2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA) | 978-1-6654-7278-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPECA56706.2023.10075890

Next Point-of-Interest Recommendation for Cold-start Users with Spatial-temporal Meta-Learning

Shaoqing Zhang
The School of Computer and Information Engineering
Henan University
Kaifeng, China
104753200835@henu.edu.cn

Jianzhong Guo
Henan Industrial Technology Academy of Spatio-Temporal Big Data
Henan University
Zhengzhou, China
10130151@vip.henu.edu.cn

Chun Liu
The School of Computer and Information Engineering
Henan University
Kaifeng, China
liuchun@henu.edu.cn

Zheng Li
The School of Computer and Information Engineering
Henan University
Kaifeng, China
lizheng@henu.edu.cn

Ruijing Li
The School of Computer and Information Engineering
Henan University
Kaifeng, China
104754201218@henu.edu.cn

***Abstract*—Next point-of-interest (POI) recommendation is to recommend the next POI that the users may want to visit according to their check-in trajectories. When there are few check-ins in the trajectories, it will cause the cold-start problem that traditional next POI recommendation methods can't capture users' preferences accurately and no longer has the ability to forecast the next POI. Based on the few-shot meta-learning method of MAML, this paper proposes a new method named STMeta to recommend the next POI for cold-start users. Following few-shot learning, the main idea of the proposed method is to train a model based on the long trajectories of other users with the aim to learn some transferrable and generalized knowledge which can be reused under the cold-start environment. For this purpose, it constructs a lot of short trajectories with only two POI check-ins next to each other from the long trajectories to simulate the cold-start environment, and uses them to train a MLP with the ability of measuring the spatial-temporal transition information between two POI check-ins. This information indicates the probability that the users visit the next POI from current POI. And by computing such transition information between users' current POI and all candidate POIs, the most nearest POI can be recommended to users. Based on the public datasets of Gowalla and Brightkite, our experiments have shown that the proposed method achieves better performance in recommending next POIs for cold-start users when compared with related methods.**

Keywords—Point-of-interest recommendation, Few-shot learning, Cold-start, Meta-learning

I. Introduction

With the emergence of location-based social networks(LBSN)[1] and the widespread use of smartphones, more and more people are engaging in social network interactions, providing a lot of data about point-of-interest (POI) check-ins reflecting when and where people have visited. In this case, the recommendation of next POI has gained much attention in recent years. It's main purpose is to recommend the POI that the target users want to visit based on their check-in history and current spatial-temporal context.

Many methods have been presented to provide users with the next POI recommendation. Traditionally, collaborative filtering(CF) methods are often used for next POI recommendation by utilizing the information of other users[2]. For example, Cai et al. proposed a framework named Friends-aware Graph Collaborative Filtering (FG-CF)[3], and Liu et al. proposed a POI group recommendation method[4]. With the wide use of deep learning in different domains, lots of neural networks based methods now are used to extract personal preference from users' trajectories. For example, He et al. proposed a time network model named TimeSAN to learn the correlation between the next POI visit and the historical visit of users through self-attention mechanism[5]. These methods are good at extracting users' spatial-temporal context information from their trajectories and have excellent performance when there are long check-in trajectories. However, in more practical circumstances, the great majority of new users of various platforms often engage with a few items and thus only few POI check-ins are there in most cases. This kind of long tail[6] phenomenon leads to that the platforms fail to deliver trustworthy recommendation services to new users. This problem is often called the user cold-start problem which is a challenge that every recommendation system must deal with.

To resolve the user cold-start problem of POI recommendation, there are two main kinds of methods at present. One kind of methods makes use of the available short sequences of POI check-ins of the cold-start users to find their most similar users, and then provides recommendation according to the preference of these similar users[7]. The other kind of methods draws support from some auxiliary data such as the personal information of gender, age and occupation, and the social network information of friends, likes, and concerns. Because the collection of these auxiliary data is concerned with users' privacy, it may make users disgusted and lead to that some fabricated information are provided.

Besides these traditional methods, researchers of POI recommendation start to address the user cold-start problem by taking it as a few-shot learning task in recent studies[8]. For example, Li et al. have proposed to use the metric learning based few-shot learning method to deal with the LBSN recommendation problem[9]. Following the meta-learning[10] which is another typical way of achieving few-shot learning, Sun et al. proposed a method based on user group preference optimizer and LSTM memory unit for next POI recommendation[11] Though grouping to find the similar check-in users, the proposed method generates the user preference by fusing the personal preference extracted from the users' trajectories and the group preference extracted from the trajectories of a group. Although this method provides a new way to solve the cold-start problem by utilizing the group preference of similar users, it is still difficult to provide good recommendation when none similar users are found around the check-in position of the target users.

