \frac{1}{N} \sum_{u=1}^N \frac{1}{map} \quad (25)$$

where $P_u(k)$ represents the set of top k POIs recommended to user u , V_u represents the POI set actually accessed by the user at the next time in the test set, and map represents the ranking of $P_u(k) \cap V_u$ in $P_u(k)$. Note that, in order to avoid a division error, we specify $\frac{1}{map} = 0$ when $P_u(k) \cap V_u = \phi$.

5.4. Parameter Setting

In the short-term preference module, we use the latest S check-ins as the user's short-term check-in sequence. The value of S should be as small as possible in order to study the user's recent interests and reduce the computing time and space. Considering indicator $Rec@10$ as an example, Figure 5a shows the performance of difference sequence length S . Considering the performance and computational complexity, S is set to be 6 as the length of the short-term trajectory sequence.

The LSMA model uses word embedding vectors to represent all users and POI information entering the model. The embedding dimensions of category, long- and short-term preference modules should be unified. We set embedding $D_c = D_v = D_l = D_d = D_t = D_{td} = D_w = d$, where d is the number of hidden units. Figure 5b shows the performance of

the difference embedding dimension d . Similarly, we chose d to be 128 as the embedding dimension considering the performance and computational complexity.

The setting of the negative sample number is also very important for model training. In the category module, the total number of POI categories is 12. In order to ensure the accuracy of the recommendation, we directly use all categories other than the current POI category as negative samples, i.e., the number of negative samples for each category sequence in the category module is 11. However, the total number of POIs is large, so we cannot use all POIs other than the current check-in POIs as negative samples. Therefore, we conducted experiments to find out the optimal number of negative samples was 5 in the long and short-term preference modules, as shown in Figure 5c.

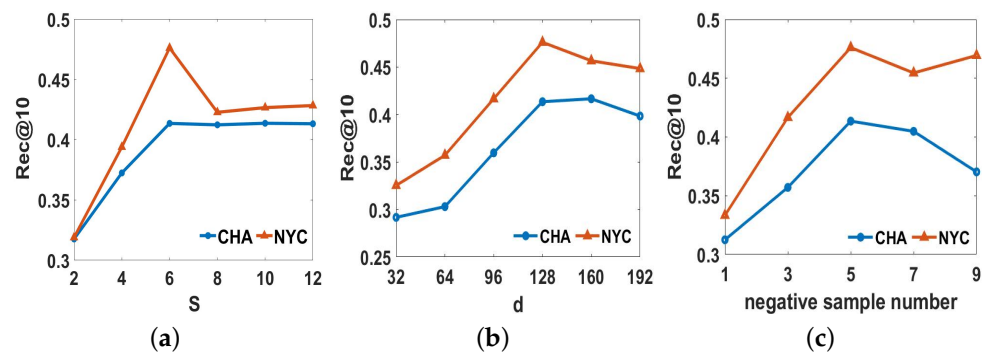


Figure 5. Performance of different hyper-parameters on the CHA and NYC datasets in LSMA. (a) different sequence length S ; (b) different embedding dimension d ; (c) different number of negative samples.

In the output layer of the LSMA, in order to reduce the computation and improve the recommendation accuracy, we designed a filter mechanism, which has two hyperparameters: the most popular POI number and the nearest POI number to the user's current location. We also conducted comparative experiments to explore the best settings of these two hyperparameters, as shown in Figure 6.

