

Article

Modeling Long and Short Term User Preferences by Leveraging Multi-Dimensional Auxiliary Information for Next POI Recommendation

Zheng Li ^{1,2,3}, Xueyuan Huang ¹, Liupeng Gong ¹, Ke Yuan ^{1,*} and Chun Liu ¹

¹ College of Computer and Information Engineering, Henan University, Kaifeng 475004, China; lizheng@henu.edu.cn (Z.L.); huangxy@henu.edu.cn (X.H.); gongliupeng@henu.edu.cn (L.G.); liuchun@henu.edu.cn (C.L.)

² Henan Engineering Laboratory of Spatial Information Processing, Henan University, Kaifeng 475004, China

³ Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China

* Correspondence: yuanke@henu.edu.cn

Abstract: Next Point-of-Interest (POI) recommendation has shown great value for both users and providers in location-based services. Existing methods mainly rely on partial information in users' check-in sequences, and are brittle to users with few interactions. Moreover, they ignore the impact of multi-dimensional auxiliary information such as user check-in frequency, POI category on user preferences modeling and the impact of dynamic changes in user preferences over different time periods on recommendation performance. To address the above limitations, we propose a novel method for next POI recommendation by modeling long and short term user preferences with multi-dimensional auxiliary information. In particular, the proposed model includes a static LSTM module to capture users' multi-dimensional long term static preferences and a dynamic meta-learning module to capture users' multi-dimensional dynamic preferences. Furthermore, we incorporate a POI category filter into our model to comprehensively simulate users' preferences. Experimental results on two real-world datasets demonstrate that our model outperforms the state-of-the-art baseline methods in two commonly used evaluation metrics.

Keywords: next POI recommendation; multi-dimensional auxiliary information; cold-start; meta-learning; POI category filter



Citation: Li, Z.; Huang, X.; Gong, L.; Yuan, K.; Liu, C. Modeling Long and Short Term User Preferences by Leveraging Multi-Dimensional Auxiliary Information for Next POI Recommendation. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 352. <https://doi.org/10.3390/ijgi12090352>

Academic Editors: Wolfgang Kainz and Hartwig H. Hochmair

Received: 17 July 2023

Revised: 18 August 2023

Accepted: 22 August 2023

Published: 25 August 2023



Copyright: © 2023 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

Next POI recommendation, as a research hotspot in location-based social networks, not only allows users to find their desired POIs quickly, but also provides opportunities for merchants to cater to user preferences. However, in commonly used datasets for next POI recommendation, the user-POI interaction matrix is relatively sparse, making accurate recommendation challenging based on limited information. This is often referred to as cold start problem in recommendation systems. One feasible solution to address the cold start problem is to leverage additional auxiliary information such as POI information, user profiles, social relationships [1], spatial information [2], social network detection [3]. However, due to privacy concerns, the obtained auxiliary information related to users is usually scarce, which makes it extremely difficult to accurately capture users' preferences based on limited information. Moreover, it is not easy to effectively simulate dynamic changes in user preferences for cold start users.

In real life, there are a large number of users who have some check-in activities in the past period, but have few or almost no check-in activities in the recent period. We define these users as recent check-in cold start users. Currently, there are methods [4,5] that study how to capture dynamic preferences of users changing with spatio-temporal factors. However, these methods mainly focus on active users with frequent check-ins,

filtering out cold start users with fewer check-in activities and cold start POIs with fewer user check-ins. Therefore, there is a lack of a next POI recommendation model for recent check-in cold-start users, which integrates multi-dimensional auxiliary information from user check-in activities to effectively simulate user dynamic preferences.

Recently, research shows that meta-learning [6] can endow models with the ability to “learn how to learn”, enabling them to quickly adapt the global model to new tasks (users) with little or no interaction information, so as to perform accurately recommendation for new tasks (users). Consequently, faced with cold start data in user check-ins, researchers began to integrate meta-learning into next POI recommendation systems, transforming the recommendation task into a few-shot learning task. For example, Zhang et al. [7] captured the transition information between two adjacent POIs in a user’s long term historical check-in sequence as transferable generalized knowledge, and then simulated the transition preferences of cold start users to visit next POI. Vinayak et al. [8] conducted lightweight unsupervised cluster-based information propagation based on long term check-in sequence information between users with similar preferences and locations in data-rich regions, thus improving the quality of recommendation in data-scarce regions. Although existing next POI recommendation approaches based on meta-learning has achieved inspiring results, there are still two limitations.

(1) Existing methods mainly learn users’ long term static preferences according to their historical check-in sequences, and ignore the impact of multi-dimensional dynamic preferences of recent check-in cold start users on next POI recommendation.

(2) Ignoring auxiliary information such as user check-in frequency and POI category, there is a lack of an integrated recommendation model for next POI to alleviate user cold start problem.

To address the issues as described above, we propose a POI Recommendation model based on Meta-Learning (ML-POIRec) for recent check-in cold start users. This model takes into account multi-dimensional auxiliary information in user check-in activities. Firstly, it utilizes a static LSTM module to capture users’ multi-dimensional long term static preferences based on their check-in activities within historical time periods. Then it employs a dynamic meta-learning module to capture users’ multi-dimensional dynamic preferences based on their check-in activities within the current time period. In addition, we incorporate a POI category filter into our model to filter candidate POIs, and calculate the preference score for each POI by taking dot product between filtered POIs and the user comprehensive preference obtained by integrating the above two user preferences, and rank preference scores for next POI recommendation. Finally, to validate the effectiveness of the proposed model, we conduct a series of experiments on two well-known LBSN datasets. The results indicate that ML-POIRec outperforms seven baseline methods on both datasets.

To summarize, the main contributions of this paper are listed as follows:

(1) We propose a novel approach for next POI recommendation by modeling long and short term user preferences with multi-dimensional auxiliary information, which greatly improves the recommendation performance.

(2) We deeply mine users’ multi-dimensional long term static preferences and dynamic preferences by static LSTM module and dynamic meta-learning module, respectively. Moreover, we incorporate a POI category filter into our model. These factors are considered comprehensively to more accurately simulate users’ preferences.

(3) Extensive experiments on two real-world datasets are conducted to evaluate the performance of the proposed model. The results show that our ML-POIRec model achieves significant performance improvements over the state-of-the-art baseline methods.

The remainder of this paper is organized as follows. Section 2 reviews the related work. In Section 3, the problem definition and data analysis are presented, along with an explanation of key symbols used in this paper. In Section 4, we introduce the details of the proposed ML-POIRec model. Next, the effectiveness of ML-POIRec is evaluated in

Section 5. Section 6 discusses the practical implications of ML-POIRec. Finally, Section 7 concludes this paper.

2. Related Work

2.1. Next POI Recommendation Based on User's Long and Short Term Preferences

Users long term preferences and short term preferences in the current period are likely to be different. To address this issue, researchers considered the applicability factors [9,10] of POI over time from the perspective of POI, but lacked personalized user needs. Later, researchers focused on users themselves and utilized LSTM network [11,12] and attention mechanism [13,14] to obtain users' long and short term preferences based on their visiting sequence patterns for POI recommendation. Li et al. [15] and Zhao et al. [16] leveraged LSTM based and gated LSTM framework to learn spatial temporal contexts for next POI recommendation, respectively. Cui et al. [17] adopted context for non-consecutive modeling, using LSTM to obtain time effects in long term module and constructing four short term sequences to capture the influence of different factors. Liu et al. [18] utilized variants of GRU to obtain users' sequential and spatio-temporal preferences, and then perform POI recommendation.

Overall, LSTM based approaches mentioned above for modeling long and short term preferences have shown inspiring results, they overlooked the impact of multi-dimensional auxiliary information on next POI recommendation. Moreover, they directly filtered out the data of cold start users and ignored the recommendations for cold start users. In contrast, our proposed model comprehensively considers multi-dimensional information such as spatio-temporal context information, user check-in frequency, POI category, etc. Furthermore, we leverage multi-dimensional static preference LSTM module and multi-dimensional dynamic preference meta-learning module to capture user's long and short term preferences, respectively. Furthermore, we consider the impact of POI categories on user preferences to further improve the recommendation accuracy.