Different from these works, this paper proposes a spatial-temporal meta-learning based method STMeta to address the user cold-start problem of next POI recommendation. Inspired by the work of Wang et al.[12] which applies the meta-learning to achieve sequence recommendation for cold-start users, the basic idea of this paper is to simulate the cold-start environment to train a recommendation model by following the meta-learning approach MAML. As an optimization based few-shot learning method, MAML aims to learn a better model parameter initialization which can quickly adapt to new tasks. For this purpose, it constructs many meta-tasks from available datasets which have a large number of labeled samples and uses these meta-tasks to train the model. Following MAML, we construct a lot of short trajectories from the users' long trajectories to simulate the settings of cold-start users, and take these short trajectories to train a recommendation model which can estimate the spatial-temporal transition information over two POIs. Once trained, the model can be further fine-tuned with the short trajectories of cold-start users. And the next POI can be recommended according to the spatial-temporal transition information between the last POI that the target user has visited and the candidate POIs. Based on two public data sets of Gowalla and Brightkite, the proposed method is compared with a group of related methods. Experimental results show that it can improve the recommendation performance by 11%.

The rest of paper is structured as follows. The second section mainly introduces the related researches on meta-learning, next POI recommendation, and cold-start recommendation. The third section presents the problem definition and the details of the proposed method. The fourth section is the experimental setup, baselines, the comparison with related methods, ablation experiments, and parameter sensitivity analysis. Finally, the conclusion is given in the last section.

II. Related Work

This section mainly introduces the related researches including meta-learning, next POI recommendation, and cold-start recommendation.

A. Meta-learning

Meta-learning aims to achieve the goal of ``learning to learn''. It can use past experience to guide the learning of new tasks, which is one of the commonly used methods to solve few-shot problem. The typical meta-learning method is the MAML proposed by Finn et al. whose purpose is to learn a model parameter initialization which can quickly adapt to new tasks[13]. It adopts a strategy of two-layer loops, i.e., inner-loop and outer-loop, to optimize the model. With one task consisting of a set of training samples and a set of testing samples, the inner-loop trains a copy of the model and then tests it to generate the task-level loss. With these task-level losses, the outer-loop trains the model to update the model parameters. Based on MAML, Sun et al. proposed a method named MTL further improved the feature extraction, task design, network structure, and other aspects of the method, drastically lowering the number of parameters that need to be modified in each gradient descent calculation[14].

B. Next POI Recommendation

At present, there are two commonly used methods to recommend the next POI. The first is to rely on other users to get more information for collaborative filtering(CF), and the second is to use neural network to extract personal preference features from users' trajectories. For example, Guo et al. proposed an Attentional Recurrent Neural Network (ARNN) [15]. Sun et al. proposed a novel method named Long-Term and Short-Term Preference Modeling (LSTPM) for next-POI recommendation[16]. Huang et al. thought that a user's travel was always related to a certain purpose. Thus, a POI recommendation model STPR based on purpose ranking was proposed to solve the personalized next POI recommendation problem[17]. Lian et al. proposed an optimized loss function of sampling and a location recommendation method of geographic encoder based on self-attention[18].

C. Cold-start Recommendation

The user cold-start problem in recommendation system is that when new users enter the system, the association between users and items cannot be established due to lack of necessary data, and the recommendation system cannot infer user preferences and provide recommendation. At present, there are two typical methods to solve this problem. The first one is collaborative filtering(CF) recommendation algorithm based on nearest neighbors. For example, Zhang et al. used a covering reduction algorithm to reduce the nearest neighbors of some cold-start users, and the reduced nearest neighbor scores were used to recommend new users[19]. The second is the model-based recommendation algorithm, including matrix decomposition[20], probability model[21], Boltzmann machine[22], etc. For example, Gao et al. used multi-level collaborative probability matrix decomposition to obtain user characteristics, used decision tree to train the relationship between users and tags, and achieved good results in cold-start recommendation[23].

III. Methodology

This section introduces the problem solved by the method proposed in this paper and the basic structure of this method.

A. Problem Statement

The goal of next POI recommendation is to recommend the next POI that the target users may want to visit based on their POI check-in histories and their current spatial-temporal context. This paper uses $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_\mathrm{G}\}$to represent a set of users, and $L = \{l_1, l_2, \cdots, l_P\}$ to represent a set of POIs where $l_k = \langle id_k, lon_k, lat_k \rangle$ represents a specific POI which contains the information of ID, longitude, and latitude. The check-in trajectory of a user over time is denoted as $S_i = \{p_1, p_2, \cdots, p_k\}$, where the POI check-in $p_j = \langle u_m, l_i, t_i \rangle$, p_j denotes the access of user u_m to the POI l_i at time t_i. Based on the above representation, the next POI recommendation problem is to recommend the $k + 1^{th}$ POI that the user will visit based on its check-in trajectoryS_i and the set of POIs L.

When there are long check-in trajectories S_i, the preference of the users can be learned effectively. However, there are only few POI check-ins in the trajectories S_i for many users, especially for the new users. This leads to the user cold-start problem in which it is difficult to make accurate recommendation with the short trajectories.