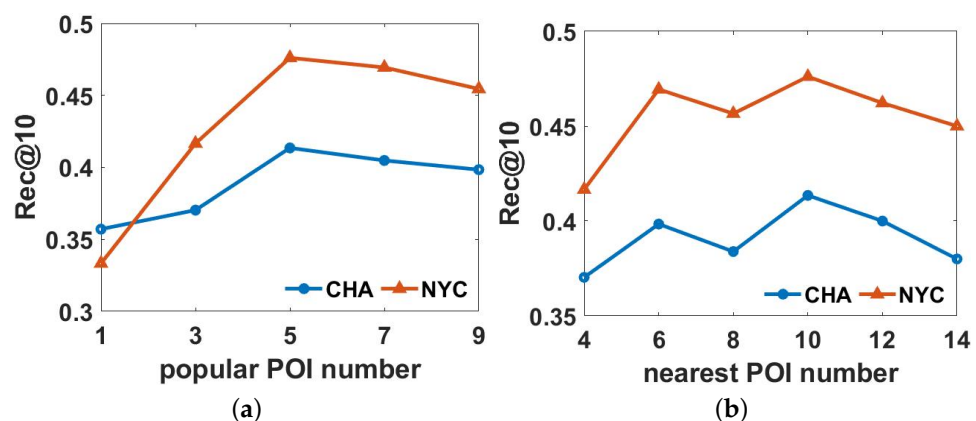


Figure 6. Performance of different hyperparameters on CHA and NYC datasets: (a) number of popular POIs; (b) number of nearest POIs.

We determined the best setting for each of the remaining hyperparameters as follows: (1) The layer number of the LSTM network of the category module and long-term preference module was 1, while the layer number of RNN of the short-term preference module was set to 1; (2) The learning rate of the three modules was 0.00001, 0.0001, and 0.0001, respectively; (3) The iteration number of the category module was 40, while the iteration number of the long- and short-term preference modules were both 20.

5.5. Results and Analysis

Tables 1 and 2 show the performance of the different methods; the results of the two evaluation indicators are listed when k is set to 5 and 10, respectively. It is obvious that the performance of PMF based on the non-neural network was the worst, lower than that of baseline methods based on RNN (ST-RNN, Time-LSTM, ATST-LSTM, iMTL, LSPL, RTPM), indicating that the neural network was very effective in modeling sequences. It was found that the *Rec* and *MAP* values of the Time-LSTM were higher than that of ST-RNN, which indicates that LSTM had better performance than RNN in long sequence modeling. Among the recommendation models based on the LSTM, the RTPM performed best on both the CHA and NYC datasets. This demonstrates the importance of considering both the user's long- and short-term preferences and the effectiveness of filtering some qualified POIs to make recommendations. However, the LSMA we proposed had a better recommendation performance than the RTPM. For example, the *Rec@5* value of the RTPM on the CHA dataset was 0.1569, while the *Rec@5* value of the LSMA was 0.2838. The LSMA increased by 80.87% on *Rec@5*. This was mainly because the LSMA considers the users' long-term and short-term preferences simultaneously. Secondly, the LSMA mines as much information contained in user check-in sequences, as well as the users' movement patterns in the category, as possible, and models user behavior in more detail and in various aspects. Finally, the LSMA designs a multi-level attention mechanism to consider the weight of each check-in attribute of the POI and the influence degree of each check-in comprehensively.

Table 1. The recommendation result of different methods for the CHA dataset.

Evaluation	<i>Rec@5</i>	<i>Rec@10</i>	<i>MAP@5</i>	<i>MAP@10</i>
PMF	0.0668	0.0943	0.0141	0.0213
ST-RNN	0.0790	0.1679	0.0213	0.0513
Time-LSTM	0.0843	0.1842	0.0425	0.0609
ATST-LSTM	0.1003	0.2083	0.0499	0.0719
iMTL	0.1138	0.2634	0.0733	0.0809
LSPL	0.1319	0.3201	0.0809	0.0957
RTPM	0.1569	0.3508	0.0813	0.1023
LSMA	0.2838	0.4135	0.1036	0.1234

Table 2. The recommendation result of different methods for the NYC dataset.