It is worth noting that in our approach, users' long term static preferences generally refer to their habitual and stable preferences. While users' short dynamic preferences indicate users' potential preferences for visiting POIs in the current time period (such as recent one or several months). Such dynamic preferences are different from users' interests drift [19–21] across geographical regions (e.g., home-town and out-of-town).

2.2. Meta-Learning for Cold-Start Recommendation

In location-based social networks, since users must physically visit POIs in the real world, the majority of users only interact with a few POIs, and there are many users who have little or no check-in activities in recent time period, so that most models are suffered from the sparsity of user check-in records [22]. While meta-learning [23] has become a popular few-shot learning method, and researchers have introduced it into recommendation systems to alleviate the cold start problem. Moreover, existing work has shown that applying meta-learning to recommendation systems can effectively alleviate the cold start problem of user-item interactions. For example, the early work [24] followed settings of few-shot learning, regarded each user's recommendation as a task and learned from a few user-item interactions and user information. In addition, some methods utilize rich auxiliary information, such as user information and item attribute information. For instance, Lee et al. [25] identified relatively reliable candidate items based on user information and item attributes, thereby making item recommendation. Dong et al. [26] also relied on auxiliary information from users and items to extend two designed memory matrices to provide personalized initialization for each user. Besides, meta-learning has also been used for click-through rate prediction [27], which can generate ideal embeddings of new items through continuous updating of meta-learning, and then achieve item recommendation based on global knowledge of all tasks.

Since meta-learning has been proven to be successful for general recommendation in cold start scenarios [28,29], researchers try to apply meta-learning to next POI recommenda-

tion. For example, Chen et al. [30] proposed to incorporate meta-learning and non-uniform sampling strategies into next POI recommendation to solve the problem of data sharing limitations between different cities and the diversity of graph search modes of different users in different cities, effectively alleviating the problem of data scarcity and POI diversity. Tan et al. [31] adopted GRU-ODE-Bayes model for city specific information modeling to solve the problem of data sharing between cities, which also utilized meta-learning to optimize parameters on cities with sufficient data and train a good global shared parameter, so as to effectively adapt to cities with insufficient data and improve the recommendation performance.

However, the above methods do not take into account users' recent dynamic preferences, so it is likely to recommend POIs that do not match users' preferences in the current time period. In addition, the above methods ignore the impact of auxiliary information such as spatio-temporal information, POI category, and check-in frequency on recommendation results. In our work, by integrating multi-dimensional auxiliary information to deeply mine users' multi-dimensional long term static preferences and recent dynamic preferences, we can more accurately simulate users' preferences at different time periods, consequently improving the recommendation performance.

3. Preliminaries and Data Analysis

In this section, we first present the relevant definitions of our ML-POIRec model and summarize primary notations and their meanings used in this paper in Table 1, and then analyze the properties of the datasets.

Table 1. Summary of key notations.

Notation	Description
U, L	Users set, POIs set
T	The set of time periods
G	The set of geographic interval labels
C	The set of POI categories
r_e	A check-in activity
o_u^t	The set of check-in activities of user u in the time period t
d	Latent dimension
x_{l_i}, a_{l_i}	Original encoding and embedded representation of POI l_i
θ^t	Model parameter of meta-learning module
$p_d^{(t)}$	Multi-dimensional dynamic preference representation for a user
$p_s^{(t)}$	Multi-dimensional static preference representation for a user
ω_u	Model parameter of LSTM module
S_u^t, Q_u^t	The corresponding support set and query set of user u in time period t
$P(T)$	Representation of meta-learning task distribution

3.1. Preliminaries

Here, we give some definitions of terms. Assume that $U = \{u_1, u_2, \dots, u_{|U|}\}$ is users set, where $|U|$ denotes the total number of users, $L = \{l_1, l_2, \dots, l_{|L|}\}$ is POIs set, where $|L|$ denotes the total number of POIs that all users visited. $C = \{c_1, c_2, \dots, c_{|C|}\}$ is the set of categories to which all POIs belong, $|C|$ represents the number of categories in the set.

Definition 1. *POI.* A POI represents a specific spatial location such as a restaurant, a cinema. Here, each POI is represented as a triple: (Pid, lat, lng) , where Pid represents a POI identifier (POI ID), lat and lng represent the latitude and longitude coordinates of the location of a POI, respectively.

Definition 2. *Check-in time period set.* In our model, we divide the check-in time of all users visiting all POIs into different time periods, denoted as $T = \{t_1, t_2, \dots, t_{|T|}\}$, $|T|$ represents the number of time periods in the set. Furthermore, we regard the time period to which the check-in time of a user visiting a POI belongs as the time label of the user visiting the POI.

Definition 3. *User check-in sequence.* The historical check-in sequence of user u is represented by $seq(u) = \{o^{t_1}, o^{t_2}, \dots, o^{t_T}\}$, where $o_u^{t_i} = \{r_1, r_2, \dots, r_m\}$ is the set of user check-in activities in the time period t_i , and m is the number of check-in activities.

Definition 4. *Geographic interval label set.* The geographic interval label set $G = \{g_1, g_2, \dots, g_{|G|}\}$ is the set of geographic interval labels, $|G|$ denotes the number of interval labels in the set. In our model, we divide the spatial distance between two POIs calculated by the corresponding latitude and longitude into geographic interval segments, and the result is used as the geographic interval label for users to access POIs.

Definition 5. *Check-in activity.* We define a user's check-in activity as a quintuple: $r_e^u = (t_k, l_e, g_e, c_e, s_e)$, $r_e^u \in o^{t_k}(u)$, representing that user u visit a POI l_e in the time period t_k , where the category of POI l_e is c_e , s_e denotes the counted check-in frequency of user visiting POI l_e . While g_e indicates the geographic interval label between POI l_e and the POI before user u visits POI l_e .

Problem 1. *Next POI recommendation.* Given a specific user's check-in sequence $seq(u_i) = \{o^{t_1}, o^{t_2}, \dots, o^{t_T}\}$, where $o^{t_i} = \{r_1, r_2, \dots, r_m\}$, its task is to recommend most likely POIs that user u_i will visit during the current time period t_i .

3.2. Preliminary Analysis

In order to accurately simulate users' visiting preferences, we conduct analysis on two public LBSN datasets (e.g., Foursquare and Gowalla) from the following two aspects: changes in users' visiting preferences over time/space, and the impact of POI categories on users' selection of next POI.

Check-in frequency represents the number of times a user visits the same category of POIs in each time period, and we use this as an indicator of changes in users' visiting preferences. Through the analysis of users check-in data from Foursquare, it is found that in different time periods and different geographical distances, users' check-in frequencies for the same category of POIs are different, indicating that users' visiting preferences are constantly changing with time and geographical distance (Gowalla dataset also presents a similar rule). For example, Figure 1 shows the check-in frequency for three categories of POIs of a user (ID:844) in Foursquare from 1 April 2012 to 31 September 2013 in different time and space periods. As can be seen from Figure 1, there are significant differences in check-in frequency of the user for POIs from three categories within different spatio-temporal intervals.

More specifically, as can be seen from Figure 1a, check-in frequency of the user to POIs of the same category is different in different time periods. The reasons may be: (1) user's preferences for POIs of different categories (such as entertainment or restaurants) may change with time; (2) user's periodic preferences may be influenced by space/time, money, or other external reasons at the time. Therefore, considering dynamic preferences of users in the current time period is conducive to improve the recommendation performance. In addition, it can be seen from Figure 1b that users have different check-in frequencies for the same category of POIs at different geographical intervals. We consider that if a user visits a POI frequently at a close geographical distance, it is difficult to judge whether the user really likes the POI. However, if a user frequently visits a POI with a long geographical distance, it can be assumed that the user likes the POI more. Therefore, we believe that user preferences will constantly change over time and geographical intervals, and this change will affect user's choice of next POI.