B. The Overview of the Proposed Method

The user cold-start problem of next POI recommendation is similar with that of few-shot learning. In few-shot learning, there are often few labeled samples, e.g., one sample or three samples, for each class. Then the problem is to predict the classes of lots of unlabeled samples according to these few labeled samples. The set of few labeled samples is called support set D_{supp}, and the set of unlabeled samples is called the query set D_{query}. Obviously, it is difficult to obtain enough knowledge from these few labeled samples to predict the classes of the unlabeled samples. In light of that, few-shot learning mainly resorts to some available large datasets called D_{base} which have lots of labeled samples to learn a generalized model which can be further fine-tuned using the samples in D_{supp}. To keep the training context same with that of testing, a lot of meta-tasks are constructed from the dataset D_{base} according to the data distribution of D_{supp} and D_{query}. That's, each task also has the support set D_{supp} and the query set D_{query}, and there are the same numbers of samples per class for the constructed support set and query set. With each task, the model is trained by the support set and tested by the query set, and updated subsequently according to the error generated by testing. In this way, the model will be trained with some transferable and generalizable prior knowledge. These knowledge can be reused under the few-shot environment. Compared with the constructed meta-tasks, the task predicting the classes of the samples in D_{query} with the help of D_{supp} is called the testing task. .

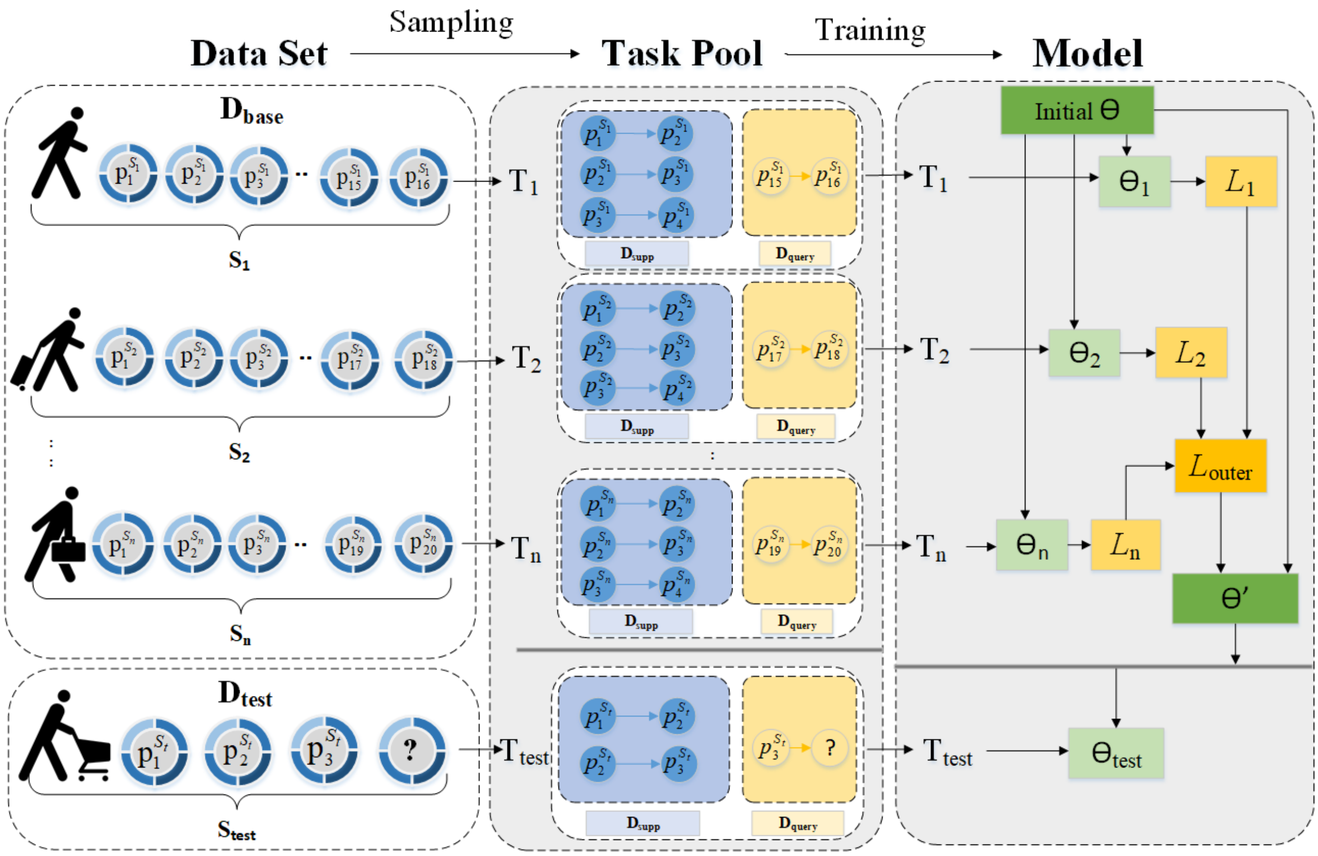

Fig.1 The framework of MAML-based method .