Evaluation	<i>Rec@5</i>	<i>Rec@10</i>	<i>MAP@5</i>	<i>MAP@10</i>
PMF	0.0322	0.105	0.022	0.0263
ST-RNN	0.0406	0.1464	0.025	0.0312
Time-LSTM	0.0794	0.1938	0.0272	0.0458
ATST-LSTM	0.1224	0.2269	0.0421	0.0621
iMTL	0.1584	0.2801	0.0693	0.0857
LSPL	0.1702	0.2964	0.0725	0.0929
RTPM	0.2057	0.3761	0.0806	0.1029
LSMA	0.2857	0.4761	0.1098	0.1315

To verify the performance obtained by considering the different contributions of the category module, short-term preference module, contextual attention mechanism, temporal attention mechanism, and POI filter, we designed five different variants of the LSMA: (1) LSMA-C removes the category module; the users' preferences at the semantic level are no longer considered; (2) LSMA-S removes the short-term preference module; the user's short-term preference is no longer considered; (3) LSMA-CA removes the contextual attention mechanism from the long-term preference module; (4) LSMA-TA removes the temporal attention mechanism from the long-term preference module; (5) LSMA-Filter removes the filter from the output layer.

Figure 7 illustrates the performance of the LSMA compared to the five variants. From Figure 7, it was found that the LSMA performed better than its variants in recall and MAP.

The performance of LSMA-C was better than that of LSMA-S, LSMA-CA, LSMA-Filter, and LSMA-TA, which indicates that the short-term preference module and the attention mechanism play an important role in the LSMA, and the category module assists the long- and short-term modules. In addition, the performance of the LSMA-S is the worst of all variants for three evaluation indicators, which confirms the important influence of short-term preference on the user's check-in behavior. The $Rec@10$ values of the LSMA-C, LSMA-S, LSMA-CA, LSMA-TA, and LSMA-Filter on the CHA dataset were 0.3225, 0.2439, 0.303, 0.3236, and 0.3508, respectively, while the $Rec@10$ value of the LSMA was 0.4135. The LSMA yielded an increase of 28.22%, 69.54%, 36.47%, 27.78%, and 17.87% on $Rec@10$, respectively. The necessity of exploiting the temporal attention and contextual attention mechanisms can be inferred. In summary, the five components were indispensable, and they enabled the LSMA to achieve a significant performance improvement.