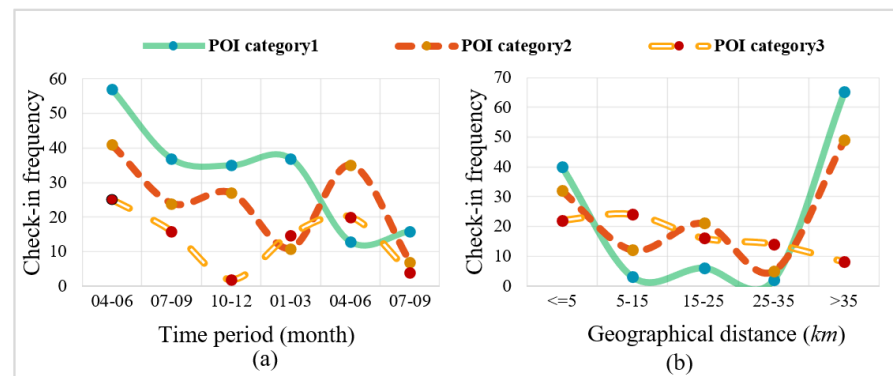


Figure 1. Illustration of dynamic changes in user (ID: 844) visiting preferences over spatio-temporal intervals on Foursquare.

In addition, in order to illustrate that POI category information has an impact on the recommendation performance, we conducted a statistical analysis on the number of POI categories visited by users from Foursquare and Gowalla, as shown in Figure 2. From Figure 2a, it can be seen that the cumulative number of POI categories visited by more than half of users in historical check-in sequences is concentrated in 4–8. While in Gowalla dataset (Figure 2b), total POI categories visited by users is concentrated in 20–40. It can be seen that the number of user preferences for POI categories is limited. These results demonstrate that POI categories help capture users' visiting preferences, which motivate us to use POI category filter in our model, so as to further improve the recommendation performance.

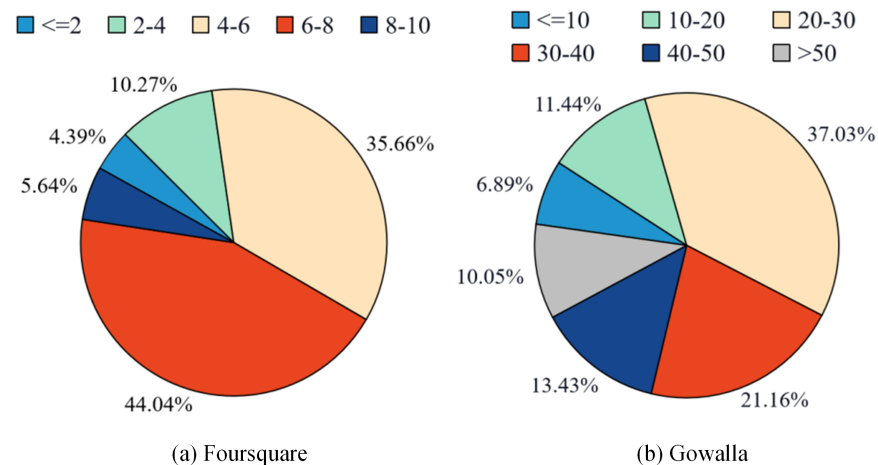


Figure 2. Statistics on the number of POI categories visited by users from Foursquare and Gowalla.

4. Proposed Method

In this section, we elaborate the overall framework of our ML-POIRec model. As shown in Figure 3, our model mainly includes the following four parts:

(1) Multi-dimensional check-in information embedding layer. It is used to obtain a dense representation of a user check-in sequence information, including POI ID, POI category information, time interval information, spatial interval information and user check-in frequency.

(2) Multi-dimensional static preference LSTM module. In order to obtain user's multi-dimensional long term static preference $p_s^{(t)}$, this module is leveraged to update LSTM model parameters according to the information of user's interaction with POIs in the past time period.

(3) Multi-dimensional dynamic preference meta-learning module. It aims to capture user's multi-dimensional short term dynamic preference $p_d^{(t)}$ by considering the information of user's interaction with POIs in the current time period. Training the meta-learning module with multiple users (tasks) enables the model to have strong generalization ability, thus making it applicable for testing new users.

(4) Next POI Recommendation. The obtained user's multi-dimensional static preference representation and dynamic preference representation are integrated to obtain the comprehensive preference representation of the user, which is then matched with the POI embedding after category filtering in the candidate pool, the preference score of next POI is calculated, and POIs with top- n scores are recommended.

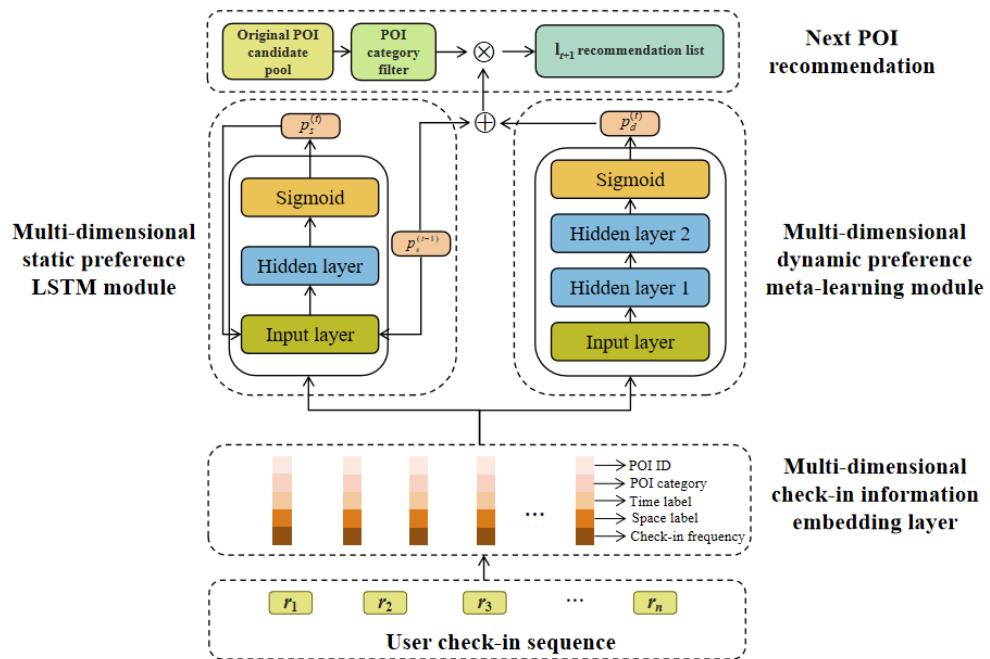


Figure 3. Illustration of ML-POIRec model.

4.1. Multi-Dimensional Check-In Information Embedding Representation

The multi-dimensional check-in information embedding layer is leveraged to encode the multi-dimensional auxiliary information of user and POI interactions into a latent representation. According to the analysis in Section 3.2, the check-in time of a user visiting a POI, the geographical distance between two visited POIs, the check-in frequency of POIs, and POI categories all have an impact on user's selection of next POI. Therefore, the multi-dimensional auxiliary information such as POI ID information, POI category information, spatio-temporal label of the user visiting POIs, and user check-in frequency are taken as inputs, and the embedding vectors are obtained according to the inputs and then concatenated.

To take full advantage of the multi-dimensional auxiliary information, we use a similar approach in [25] to generate the initial representation of POI using the embedding matrix $\mathbf{E}^L \in \mathbb{R}^{d \times m}$, where d is the latent dimension and m is the dimension of POI feature. First, POI l_i is encoded as a binary vector $x_{l_i} \in \mathbb{R}^m$, and then $\mathbf{E}^L$ is used to transform to obtain $a_{l_i} = \mathbf{E}^L x_{l_i}$. Thus, the obtained embedding vectors of POIs can be stacked as:

$$I = [a_{l_1}, a_{l_2}, \dots, a_{l_{|L|}}] \quad (1)$$

where L is the total number of POIs from a user check-in sequence. After learning user's latent representation, $\mathbf{E}^L$ will be optimized as the model is trained, as described in Section 4.5.