For the user cold-start problem of next POI recommendation, the task is to predict the next POI when there are few POI check-ins in the trajectories. Thus, as shown in Fig.1, the main idea of the few-shot learning based methods to address the user cold-start problem is to construct a lot of meta-tasks from the available long trajectories to train a generalizable model. Specially, for the query set of the meta-task, there is a trajectory with two POI check-ins. With the two POI check-ins trajectory, it is to test the trained model by inputting the first POI check-in of trajectory and predicting the next POI, while taking the second POI check-in of the trajectory as the target to calculate the accuracy. At the same time, the support set of the meta-task consists of a lot of trajectories with two POI check-ins to train the model.

Based on the constructed meta-tasks, different methods adopt different models and different ways to train the models for different purposes. For example, the metric-based methods use these constructed meta-tasks to train a generalizable metric which can measure the similarity or distance between the samples in the query set and the support set. Different from the metric-based methods, this paper follows the optimization based few-shot learning method MAML. MAML takes these constructed meta-tasks to train a model aiming at obtaining a better initialization of model parameters which can be reused under the few-shot environment. For this purpose, it adopts a two-layer optimization loops, i.e., the inner-loop and the outer-loop, to train the model. As can be seen in the right part of the Figure 1, it uses the samples in the D_{supp} of a meta-task T_1 to train the model, and the model parameters are updated from the initial θ to $θ_1$. Using the updated model, it further uses the samples in the D_{query} of T_1 to test the model, and then the loss $\mathcal{L}_1$ is generated. This is the inner-loop of the MAML optimization process based on each meta-task. Once all the meta-tasks have been executed, all the generated losses $\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_n$ are aggregated to update the initial model parameters θ to obtain a better initialization. This is the outer-loop of the MAML optimization process. The above two-layer optimization loops can be iterated several times. And each new iteration is based on the updated initial parameters generated by the outer-loop.

Following the MAML framework, we construct the meta-tasks from the long trajectories of each user through sampling as shown in the left part of Figure 1 and then adopt the two-layer optimization loops to train a model which can predict the POI that target user may visit according to the POI that target user interacts currently. To achieve this purpose, the idea of the proposed method trains the model with the ability of measuring the spatial-temporal transition information between two POI check-ins in a latent space, and then predict the next POI by measuring the transition information between user's current POI and all the candidate POIs and selecting the nearest POI for recommendation. Thus, as shown in Fig 2, the inputs of the model are the features of the two POI check-ins of one short trajectory in the support set of the meta-tasks. The outputs are the vectors indicating the spatial-temporal transition information between the two POI check-ins in the latent space. To measure the transition information between two POI check-ins, this paper implements the model by a Multilayer perceptron (MLP). We detail the construction of the meta-tasks from long trajectories and the training of the model as follows.

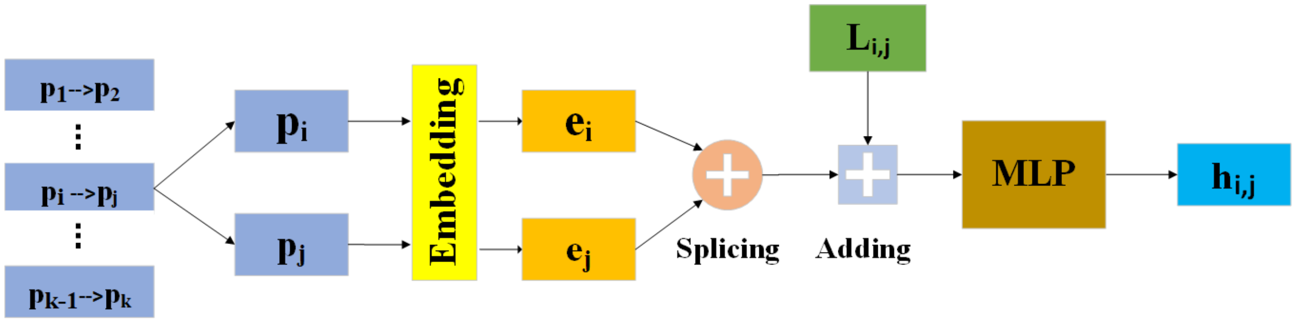

Fig.2 The forward learning process of the model proposed. The inputs are the short trajectories with two POI check-ins, and the outputs are the spatial-temporal transition information between the two check-ins in the short trajectories. And the $L_{i,j}$ represents the normalized spatial distance between the two check-ins.

C. The Construction of the Meta-task Pool

To construct the meta-tasks, we select randomly two POI check-ins from the long trajectories of each user. In detail, given the long trajectory S_i of user u_i, we first randomly select one POI check-in from the trajectory, and then its next POI check-in in the trajectory is also selected. In this way, a short trajectory with only two POI check-ins is obtained. By repeating the above process, we can get K (e.g., 3) short trajectories from the long trajectory of one user, which will constitute the support set of a meta-task. In the meanwhile, we select the last two POI check-ins of the long trajectory as the short trajectory constituting the query set of the meta-task.