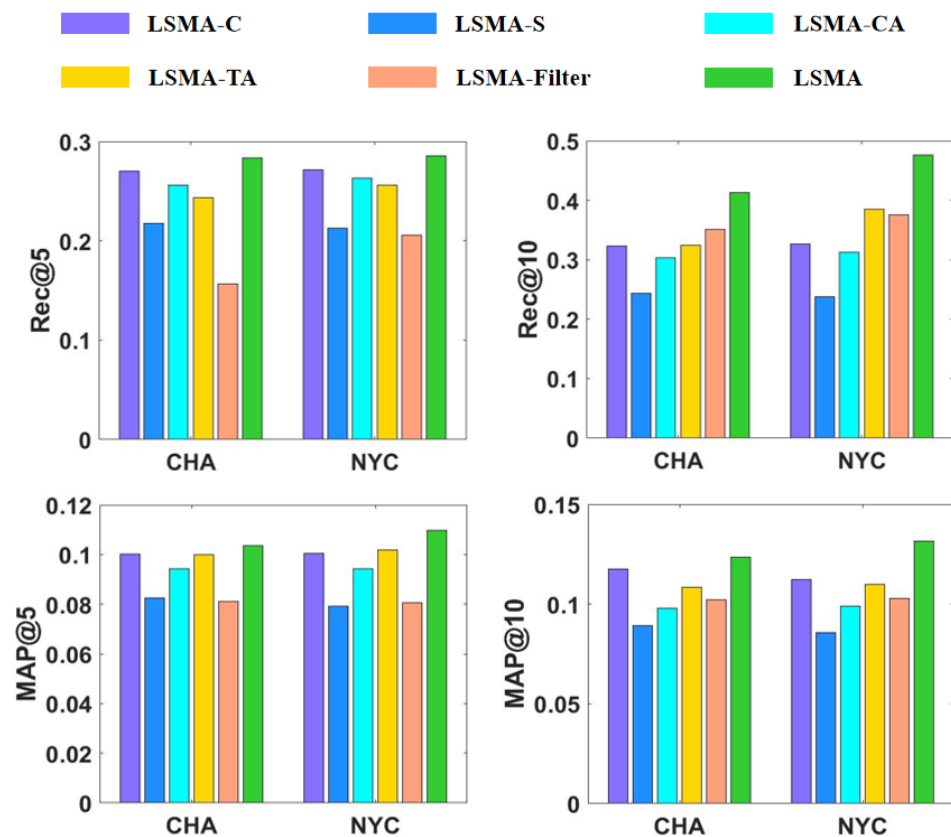


Figure 7. Performance comparison of LSMA and its variants on the CHA and NYC datasets.

6. Conclusions

A next POI recommendation algorithm LSMA was proposed in this paper, which models the user's long- and short-term preferences based on multi-level attention. Specifically, the LSMA designs the category module to obtain the category transition preference of users and participates in check-in representation in long- and short-term preference modules as an auxiliary function. The long-term preference module of the LSMA is constructed to achieve the users' long-term POI preferences according to the LSTM network and multi-level attention mechanism. The short-term preference module of the LSMA is used to obtain users' short-term POI preferences according to the RNN and the attention mechanism. Moreover, focusing on the key attribute of the user's check-in and the key time step of the check-in sequence, the user's movement behavior patterns can be fully mined through a multi-level attention mechanism. The experimental results showed that the LSMA performance was superior to that of the other seven comparative recommendation methods for the next POI recommendation.

In the future, our work will continue to optimize the LSMA model by considering the user's comment information and further study the privacy protection for the next POI recommendation.

Author Contributions: Conceptualization, Xueying Wang and Xu Zhou; methodology, Xueying Wang; validation, Xueying Wang, Yanheng Liu, Zhaoqi Leng, and Xican Wang; formal analysis, Xueying Wang; writing—original draft preparation, Xueying Wang; writing—review and editing, Xueying Wang and Xu Zhou; supervision, Yanheng Liu and Xu Zhou; funding acquisition, Xu Zhou and Yanheng Liu. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant Nos. 61806083, 61872158, and 62172186) and the Fundamental Research Funds for the Central Universities, JLU under Grant No. 93K172021Z02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on the website: <https://sites.google.com/site/yangdingqi/home/foursquare-dataset>, <https://www.foursquare.com> (accessed on 31 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, M.; Shen, D.; Kou, Y.; Nie, T.; Yu, G. Next point-of-interest recommendation by sequential feature mining and public preference awareness. *J. Intell. Fuzzy Syst.* **2021**, *40*, 4075–4090. [[CrossRef](#)]
2. Jiao, W.; Fan, H.; Midtbø, T. A grid-based approach for measuring similarities of taxi trajectories. *Sensors* **2020**, *20*, 3118. [[CrossRef](#)] [[PubMed](#)]
3. Sarkar, J.L.; Majumder, A.; Panigrahi, C.R.; Roy, S. MULTITOUR: A multiple itinerary tourists recommendation engine. *Electron. Commer. Res. Appl.* **2020**, *40*, 100943. [[CrossRef](#)]
4. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
5. Zhan, G.; Xu, J.; Huang, Z.; Zhang, Q.; Xu, M.; Zheng, N. A semantic sequential correlation based LSTM model for next POI recommendation. In Proceedings of the 20th IEEE International Conference on Mobile Data Management, MDM 2019, Hong Kong, China, 10–13 June 2019; pp. 128–137.
6. Chang, B.; Jang, G.; Kim, S.; Kang, J. Learning Graph-Based Geographical Latent Representation for Point-of-Interest Recommendation. In Proceedings of the CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 19–23 October 2020; pp. 135–144.
7. Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 194–200.
8. Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q.V.H.; Yin, H. Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 214–221.
9. Wang, K.; Wang, X.; Lu, X. POI recommendation method using LSTM-attention in LBSN considering privacy protection. *Complex Intell. Syst.* **2021**, *1*–12. [[CrossRef](#)]
10. Zhang, W.; Wang, J. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1221–1230.
11. Xu, Y.N.; Xu, L.; Huang, L.; Wang, C.D. Social and content based collaborative filtering for point-of-interest recommendations. In Proceedings of the International Conference on Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017; pp. 46–56.
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
13. Davtalab, M.; Alesheikh, A.A. A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization. *Knowl. Inf. Syst.* **2021**, *63*, 65–85. [[CrossRef](#)]
14. Lv, Q.; Qiao, Y.; Ansari, N.; Liu, J.; Yang, J. Big data driven hidden markov model based individual mobility prediction at points of interest. *IEEE Trans. Veh. Technol.* **2017**, *66*, 5204–5216. [[CrossRef](#)]