4.2. Multi-Dimensional Static Preference Modeling for Users

The historical check-in information of users is particularly important for modeling their long term stable (static) preferences. By capturing users' multi-dimensional static preferences and combining them with their multi-dimensional dynamic preferences, it is helpful to more accurately simulate users' comprehensive preferences. Considering LSTM is very effective for processing sequential data with rich contextual information, and can alleviate the gradient vanishing problem of RNN. Therefore, we use LSTM network to encode user's historical check-in information to simulate user's long term stable latent preference representation $p_s^{(t)}$, defined as:

$$p_s^{(t)} = \zeta_{LSTM}^t(p_s^{(t-1)}, o_u^t; \omega_u) \quad (2)$$

where $\zeta_{LSTM}^t(\cdot)$ represents the LSTM model, $p_s^{(t-1)}$ denotes user's multi-dimensional static latent preference factor in the previous time period, o_u^t represents the set of check-in activities of user u in the time period t , and ω_u is the network parameter. We update ω_u using the stochastic gradient descent method, shown as:

$$\omega_u \leftarrow \omega_u - \eta \nabla_{\omega_u} (\mathbf{L}_{T_u^t}(f_{\theta_u^t, \omega_u}) + \frac{\lambda}{2} \|\omega_u\|_2^2) \quad (3)$$

where η is the step size hyperparameter and λ denotes the regularization parameter. For T_u^t and θ_u^t , see Section 4.3.

4.3. Multi-Dimensional Dynamic Preference Modeling for Users

The main task of user's multi-dimensional dynamic preference meta-learning module is to capture user's multi-dimensional dynamic preference by considering the interaction information between users and POIs in the current time period. Here, we adopt meta-learning because it is a popular few-shot learning method and can alleviate the cold start problem, as described in Section 2.2. However, different from existing meta-learning based POI recommendations [26,27], our model takes input from user's check-in activity in the current time period. In this way, it can capture user's potential preferences for visiting POIs in the current time period, so as to provide more accurate next POI recommendation. We consider each user as a learning task to learn a meta-learning parameter θ^t that represents the global user model parameter within time period t in the meta-training users set including users with enough check-in activities. We follow settings of few-shot learning [32], where tasks' distribution is represented as $P(T)$. The model is iteratively trained by sampling task T_u^t from $P(T)$, and user's multi-dimensional dynamic latent preference within time period t generated by this module is shown as:

$$p_d^{(t)} = \zeta_{ML}^t(T_u^t, o_u^t, \theta^t) \quad (4)$$

where $\zeta_{ML}^t(\cdot)$ represents the meta-learning model, and T_u^t indicates the task of user u in time period t , that is, the check-in activity set o_u^t of the user at current time period t . While each task includes support set S_u^t and query set Q_u^t , where S_u^t includes the first k check-in activities of a user in current time period t , and Q_u^t includes the remaining check-in activities of a user in current time period t .

Specifically, we adopt optimization-based meta-learning methods to learn users' multi-dimensional dynamic preferences. The input to the meta-learning module is the embeddings of POIs visited by the user during current time period t . The training process of the meta-learning module includes both inner and outer loops as described below.

Inner loop: we pass support set S_u^t of a task T_u^t to $\zeta_{ML}^t(\cdot)$ and adjust the model parameter θ_u^t of user u based on the global user model parameter θ^t to perform local update, as shown in Formula (5).

$$\theta_u^t = \theta^t - \alpha \nabla_{\theta^t} \mathbf{L}_{T_u^t}(f_{\theta^t, \omega_u}) \quad (5)$$

The update is achieved by one or more gradients with respect to the global parameter θ^t . Where α is the step size hyperparameter, $\nabla_{\theta^t} \mathbf{L}_{T_u^t}(f_{\theta^t, \omega_u})$ represents the gradient error with respect to the initial parameter θ^t . While $\mathbf{L}_{T_u^t}(f_{\theta^t, \omega_u})$ is the loss function of model propagation sample, which is calculated based on support set S_u^t . Our ML-POIRec model mainly calculates such loss using mean square error (MSE), defined as:

$$\mathbf{L}_{T_u^t:P(T)}(f_{\theta^t, \omega}) = \sum_{l_i \in L} \|sco_{(u, l_i)} - fre_{(u, l_i)}\|_2^2 \quad (6)$$

where θ_u^t and ω represent the learnable model parameters, and $sco_{(u, l_i)}$ denotes the predicted score of user u for POI l_i , which will be described in Section 4.4. The initial $fre_{(u, l_i)}$ represents check-in frequencies of users visiting POI l_i , indicating the degree of users preferences for visiting that POI.

After performing batch-size T_u^t to train $\zeta_{ML}^t(\cdot)$, the parameter of LSTM module ω_u is updated through user multi-dimensional static preference module based on the historical check-in activity set $o_u^{1:t-1}$ of batch-size users before the time period t . It is trained iteratively according to different tasks of different users. From here on, the inner loop ends.

Outer loop: only after the completion of the inner loop, we can continue to update the initial parameter θ^t of the original meta-learning model based on feedback from all tasks T_u^t . That is, we need to obtain the total loss function for all tasks and perform a second gradient descent on the meta-learning model. We use query set Q_u^t of each T_u^t in batch-size to test the effect of the meta-learning model with parameter θ_u^t , and calculate the total loss, defined as:

$$\mathbf{L}_{T_u^t}(f_{\theta_u^t, \omega_u}) = \sum_{i=1}^{batchsize} loss_i(\theta_u^t, \omega_u) \quad (7)$$

The total loss function is expressed as the sum of loss $loss_i(\theta_u^t, \omega_u)$ in the meta-learning model on the corresponding parameters of the query set of each task in a batch size. Therefore, the global update is to update the parameter of meta-learning module at the current time period according to query set Q_u^t in the inner loop, and the weight update of the global model parameter is defined as:

$$\theta^t \leftarrow \theta^t - \beta \nabla_{\theta^t} \sum_{T_u^t:p(T)} \mathbf{L}_{T_u^t}(f_{\theta_u^t, \omega_u}) \quad (8)$$

where β is the step size hyperparameter. In this way, continuous inner and outer loop training can obtain model parameters with good generalization ability on the dataset, thereby obtaining global knowledge shared by users, so as to accurately simulate dynamic visiting preferences of cold start users with only a small number of check-in activities in the current time period in the meta-test users set.

We provide the pseudo code for the model training process in Algorithm 1. Algorithm 1 summarizes the process of updating user multi-dimensional dynamic preference model parameter through meta-learning and updating user multi-dimensional static preference model parameter through LSTM. Among them, lines 3–8 represent the process of locally updating the meta-learning module and updating LSTM model parameters through the support set in user's check-in activities in the current time period and check-in activities in the past time period, respectively. Line 9 represents the global update of the meta-learning module through the query set in user's check-in activities in the current time period.

4.4. Next POI Recommendation by Integrating User Static and Dynamic Preferences

Considering that user's preference for POI category will affect user's selection of next POI recommendation [33], therefore, before returning the recommendation result, we add a POI category filter to remove those POIs belonging to the POI category that has never been visited in user's check-in sequence in the POI candidate pool, so as to further improve the recommendation accuracy.

After filtering, we calculate the preference score for each POI by taking dot product between the captured user comprehensive preference and the filtered POIs using Formula (9) and recommend the user a top- n list of POIs with high scores.

$$sco_{(u,l_i)} = (p_s^{(t)} + p_d^{(t)}) \cdot I_{l_i} \quad (9)$$

where user's comprehensive preference representation is the vector sum of user's multi-dimensional static preference $p_s^{(t)}$ in the past time period and user's multi-dimensional dynamic preference $p_d^{(t)}$ from the current time period. I_{l_i} represents the embedding of POI l_i filtered by the POI category filter.