From the perspective of model training, these selected short trajectories are only the positive samples. Thus, while constructing these positive short trajectories, we also construct the same number of negative short trajectories for the balance of the positive and negative samples. When selecting one POI check-in randomly from the long trajectory to construct the positive short trajectory, we also randomly choose another POI that the target user has not visited to build a fake next check-in point by setting a random check-in time. In this way, there are the same numbers of negative short trajectories as that of positive short trajectories in the support set and the query set of a constructed meta-task. Given the trajectories of all users, lots of meta-tasks will be constructed, which constitute the meta-task pool.

D. The Training Process

Based on these constructed meta-tasks, this paper follows the two-layer optimization loops shown in the right part of Fig.1, i.e., the inner-loop and the outer-loop, to train the model.

1) The inner-loop optimization: As shown in Figure 1, given one meta-task, the inner-loop optimization process executes the task by training the model with the support set D_{supp} and testing the model with the query set $D\_\{query\}$. Finally, a loss, for example, the $\mathcal{L}_1$ shown in Fig.1, will be generated. Specially, for one short trajectory $p_i \to p_j$ with two POI check-ins, the attributes of each POI check-in in the trajectory are embedded to obtain their features. Their features are further added together to generate the feature of the check-ins as shown in Eq. (1). Each check-in has the three attributes of the user ID u_m, the POI's ID l_i, and the check-in time t_i. We map the scalar values of these attributes into the features of e_{u_m}, e_{l_i}, and e_{t_i} with the same dimension by word embedding.

$$e_i = e_{u_m} + e_{l_i} + e_{t_i} (1)$$

Once the embedded features of the two POI check-ins of the short trajectory are obtained, we concatenate them together and input the concatenation into the MLP as shown in Fig. 2 to measure the spatial-temporal transition information $h_{i,j}$ between two points. In this process, the spatial distance between the two points as an important factor is also input into the model. The whole calculation process is defined in the Eq. (2) where W represents the trainable transformation matrix of MLP, b denotes the bias vector, $L_{i,j}$ is the spatial distance between p_i and p_j, and $\sigma(\cdot)$ is the Sigmoid activation.

$$h_{i,j} = \sigma(W(e_i || e_j + L_{i,j}) + b)(2)$$

When calculating the spatial distance $L_{i,j}$ between p_i and p_j, the Euclidean distance $d_{i,j}$ between p_i and p_j is first calculated and then normalized in the way shown in Eq. (3) where $mean_{poi}$ and SD_{poi} represents the mean and standard deviation of the Euclidean distance between all POIs.

$$L_{i,j} = \frac{1}{SD_{poi}}\left(d_{i,j} - mean_{poi}\right)(3)$$

Based on Eq.(2), for all the short trajectories in the support set D_{supp} of a meta-task, the positive spatial-temporal transition information $h_{i,j}$ between the points of the positive short trajectories and the negative spatial-temporal transition information $h_{i,j'}$ between the points of the negative short trajectories can be calculated. Then, the mean of the spatial-temporal transition information of a single meta-task can be calculated by following the Eq. (4), where $h_{i,j}$ represents the spatial-temporal transition information generated from the positive short trajectories in the support set D_{supp}, and K is the number of positive short trajectories.

$$\bar{h} = \frac{1}{K}\sum h_{i,j}(4)$$

With the mean spatial-temporal transition information $\bar{h}$, we expect that $e_i + \bar{h} = e_j$. That's, we can reach e_j from e_i when crossing over $\bar{h}$. In this way, we can obtain the loss by calculating the error $\left|\left|e_i + \bar{h} - e_j\right|\right|^2$ from each short trajectory. Then, while defining $score = \left|\left|e_i + \bar{h} - e_j\right|\right|^2$, the loss function of the inner-loop optimization is defined as Eq. (5) where $[\cdot]_+$ represents the function of $max(0,\cdot)$, and γ is a hyperparameter of $margin$.

$$\mathcal{L}_i = \sum\left[\gamma + score\left(h_{i,j}\right) - score\left(h_{i,j'}\right)\right]_+(5)$$

Once trained with the support set D_{supp} of a meta-task, the model parameter will be updated from initial θ to θ_i as shown in Fig.1. Then, the model will be further tested by the short trajectories in the query set D_{query} and the loss $\mathcal{L}_i$ will be generated according to the function of Eq. (5).

2) The outer-loop optimization: For each meta-task, the model will be trained with its support set and tested by its query set, and finally generate the loss $\mathcal{L}_i$. The outer-loop optimization process is to aggregate all the losses generated by the inner-loop optimization process, and to update the model parameters from the initial θ to a new one θ' by minimizing the aggregated losses, as shown in Eq. (6). In practice, the optimization process of inner-loop and outer-loop can be iterated several times and the new iteration will be based on the new updated model parameters.

$$\theta' = \min \sum L_i(6)$$

E. The Prediction of Next POI under Cold-start Environment

After training the model by the two-layer optimization loops, we can use the model to predict the next POI given the short trajectory of a cold-start user. When there is only one check-in point in the trajectory of the target user, we build the check-in point for each candidate POI by setting a same random visiting time and concatenate the check-in point with these built points to construct different trajectories. Then, for each trajectory $p_0 \rightarrow p_i$, we input it into the trained model to predict their spatial-temporal transition information h as that shown in Eq.(2), and rank all the candidate POIs according to the scores defined as follows, where e_0 and e_i are the features of the check-in points of p_0 and p_i.