15. Cheng, C.; Yang, H.; Lyu, M.R.; King, I. Where you like to go next: Successive point-of-interest recommendation. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2605–2611.
16. Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; Cai, D. What to do next: Modeling user behaviors by Time-LSTM. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017; pp. 3602–3608.
17. Zhang, L.; Sun, Z.; Zhang, J.; Lei, Y.; Li, C.; Wu, Z.; Kloeden, H.; Klanner, F. An interactive multi-task learning framework for next POI recommendation with uncertain check-ins. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2020; pp. 3551–3557.
18. Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; Liu, T. Sequential click prediction for sponsored search with recurrent neural networks. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1369–1375.
19. Huang, L.; Ma, Y.; Wang, S.; Liu, Y. An attention-based spatiotemporal LSTM network for next POI recommendation. *IEEE Trans. Serv. Comput.* **2019**, *14*, 1585–1597. [[CrossRef](#)]
20. Zheng, C.; Tao, D.; Wang, J.; Cui, L.; Ruan, W.; Yu, S. Memory augmented hierarchical attention network for next point-of-interest recommendation. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 489–499. [[CrossRef](#)]
21. Liu, Y.; Pei, A.; Wang, F.; Yang, Y.; Zhang, X.; Wang, H.; Dai, H.; Qi, L.; Ma, R. An attention-based category-aware GRU model for the next POI recommendation. *Int. J. Intell. Syst.* **2021**, *36*, 3174–3189. [[CrossRef](#)]
22. Xia, T.; Qi, Y.; Feng, J.; Xu, F.; Sun, F.; Guo, D.; Li, Y. AttnMove: History enhanced trajectory recovery via attentional network. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; pp. 4494–4502.
23. Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; Jin, D. DeepMove: Predicting human mobility with attentional recurrent networks. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018; pp. 1459–1468.
24. Jannach, D.; Lerche, L.; Jugovac, M. Adaptation and evaluation of recommendations for short-term shopping goals. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, 16–20 September 2015; pp. 211–218.
25. Wu, Y.; Li, K.; Zhao, G.; Qian, X. Long- and short-term preference learning for next POI recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2301–2304.
26. Liu, X.; Yang, Y.; Xu, Y.; Yang, F.; Huang, Q.; Wang, H. Real-time POI recommendation via modeling long-and short-term user preferences. *Neurocomputing* **2022**, *467*, 454–464. [[CrossRef](#)]
27. Ye, J.; Zhu, Z.; Cheng, H. What's your next move: User activity prediction in location-based social networks. In Proceedings of the 2013 SIAM International Conference on Data Mining, Philadelphia, PA, USA, 6–13 October 2013; pp. 171–179.
28. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
29. Yang, D.; Zhang, D.; Zheng, V.W.; Yu, Z. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Trans. Syst. Man, Cybern. Syst.* **2015**, *45*, 129–142. [[CrossRef](#)]