Algorithm 1 The training process of the model

Input: Meta training task set $P(T)$, initialization parameter θ^t

Output: The optimal parameter θ^t of the model

```

1: while not converge do
2:   Task sampling  $T_u^t \sim P(T)$ 
3:   for all  $T_u^t$  do
4:     Calculate the gradient error  $\nabla_{\theta^t} L_{T_u^t}(f_{\theta_u^t, \omega_u})$  of the initial parameter  $\theta^t$ 
5:     Sampling support set  $S_u^t$  for local updates
6:     Calculate the descent gradient to update model parameter  $\theta_u^t$  through Formula (6)
7:     Use Formula (3) to update user's multi-dimensional static preference module
       parameter based on the check-in activity set  $o_u^{1:t-1}$  in the past time period
8:   end for
9:   Use Formula (8) to perform meta updates on the multi-dimensional dynamic prefer-
       ence module based on the query set  $Q_u^t$ 
10: end while

```

4.5. Model Optimization

This subsection is to optimize the model we proposed. Based on user's historical check-in sequence $seq(u) = \{o^{t_1}, o^{t_2}, \dots, o^{t_T}\}$, after learning the latent preferences of the user multi-dimensional static and dynamic representations $p_s^{(t)}$ and $p_d^{(t)}$, the matrix E^L is optimized with the process of model training.

We represent L_{emd} as a differentiable loss function for training the embedding matrix E^L , represented as:

$$L_{emd} = - \sum_{i \in L} \sum_{j \in k} [x_{l_i}]_j \log [x'_{l_i}]_j \quad (10)$$

where x_{l_i} is the original embedding representation of POI l_i , k represents the number of meta-learning samples, and x'_{l_i} is the embedded representation of the POI after updating, which is obtained by:

$$\begin{aligned} m_{l_i} &= \zeta(E^L \cdot x_{l_i}) \\ [x'_{l_i}] &= \text{sigmoid}(\psi^T \cdot m_{l_i}) \end{aligned} \quad (11)$$

where $\zeta(\cdot)$ represents the decoding model and m_{l_i} represents the intermediate representation after POI l_i decoding, which is converted by the sigmoid function of attribute-wise. While ψ^T is the parameter of the sigmoid transformation.

In order to jointly train the POI embedding matrix, user multi-dimensional dynamic preference meta-learning module, and user multi-dimensional static preference LSTM module, we optimize the total loss function defined as:

$$\arg \min_{\theta^t, \omega_u, E^L} \sum_{T_u^t: P(T)} L_{T_u^t}(f_{\theta_u^t, \omega_u, E^L}) + \tau L_{emd} + \frac{\lambda}{2} (\|\theta^t\|_2^2 + \|\omega_u\|_2^2) \quad (12)$$

where τ represents the weight and λ is the regularization parameter.

5. Experiments

5.1. Datasets and Preprocessing

We evaluate our ML-POIRec model on two real-world LBSNs datasets [34], i.e., Foursquare and Gowalla. The Foursquare dataset contains check-in records of users collected from April 2012 to September 2013, and Gowalla contains users' check-ins extracted from February 2009 to October 2010. Due to the fact that neither dataset (Foursquare and Gowalla) has been processed and already includes many cold start users with less than 5 check-in activities, so both datasets can be considered as cold start datasets (The density of user-POI interaction matrix is 0.13% and 0.22%, respectively).

Next, we sorted check-in activities of each user in both datasets based on the corresponding timestamps of check-in activities. For the check-in time information of users, we divide each dataset into different check-in time periods based on timestamps of users' visit POIs. As a result, Foursquare dataset is divided into six check-in time periods, and each check-in time period is three months. While Gowalla dataset is divided into ten check-in time periods, each of which is two months.

Similarly, we divide the geographic information of user's visit POIs by geographic distance. Specifically, we use *Haversine()* to calculate the spatial (geographic) distance between two POIs according to the corresponding latitude and longitude. The geographic information in user's current check-in activity is the geographic distance between the POI visited in the current check-in activity and the previous POI visited (the geographic information in user's first check-in activity is zero). Furthermore, we classify the spatial distance into different geographic segments, each with a 10 km interval, as the geographic interval label for users to visit the POI.

We follow the setting [25] of meta-learning model and dynamically divide users into meta-training users sets and meta-test users sets. Specifically, we consider users with fewer than five check-in activities in the current time period as meta-test users, and the remaining users as meta-training users. We consider the distribution of task $P(T)$, and each user is represented as a few-shot regression task T_u^t sampled from a given task distribution. Each task includes a support set S_u^t and a query set Q_u^t . Here, S_u^t includes user's first k check-in activities in time period t , where k is represented as the number of meta-learning samples. While user's remaining check-in activities within the time period t is considered as query set.

For the non-meta-learning comparison methods, we first collect user's check-in activities in the current time period, and then obtain the first k check-in activities in current time period as training set, and select the remaining check-in activities in current time period as test set, so as to simulate the few-shot problem of users visiting POIs in the current time period.

5.2. Baselines

We adopt the following baselines to demonstrate the effectiveness of our ML-POIRec model.

FM [35]: a standard model of factor factorization, which considers second-order interactions between users and items.

NeuMF [36]: a state-of-the-art approach for item recommendation which combines traditional MF and MLP in neural networks to predict user-item interactions.

GRURec [37]: it is the first to use RNN in recommendation systems, optimizing GRU units on the basis of traditional RNN, which can effectively alleviate the long-standing problem of traditional RNN in obtaining users' visiting preferences.

Caser [38]: a sequential recommendation approach with convolutional sequence embedding to capture preferences and sequential patterns by modeling recommendation as a uniform and flexible structure.

SASRec [39]: a Transformer-based sequential recommendation model that captures users' long term visiting preferences. At each moment, the model aggregates information related to the current check-in activity and uses this information for next-step prediction.

MeLU [25]: a recommendation method based on meta-learning is to estimate preferences of a new user according to a small number of user-item interaction records. In addition, a candidate item selection strategy is provided for personalized preference estimation.

MAMO [26]: in order to solve the problem of generating simple initialization vectors of users or POIs based on MAML methods, and quickly convergence through a small amount of training to obtain a good global knowledge, resulting in local convergence of some users and poor generalization performance, MAMO introduces several additional memory modules to improve the generalization ability of the model.

These methods can be divided into four categories: one is Matrix Factorization based Approach (MFbA), such as FM, NeuMF; the second is Neural Network based Approach (NNbA), including GRU-based method (GRURec) and CNN-based method (Caser); the third is Self-Attention Network based Approach (SANbA), e.g., SASREC; the fourth is Meta-Learning based Approach (MLbA), such as MeLU, MAMO and ML-POIRec.

5.3. Parameter Settings and Evaluation Metrics

We use Torch 1.2.0 as a machine learning framework and NumPy 1.19.5 for the experiments. In our proposed model, we conducted grid search for all parameters and selected the best values of the corresponding parameters to make the model perform best. Specially, the first and second hidden layers in the meta-learning module have 128 and 64 hidden units with ReLU activation function, respectively, and the size of both the input layer and the output layer is 128. We set the number of samples for meta-learning $k = 5$ as a limited number of user check-in activities for the support set. The latent dimension d for both datasets is set to 128. We optimize the model through Adam optimizer [40] using L2 regularization, and set the regularization parameter to 0.0001. The learning rate is 0.001 and dropout is 0.2 to avoid overfitting. The epoch is 50 and batch-size is 128. The source code of our proposed model is publicly available at <https://github.com/glpppp/ML-POIRec> (accessed on 24 August 2023).

To evaluate the performance of ML-POIRec, we adopt two widely used evaluation metrics [41]: $NDCG@n$ and $Recall@n$, where n is the number of recommended POIs. $NDCG@n$ evaluates the quality of ranking, which indicates that whether POIs that users actually check-in rank at the top of next POI recommendation list. While $Recall@n$ measures the presence of POIs that users actually visit among the top- n recommended POIs. The higher the $NDCG@n$ and $Recall@n$, the better the model performance. $NDCG@n$ and $Recall@n$ are defined as follows.

$$NDCG@n = \frac{1}{|U|} \sum_{u \in U} \frac{DCG_u@n}{IDCG_u@n} \quad (13)$$

$$DCG_u@n = \sum_{i=1}^n \frac{2^{r_i} - 1}{\log_2(i+1)} \quad (14)$$

where r_i denotes the graded correlation of POI ranked i . The value of r_i can be obtained by binary relevance, i.e., $r_i = 1$ if a POI actually visited by the user is in the POI recommendation list, and 0 otherwise. $IDCG_u@n$ indicates the value of $DCG_u@n$ at ideal ranking.

$$Recall@n = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i=1}^n r_i}{|V_u|} \quad (15)$$

where $|V_u|$ represents positive POIs that user u has visited. Note that $|V_u| = 1$ in our work.