$$score = -\left|\left|e_0 + h - e_i\right|\right|^2(7)$$

It is worth noting that cold-start users often have a small number of initial POI check-ins other than only one check-in. In this case, we can construct a testing meta-task as what we have done to construct a meta-task to fine-tune the trained model. That is, we construct a support set by generating the positive and negative short trajectories, and use the trajectories in the support set to train the model in the way of inner-loop optimization process. Once fine-tuned, we build the check-in points for each candidate POI and concatenate the last check-in point of user's trajectory with these built points to construct the two-point trajectories, and predict the next POI.

IV. EXPERIMENTS

We describe the datasets used for experiments, the data preprocessing operations, the parameter settings, and the experimental results in this section.

A. Experimental Setup

Dataset and Preprocessing: This paper uses two public datasets to evaluate the proposed method: Gowalla and Brightkite[24]. Each dataset contains the information including user ID, POI ID and check-in time. A single POI also contains specific longitude and latitude coordinates. The time information in the data set is in the form of hh:mm:ss. In this paper, the time in a year is converted into a number from 1-168 (7*24). We have selected 30% of the total users as cold-start users for testing. To simulate the cold-start environment, the first 10 check- in points in the trajectories of these selected users are kept to constitute the testing set. In the meanwhile, the remaining users and their trajectories are used for training, and the first 150 check-in points of the trajectories are kept to constitute the training set. Moreover, the POIs whose check-in times are less than 10 are removed. The two datasets are shown in Table I.

TABLE I DATA SET STATISTICS

Datasets	#user	*#POI*	*#checkin*
Gowalla	31708	131329	2963373
Brightkite	5247	37344	1699579

Experimental Settings: Because there are so many POIs candidate for recommendation, we randomly select 100 of them as the candidates for recommendation during testing. And to measure the performance of the proposed method, the metrics of normalized cumulative loss gain (nDCG)[25] and Hits Rates are used as indicators in this paper. The definition of them are shown in Eq.(8) and Eq.(9). In the Eq.(8), $|REL|$ means that the candidate POIs are sorted in descending order according to their relevance, and the top-p POIs are recommended. In the definition of HitsRates, Top@K means the number of the right predictions when the ground truth is in the top K recommended targets among the N predictions. What's more, the average of the values of these metrics in five experiments are taken as the final results.

$$nDCG_p = \frac{DCG_p}{IDCG_p} = \frac{\sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}}{\sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}} \quad (8)$$

$$HitsRates = \frac{N}{Top@K} \quad (9)$$

B. Baselines

To evaluate the performance of the proposed method, we have selected the following methods as baselines for comparison.

PRME-G(2013) used metric learning to transform users and places into the sequential switching space and user preference space to learn the user-specific switching mode[26].

GRU(2017) is a simple and improved model based on GRU4Rec[27], which has achieved good results in learning sequence information[28].

SASRec(2018) is the most advanced sequential recommendation method which is based on self-attention[29].

STGN(2019) enhanced the LSTM network by introducing ST-gates to capture the ST-relationship between successive check-ins[30].

GeoSAN(2020) proposed a geo-aware self-attention sequence position recommender, and updated the loss function based on importance sampling [18].

MetaTL(2021) adopts the meta-learning method to solve the cold-start problem in sequence recommendation[12].

C. The Comparison with the Baselines.

The results of the comparison between the proposed model STMeta of this paper and all the baselines are shown in Table II. Part of the comparative experimental data in this paper are obtained from the work of GeoSAN [18]. It can be seen that compared with MetaTL, STMeta has increased by 11% on Gowalla dataset and increased by 9% on Brightkite dataset in average in terms of all indicators. And it is worth mentioning that MetaTL, which is also a meta-learning model, is also improved by 15% compared with the traditional POI recommendation model GeoSAN. This shows that the meta-learning method can achieve excellent results in solving the cold- start problem of the next POI recommendation.

Among these methods, STGN has shown the worst performance under cold-start environment. This indicates that although it uses the neural networks to learn from user's trajectory, it still has no ability to deal with sparse datasets and cold-start problem. While adopting metric learning to embed users and items into the sequence transition space and user preference space with the aim to learn the user-specific transition mode, FPMC-G has also achieved excellent performance in cold-start problem. But in comparison, SASRec and GeoSAN which use Transfomer to extract sequence patterns work better. This may be because they can aggregate items through attention mechanism, and thus can obtain more information for users with limited interaction. Among them, GeoSAN is superior to SASRec because it uses geographic information encoder and combines the geographic spatial-temporal information of the POIs. This also indicates that the geographic spatial-temporal information is very important for solving POI recommendation problems. MetaTL is state of the art method to solve the cold-start problem of sequence recommendation. The experimental results in Table II reflect that it can achieve very good results by only using the information of the check-in trajectories when solving the sequence recommendation and POI recommendation problems.

TABLE II PRESENTATION OF RESULTS

	Gowalla				***Brightkite***			
	Hits@5	***NDCG@5***	***Hits@10***	***NDCG@10***	***Hits@5***	***NDCG@5***	***Hits@10***	***NDCG@10***
STGN	0.229	0.163	0.331	0.196	0.199	0.142	0.305	0.175
FPMC-G	0.401	0.314	0.502	0.347	0.367	0.268	0.490	0.307
SASRec	0.381	0.292	0.493	0.328	0.335	0.251	0.444	0.287
GeoSAN	0.495	0.389	0.623	0.434	0.525	0.413	0.643	0.449
MetaTL	0.651	0.559	0.717	0.601	0.556	0.489	0.615	0.510
STMeta	0.716	0.619	0.791	0.645	0.593	0.533	0.666	0.555

Compared with MetaTL, our method has achieved better performance. For example, the performance in terms of NDCG@10 on the dataset of Gowalla is increased from 0.510 to 0.555. This improvement is because we have taken the spatial-temporal information into the model and the results have shown its contribution to the next POI recommendation under cold-start environment.

It can be seen that the model of this paper has achieved better results on Gowalla data set, but the effect on Brightkite is poor. This may be because there are more cold-start users in Gowalla, and the distribution of users' POI check-in points is more sparse, mainly in geographical location and time distribution. In the meanwhile, Brightkite check-in points are more densely distributed.

D. Ablation Experiments

To solve the cold-start problem of next POI recommendation, the proposed STMeta has taken the spatial-temporal information into consideration. To further evaluate the function of the spatial and temporal information, we have done the ablation experiments. The results are shown in Table III.

TABLE III ABLATION EXPERIMENT

	Brightkite	
	NDCG@5	***NDCG@10***
Meta	0.489	0.510
T-Meta	0.501	0.524
L-Meta	0.516	0.537

The term of 'Meta' represents the method that we use the original MAML method to deal with the next POI recommendation under cold-start environment, and the 'T-Meta' denotes the improved MAML method by taking the temporal information into consideration. Similarly, the 'L-Meta' denotes the method by taking the spatial information of location into the MAML for next POI recommendation. The 'ST-Meta' means the proposed method of this paper which takes both the spatial and temporal information into the model. From the results shown in Table III, it can be seen that both the spatial and temporal information are helpful for the next POI recommendation under cold-start environment. For example, the performance of T-Meta and L-Meta is about 5% higher than that of Meta.

E. Parameter Sensitivity Analysis

In this paper, we have also tested the impact of the number K of short trajectories randomly selected from the long trajectories of users when constructing the meta-tasks which is showed in Fig.3. It can be observed that the indicators of NDCG@5 and NDCG@10 reach the optimal value when $K = 3$. This shows that with the increase of the number of sampled short trajectories, the recommendation performance decreases. The reason may be that the constructed meta-tasks may not be suitable for the cold-start users of this dataset. The cold-start users often have only a few check-ins. It may not contribute to model the behaviors of these cold-start users well when utilizing much information.

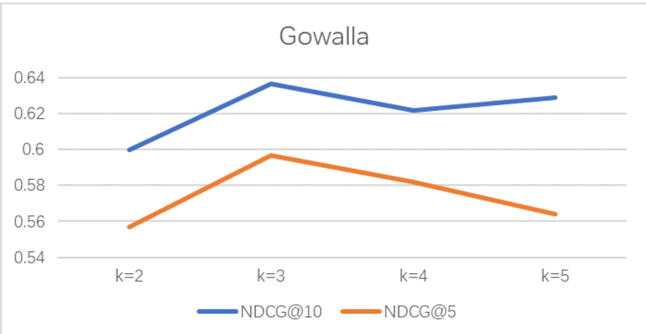

Fig.3 The sensitivity analysis of the number K of short trajectories randomly selected.

V. Conclusions

This paper proposes a next POI recommendation method for cold-start users by following the meta-learning method MAML. Its main idea is to simulate the cold-start environment to train a recommendation model with some transferrable and generalized knowledge which can be reused for cold-start users. Based on a large number of long trajectories of other users, it constructs a lot of short trajectories with two check-in points through sampling, and takes these short trajectories to train the model to estimate the spatial-temporal transition information over the two check-ins of a short trajectory. Once trained, the model can be further fine-tuned with the short trajectories of cold-start users and the next POI can be recommended according to the spatial-temporal transition information between the last POI that the target user has visited and the candidate POIs. Based on two public data sets of Gowalla and Brightkite, the proposed method is compared with a group of related methods. Experimental results show that it can improve the recommendation performance by 11% in average.

References

[1] M. Narayanan and A. K. Cherukuri, "A study and analysis of recommendation systems for location-based social network (lbsn) with big data," Iimb Management Review, pp. 25–30, 2016.

[2] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," ACM, 2015.