5.4. Sensitive Analysis of Parameters

In this section, we investigate whether the performance of ML-POIRec is sensitive to the latent dimension d , value of epochs, and number of meta-learning samples k . We adopt $NDCG@n$ to analyze the effects of these parameters. Note that the results on $Recall@n$ are similar to that on $NDCG@n$.

5.4.1. Effect of Latent Dimension

In order to study how sensitive ML-POIRec is to latent dimension d , we conduct the experiments while keeping other hyperparameters unchanged. Figure 4 shows the results. It can be seen that the performance first increases and then decreases slowly as d increases. The reason is that d denotes the model complexity, and a large d is too complicated to describe the datasets, while a small d is not enough to model the datasets. Therefore, the optimal value of d is 128 on both datasets.

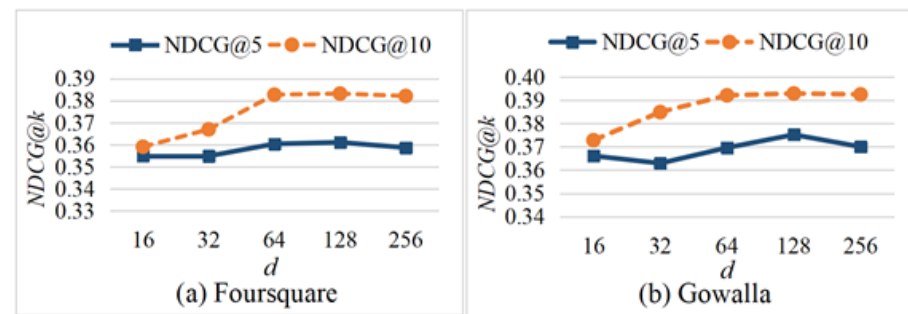


Figure 4. Effect of different latent dimension d on model performance.

5.4.2. Effect of Epochs

In general, in neural networks, we need to transfer all datasets in the same neural network for several times to train the model. Too many times of epochs will easily result in overfitting, while too few times will easily make the training parameters fail to reach the optimal. Based on this, we take model performance convergence as the basis for setting epochs. As shown in Figure 5, the model performance first increases as epochs increase, then grows slowly and gradually becomes stable roughly after 40–50 epochs in two datasets. Therefore, we set epochs to 50.

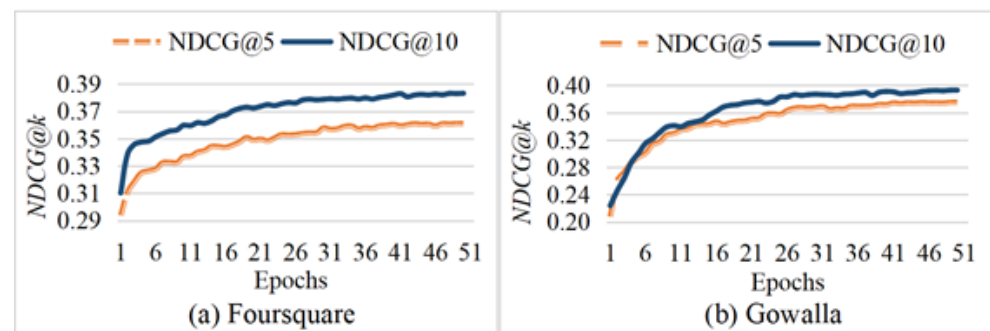


Figure 5. Effect of epochs on model performance.

5.4.3. Effect of the Number of Meta-Learning Samples

In this section, we demonstrate the effect of the number of samples k on the model performance. The meta-learning model with too small number of samples is not enough to simulate user's comprehensive preference. While the number of samples is too large, it does not conform to the concept of few-shot learning. Therefore, we turned k from {1, 2, 3, 4, 5}. In order to illustrate the effect of k and the effectiveness of ML-POIRec, we compare two methods (MELU and MAMO) based on meta-learning. Figure 6 shows the results. We can observe that the performance of three models improves with the increase of k , and the performance of ML-POIRec is the best. This is because the more check-in activities included in the support set for each task, the more it reflects the known visiting behavior of the target cold start user, which in turn can more accurately simulate user's preferences, so as to achieve more improvements. Therefore, we set k to 5.

Next, we will use the selected parameter values for subsequent experiments.

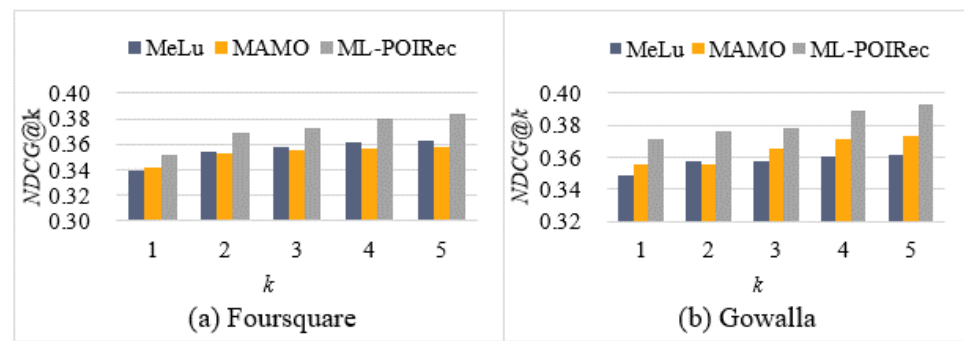


Figure 6. Effect of different meta-learning samples k on model performance.

5.5. Recommendation Performance

The performance comparisons of eight methods on two datasets are shown in Tables 2 and 3, in the form of mean and standard deviation of $NDCG@n$ and $Recall@n$. The numbers shown in bold in Tables 2 and 3 represent the best performance of each column in the corresponding tables. From the experimental results shown in Tables 2 and 3, we can make the following observations.

Table 2. Performance comparison on Foursquare dataset (mean $\pm$ std).

Category	Methods	NDCG@5	NDCG@10	Recall@5	Recall@10
MFbA	FM	0.2272 $\pm$ 0.0109	0.2487 $\pm$ 0.0123	0.2556 $\pm$ 0.0087	0.2950 $\pm$ 0.0076
	NeuMF	0.2341 $\pm$ 0.0065	0.2620 $\pm$ 0.0082	0.2887 $\pm$ 0.0061	0.3496 $\pm$ 0.0070
NNbA	GRURec	0.2822 $\pm$ 0.0037	0.3184 $\pm$ 0.0065	0.3107 $\pm$ 0.0050	0.3648 $\pm$ 0.0056
	Caser	0.2910 $\pm$ 0.0109	0.3261 $\pm$ 0.0127	0.3409 $\pm$ 0.0081	0.3836 $\pm$ 0.0079
SANbA	SASRec	0.3214 $\pm$ 0.0092	0.3348 $\pm$ 0.0096	0.3588 $\pm$ 0.0075	0.3958 $\pm$ 0.0064
MLbA	MeLU	0.3230 $\pm$ 0.0012	0.3623 $\pm$ 0.0037	0.3824 $\pm$ 0.0040	0.4118 $\pm$ 0.0044
	MAMO	0.3397 $\pm$ 0.0078	0.3579 $\pm$ 0.0066	0.3653 $\pm$ 0.0069	0.3924 $\pm$ 0.0034
	ML-POIRec	0.3611 $\pm$ 0.0090	0.3832 $\pm$ 0.0065	0.3918 $\pm$ 0.0101	0.4309 $\pm$ 0.0072

Table 3. Performance comparison on Gowalla dataset (mean $\pm$ std).