[3] Z. Cai, G. Yuan, S. Qiao, S. Qu, Y. Zhang, and R. Bing, "Fg-cf: Friends-aware graph collaborative filtering for poi recommenda- tion," Neurocomputing, vol. 488, pp. 107–119, 2022.

[4] Y. Liu, M. Yin, and X. Zhou, "A collaborative filtering algorithm with intragroup divergence for poi group recommendation," Applied Sciences, vol. 11, no. 12, p. 5416, 2021.

[5] J. He, J. Qi, and K. Ramamohanarao, "Timesan: A time- modulated self-attentive network for next point-of-interest rec- ommendation," in 2020 International Joint Conference on Neu- ral Networks (IJCNN). IEEE, 2020, pp. 1–8.

[6] Y. J. Park and A. Tuzhilin, "The long tail of recommender systems and how to leverage it," Acm Recsys, 2008.

[7] Z. K. Zhang, C. Liu, Y. C. Zhang, and Z. Tao, "Solving the cold-start problem in recommender systems with social tags," EPL (Europhysics Letters), vol. 92, no. 2, 2010.

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few- shot learning," 2017.

[9] R. Li, X. Wu, X. Wu, and W. Wang, "Few-shot learning for new user recommendation in location-based social networks," in Proceedings of The Web Conference 2020, 2020, pp. 2472– 2478.

[10] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," IEEE Transac- tions on Pattern Analysis and Machine Intelligence, vol. PP, no. 99, pp. 1–1, 2021.

[11] H. Sun, J. Xu, K. Zheng, P. Zhao, P. Chao, and X. Zhou, "Mfnp: A meta-optimized model for few-shot next poi recom- mendation," in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), 2021.

[12] D. K. Wang J, "Sequential recommendation for cold-start users with meta transitional learning," Proceedings of the 44th Inter- national ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1783–1787, 2021.

[13] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta- learning for fast adaptation of deep networks," in International conference on machine learning. PMLR, 2017, pp. 1126–1135.

[14] Q. Sun, Y. Liu, T. S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[15] Q. Guo, Z. Sun, J. Zhang, and Y. L. Theng, "An attentional recurrent neural network for personalized next location recom- mendation," 2020, pp. 83–90.

[16] K. Sun, T. Qian, T. Chen, Y. Liang, Q. Nguyen, and H. Yin, "Where to go next: Modeling long- and short-term user prefer- ences for point-of-interest recommendation," in National Con- ference on Artificial Intelligence, 2020.

[17] F. Huang, S. Qiao, J. Peng, B. Guo, and N. Han, "Stpr: A personalized next point-of-interest recommendation model with spatio-temporal effects based on purpose ranking," IEEE Transactions on Emerging Topics in Computing, vol. PP, no. 99, pp. 1–1, 2019.

[18] D. Lian, Y. Wu, Y. Ge, X. Xie, and E. Chen, "Geography- aware sequential location recommendation," in Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 2009–2019.

[19] Z. Zhang, Y. Zhang, and Y. Ren, "Employing neighborhood reduction for alleviating sparsity and cold-start problems in user-based collaborative filtering," Information Retrieval Jour- nal, vol. 23, no. 4, pp. 449–472, 2020.

[20] J. Li, "Research on collaborative filtering recommendation algo- rithm based on matrix decomposition method," in International Conference on Mechatronics, 2015.

[21] W. Weng and H. Miao, "Context-based service recommendation system using probability model in mobile devices," in Interna- tional Conference on Enterprise Systems, 2017.

[22] C. Zcab, C. Wmab, D. Wei, C. Wpab, and M. Zhong, "Condi- tional restricted boltzmann machine for item recommendation," Neurocomputing, vol. 385, pp. 269–277, 2020.

[23] H. Gao, J. Tang, and H. Liu, “Addressing the cold-start problem in location recommendation using geo-social correlations,” Data Mining and Knowledge Discovery, vol. 29, no. 2, pp. 299–323, 2015.

[24] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: user movement in location-based social networks,” in Proceed- ings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 1082–1090.

[25] Y. Wang, L. Wang, Y. Li, D. He, T. Y. Liu, and W. Chen, “A theoretical analysis of ndcg type ranking measures,” Journal of Machine Learning Research, 2013.

[26] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, “Personalized ranking metric embedding for next new poi recommendation,” in Twenty-Fourth International Joint Con- ference on Artificial Intelligence, 2015.

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

[28] R. Dey and F. M. Salem, “Gate-variants of gated recurrent unit (gru) neural networks,” in 2017 IEEE 60th international mid- west symposium on circuits and systems (MWSCAS). IEEE, 2017, pp. 1597–1600.

[29] W. Kang and J. McAuley, “Self-attentive sequential recommen- dation,” in 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018, pp. 197–206.

[30] P. Zhao, A. Luo, Y. Liu, F. Zhuang, J. Xu, Z. Li, V. S. Sheng, and X. Zhou, “Where to go next: A spatio-temporal gated network for next poi recommendation,” IEEE Transactions on Knowledge and Data Engineering, 2020.