Category	Methods	NDCG@5	NDCG@10	Recall@5	Recall@10
MFbA	FM	0.2652 $\pm$ 0.0108	0.2945 $\pm$ 0.0111	0.2945 $\pm$ 0.0092	0.3225 $\pm$ 0.0103
	NeuMF	0.3113 $\pm$ 0.0099	0.3277 $\pm$ 0.0082	0.3314 $\pm$ 0.0076	0.3613 $\pm$ 0.0063
NNbA	GRURec	0.3415 $\pm$ 0.0077	0.3568 $\pm$ 0.0049	0.3611 $\pm$ 0.0060	0.3989 $\pm$ 0.0072
	Caser	0.3576 $\pm$ 0.0102	0.3779 $\pm$ 0.0097	0.3863 $\pm$ 0.0049	0.4250 $\pm$ 0.0071
SANbA	SASRec	0.3568 $\pm$ 0.0033	0.3762 $\pm$ 0.0071	0.3786 $\pm$ 0.0086	0.4232 $\pm$ 0.0088
MLbA	MeLU	0.3431 $\pm$ 0.0066	0.3617 $\pm$ 0.0080	0.3774 $\pm$ 0.0088	0.4184 $\pm$ 0.0045
	MAMO	0.3562 $\pm$ 0.0111	0.3730 $\pm$ 0.0100	0.3822 $\pm$ 0.0084	0.4233 $\pm$ 0.0070
	ML-POIRec	0.3752 $\pm$ 0.0086	0.3929 $\pm$ 0.0037	0.4001 $\pm$ 0.0058	0.4527 $\pm$ 0.0055

First, compared with the other seven models, the performance of ML-POIRec model has improved significantly on both two datasets. Taking FourSquare dataset as an example, compared with the best baseline, ML-POIRec improves $NDCG@10$ and $Recall@10$ by 5.77% and 4.64%, respectively. The performance improvements on Gowalla are also great. These results clearly demonstrate the effectiveness of ML-POIRec.

Second, in most cases, deep learning-based methods (GRU4REC, Caser and SASREC) perform better than traditional matrix factorization methods (FM and NFM). The reason is that those deep learning methods simulate user's preferences through training a large

amount of data, which can more accurately simulate the trend of user preferences over time, space and other factors, so as to more accurately predict user's selection of next POI.

Third, when considering the impact of user check-in sequences, the attention network-based method (SASRec) performs better than RNN based method (GRU4Rec). The reason is that attention network-based methods can effectively alleviate the defects of RNN models, which can effectively simulate users' long term preferences, and perform parallel operations to improve model efficiency.

In addition, when faced with cold start data, the performance of deep learning based models appears to be quite limited. This is because deep learning methods require sufficient training data to simulate the comprehensive preferences of users as much as possible. Therefore, the performance in few-shot recommendation tasks has been greatly affected. Among them, SASRec model depends on the self-attention network and can obtain the long term dependence of users in visiting POIs. However, when users have fewer or no check-in activities in the current time period, SASRec does not perform well for simulating the short term preferences of users. While meta-learning based methods (MeLU, MAMO) can alleviate the problem of incomplete user preference acquisition caused by sparse user check-in activities by utilizing the shared knowledge among users in new user recommendation tasks. Therefore, the methods based on meta-learning can mine POI visiting preferences of individual users in limited data, so as to make next POI recommendation.

Finally, the performance of ML-POIRec is better than MeLU and MAMO on both datasets, mainly because it not only considers the sequential effects between user check-in activities, but also considers some auxiliary information such as geographic distance between two POIs in user check-in activities, check-in frequency, and POI category. Furthermore, in order to comprehensively simulate users' visiting preferences, we divide users' historical check-in activities according to time periods. Moreover, we conduct multiple iterative training through the meta-learning model to obtain the global knowledge of users' check-in activities in the current time period, so that it can be applied to those recent check-in cold start users, and capture users' long term static preferences in the past time period by combining with LSTM network. In addition, we leverage a POI category filter to filter POIs in the candidate pool to further improve the recommendation performance.

5.6. Ablation Study

In this section, we report the effectiveness of different components in ML-POIRec, including sequence impacts of user check-in sequence (ML-SE), user preferences at different time periods (ML-TP), geographic distance information (ML-GD) and POI category information (ML-POIRec). To this end, we conduct experiments on the following variants of ML-POIRec.

ML-SE: this model only considers the sequence influence, and does not distinguish users' visiting preferences in different time periods.

ML-TP: the model divides user's historical check-in sequence according to different time periods, and jointly considers user's past(long) check-in preference and current (short) check-in preference.

ML-GD: this model leverages the geographical distance between POIs in user check-in activities, and divides geographical intervals according to different geographical distances, as a geographical interval label for users visiting POIs.

Table 4 shows the characteristics of different ML-POIRec variants. We use $NDCG@n$ and $Recall@n$ to illustrate the effectiveness of different variants on two datasets. The results are presented in Tables 5 and 6.

Table 4. Characteristics of different ML-POIRec variants.

Variants	Sequence Impact	Time Periods Information	Geographic Distance Information	POI Category Filter
ML-SE	✓	×	×	×
ML-TP	✓	✓	×	×
ML-GD	✓	✓	✓	×
ML-POIRec	✓	✓	✓	✓

Table 5. Performance comparison of ML-POIRec variants on Foursquare.

Variants	NDCG@5	NDCG@10	Recall@5	Recall@10
ML-SE	0.3270	0.3517	0.3576	0.3860
ML-TP	0.3416	0.3634	0.3793	0.4197
ML-GD	0.3485	0.3764	0.3842	0.4231
ML-POIRec	0.3611	0.3832	0.3918	0.4309

Table 6. Performance comparison of ML-POIRec variants on Gowalla.

Variants	NDCG@5	NDCG@10	Recall@5	Recall@10
ML-SE	0.3467	0.3708	0.3789	0.4193
ML-TP	0.3551	0.3795	0.3883	0.4297
ML-GD	0.3653	0.3829	0.3921	0.4434
ML-POIRec	0.3752	0.3929	0.4001	0.4527

Take $NDCG@n$ as an example. From Tables 5 and 6, it can be seen that in the three variants of ML-POIRec, ML-SE has poor performance compared with ML-TP and ML-GD. This is because ML-SE only considers the influence of the sequence of user's check-in activities and cannot obtain users' visiting preferences in different time periods, so it may cause outdated recommendation.

While ML-GD achieves better performance than ML-TP because ML-GD can obtain users' geographical distance preferences for visiting POIs from their historical check-in sequences. It can assign larger weights to POIs with large distance interval and high user check-in frequency, so it is more reliable than ML-TP that only considers time information.

In contrast, the performance of complete ML-POIRec is superior to ML-SE, ML-TP and ML-GD. This is because when we simulate user preferences, we incorporate multi-dimensional information such as spatio-temporal interval information, user check-in frequency to POI, and POI categories. We comprehensively simulate users' long term static preferences and recent dynamic preferences, and add POI category filter to filter POIs in the candidate pool, so as to further improve the model performance.

5.7. Complexity Analysis

In this subsection, we analyze the approximate time complexity of the baseline approaches. Considering that it is difficult to directly analyze the time complexity of each approach, we estimate the approximate complexity of calculating user preference representation in each method to evaluate its efficiency.

Assume that all baseline approaches have the same sample size (denoted by n), the same hidden layer dimension (denoted by m) and the same check-in activities (denoted by l). For matrix factorization based approach, since FM method does not involve neural networks, its time complexity is $O(nl)$, while the time complexity of NeuFM is $O(nm^2l)$. For neural network based approach, the time complexity of GRURec is mainly related to the number of GRU layers and the size of hidden units in the GRU layer, its time complexity is $O(tnml)$, where t is the number of GRU layers. Caser used a convolutional neural network with a time complexity of $O(l + nl)$.

For the self-attention network based approach, SASREC used multi-head attention to capture sequence information, with a time complexity of $O(hn^2 + nl)$, where h is the number of attention heads. For meta-learning based approach, MeLU used multiple fully connected layers for feature extraction and combination, with a time complexity of $O(Lm^2 + nl)$, where L represents the model consisting of L fully connected layers with 2 layers. MAMO designed two memory matrices to provide personalized initialization for the recommendation model, with a time complexity of $O(2nml)$. Since our model includes two modules: static preference LSTM module and dynamic preference meta-learning module, with 1 and 2 hidden layers respectively, its time complexity is $O(nml + nm^2l)$.

6. Discussion

According to our statistics, user-POI interaction matrix in LBSN datasets is relatively sparse compared with user-item interaction matrix in the traditional e-commerce recommendation, and there are a large number of cold start users. Therefore, how to accurately model user preferences in the limited check-in records is an urgent problem to be solved. For recent check-in cold start users, we proposed the ML-POIRec model, which captures multi-dimensional static preferences and dynamic preferences of users by LSTM module and meta-learning module, and adds a POI category filter to further improve the accuracy of recommendation and effectively alleviate the cold start problem. It is of great significance to ensure the quality of next POI recommendation and provides a wide range of practical implications. More specifically, on the one hand, for social platforms and merchants, it is beneficial for LBSN service providers to recommend new POIs to users based on the captured long and short term visiting preferences, which can not only improve user experience, but also help them find more potential customers and bring potential benefits. On the other hand, for users, high-quality next POI recommendation can help users quickly find POIs that well meet their personalized needs, which is the key to attracting users and improving user satisfaction.

7. Conclusions and Future Work

In this paper, we propose a novel model, namely, ML-POIRec for next-POI recommendation. Specifically, we leverage multi-dimensional static LSTM module and dynamic meta-learning module to learn long term stable preferences and recent dynamic preferences of users. Furthermore, we incorporate a POI category filter into our model to filter candidate POIs, so as to more accurately simulate users' preferences. The experimental results on two public datasets demonstrate that ML-POIRec substantially improves the recommendation performance compared with the state-of-the-art methods.

In the future, we plan to adopt the method of base decomposition, and use the known user (POI) configuration file to generate the embedded representation of the unknown user (POI), thereby further improving the generalization ability of the model. Another future work is to attempt to utilize the recently hot self-supervised contrastive learning [42] for next POI recommendation, so as to further improve the recommendation performance.

Author Contributions: Zheng Li: Writing—review and editing, Supervision, Funding acquisition; Xueyuan Huang: Conceptualization, Methodology, Software, Writing—original draft, Writing—review and editing; Liupeng Gong: Validation, Software, Writing—review and editing; Ke Yuan: Methodology, Writing—review and editing; Chun Liu: Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The work is partially supported by the National Natural Science Foundation of China (No. 61402150, 61806074); Science and Technology Research Project of Henan Province (No.232102211029); Key Scientific Research Project Plan of Colleges and Universities in Henan Province (No. 23A520016).

Data Availability Statement: Publicly available datasets can be found here: https://github.com/YijunSu/LBSN_Dataset (accessed on 30 October 2022).

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

- 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. [\[CrossRef\]](#)
- Acharya, M.; Yadav, S.; Mohbey, K.K. How can we create a recommender system for tourism? A location centric spatial binning-based methodology using social networks. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100161. [\[CrossRef\]](#)
- Acharya, M.; Mohbey, K.K. Differential privacy-based social network detection over spatio-temporal proximity for secure POI recommendation. *SN Comput. Sci.* **2023**, *4*, 252. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Zou, Z.; He, X.; Zhu, A.X. An automatic annotation method for discovering semantic information of geographical locations from location-based social networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 487. [\[CrossRef\]](#)
- Wang, J.; Ding, K.; Caverlee, J. Sequential recommendation for cold-start users with meta transitional learning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021; pp. 1783–1787.
- Zhang, S.; Guo, J.; Liu, C.; Li, Z.; Li, R. Next point-of-interest recommendation for cold-start users with spatial-temporal meta-learning. In Proceedings of the 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 29–31 January 2023; pp. 80–87.
- Gupta, V.; Bedathur, S. Doing more with less: Overcoming data scarcity for poi recommendation via cross-region transfer. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–24. [\[CrossRef\]](#)
- Yao, Z.; Fu, Y.; Liu, B.; Liu, Y.; Xiong, H. POI recommendation: A temporal matching between POI popularity and user regularity. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 549–558.
- Si, Y.; Zhang, F.; Liu, W. A time-aware poi recommendation method exploiting user-based collaborative filtering and location popularity. In Proceedings of the 2017 2nd International Conference on Communications, Information Management and Network Security, Beijing, China, 22–23 October 2017; pp. 1–7.
- 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.
- 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 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 214–221.
- 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.* **2020**, *8*, 489–499. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Li, R.; Shen, Y.; Zhu, Y. Next point-of-interest recommendation with temporal and multi-level context attention. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 1110–1115.
- Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V.S.; Zhou, X. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 2512–2524. [\[CrossRef\]](#)
- Cui, Q.; Zhang, Y.; Wang, J. CANS-Net: Context-aware non-successive modeling network for next Point-of-Interest recommendation. *arXiv* **2021**, arXiv:2104.02262v1.
- Liu, C.; Liu, J.; Wang, J.; Xu, S.; Han, H.; Chen, Y. An attention-based spatiotemporal gated recurrent unit network for point-of-interest recommendation. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 355. [\[CrossRef\]](#)
- Ding, J.; Yu, G.; Li, Y.; Jin, D.; Gao, H. Learning from hometown and current city: Cross-city POI recommendation via interest drift and transfer learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 1–28. [\[CrossRef\]](#)
- Yin, H.; Zhou, X.; Cui, B.; Wang, H.; Zheng, K.; Nguyen, Q.V.H. Adapting to user interest drift for poi recommendation. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2566–2581. [\[CrossRef\]](#)
- Wang, X.J.; Liu, T.; Fan, W. TGVx: Dynamic personalized POI deep recommendation model. *INFORMS J. Comput.* **2023**, *35*, 711–908. [\[CrossRef\]](#)
- 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, Lyon, France, 23–27 April 2018; pp. 1459–1468.
- Bengio, S.; Bengio, Y.; Cloutier, J.; Gecsei, J. On the optimization of a synaptic learning rule. In Proceedings of the Preprints Conference Optimality in Artificial and Biological Neural Networks, Montreal, QC, Canada, 2002; pp. 1–29.
- Li, J.; Jing, M.; Lu, K.; Zhu, L.; Yang, Y.; Huang, Z. From zero-shot learning to cold-start recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 4189–4196.
- Lee, H.; Im, J.; Jang, S.; Cho, H.; Chung, S. Melu: Meta-learned user preference estimator for cold-start recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1073–1082.

26. Dong, M.; Yuan, F.; Yao, L.; Xu, X.; Zhu, L. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 688–697.
27. Pan, F.; Li, S.; Ao, X.; Tang, P.; He, Q. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 695–704.
28. Bharadhwaj, H. Meta-learning for user cold-start recommendation. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
29. Lu, Y.; Fang, Y.; Shi, C. Meta-learning on heterogeneous information networks for cold-start recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 1563–1573.
30. Chen, Y.; Wang, X.; Fan, M.; Huang, J.; Yang, S.; Zhu, W. Curriculum meta-learning for next POI recommendation. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2692–2702.
31. Tan, H.; Yao, D.; Huang, T.; Wang, B.; Jing, Q.; Bi, J. Meta-learning enhanced neural ODE for citywide next POI recommendation. In Proceedings of the 2021 22nd IEEE International Conference on Mobile Data Management (MDM), Toronto, ON, Canada, 15–18 June 2021; pp. 89–98.
32. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2017; pp. 1126–1135.
33. Lin, I.C.; Lu, Y.S.; Shih, W.Y.; Huang, J.L. Successive poi recommendation with category transition and temporal influence. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; pp. 57–62.
34. Su, Y.; Li, X.; Liu, B.; Zha, D.; Xiang, J.; Tang, W.; Gao, N. Fgrec: Fine-grained geographical characteristics modeling for point-of-interest recommendation. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), Virtual Event, 7–11 June 2020; pp. 1–6.
35. Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidt-Thieme, L. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 635–644.
36. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
37. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
38. Tang, J.; Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Virtual Event, 21–25 February 2018; pp. 565–573.
39. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
40. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
41. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
42. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 4116–4126.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.