



# Interest-Aware Next POI Recommendation for Mobile Social Networks

Ming Chen<sup>1</sup>, Wenzhong Li<sup>1,2</sup>(✉) , Lin Qian<sup>1,3</sup>, Sanglu Lu<sup>1,2</sup>,  
and Daoxu Chen<sup>1</sup>

<sup>1</sup> State Key Laboratory for Novel Software Technology,  
Nanjing University, Nanjing, China  
chenming@dislab.nju.edu.cn, lwz@nju.edu.cn

<sup>2</sup> Sino-German Institutes of Social Computing,  
Nanjing University, Nanjing, China

<sup>3</sup> NARI Group Corporation/State Grid Electric Power Research Institute,  
Nanjing, China

**Abstract.** Recommending the next point-of-interest (POI) to mobile users is an interesting topic for mobile social networks to provide personalized location-based services. In this paper, we propose an interest-aware next POI recommendation approach, which consider the location interest among similar users and the contextual information (such as time, current location, and friends preference) for POI recommendation. We develop a spatial-temporal topic model to describe users location interest, based on which we form comprehensive feature representations regarding user interest and contextual information. We propose a supervised learning prediction model for next POI recommendation. Experiments based on the Gowalla dataset verify the accuracy and efficiency of the proposed approach.

**Keywords:** Location-based service · POI recommendation  
Mobile social network

## 1 Introduction

With the development of mobile Internet in the recent years, various location-based services (LBS) in mobile social networks are becoming more and more popular. In the LBS applications, users upload their locations to the LBS server, such as checking-in a restaurant using Foursquare [1]. The check-in locations are

---

This work was partially supported by the National Key R&D Program of China (Grant No. 2017YFB1001801), the National Natural Science Foundation of China (Grant Nos. 61672278, 61373128, 61321491), the science and technology project from State Grid Corporation of China (Contract No. SGSNXT00YJJS1800031), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing.

known as point-of-interests (POIs) representing the stores, bars, restaurants that maybe interested to the users. Next POI recommendation is an important topic in mobile social network, which has a broad range of applications such as Ad pushing, city plan, traffic prediction, emergency alert, etc.

Numerous works had been done on next POI recommendation. Some works simply recommended the most popular nearby locations to the users [2]. Such recommendation is not personalized and inaccurate. The Matrix Factorization approaches [9] adopted conventional recommendation algorithms such as Collaborative Filtering (CF) [12] and FPMC [14] to recommend POI based on the similarity of check-in behaviors among users. However, they neglected the context of the users. Some works proposed the usage of Markov Chain for POI prediction [22]. However, they failed to capture the temporal dependency and personalized location preferences. A few works predicted next POI based on neural network (NN), which is a black box and its factors are not easy to be explained. In this paper, we propose an interest-aware next POI recommendation approach, which comprehensively consider the location interest among similar users, the contextual information such as time and current location, and the location preferences of friends in mobile social network, to achieve high accurate POI prediction.

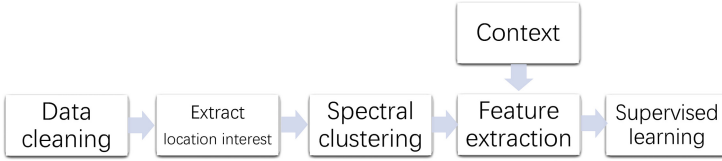
Intuitively, the historical check-in records reflect the location interest of a user, and the users with similar interests may have similar check-in preference. To extract location interest information, we propose a spatial-temporal topic model based on latent Dirichlet Allocation (LDA), which takes the historical POI check-in records as a document to represent users' interest as the distribution among a number of topics. We further apply a spectral clustering algorithm to find the group of users with similar interests. We combine the extracted interest information with contextual features (e.g., current location, time, friends) to build a supervised learning model for next POI recommendation. We test the performance of the proposed model using the dataset from Gowalla [5]. Extensive experiments show that the proposed interest-aware POI prediction approach outperforms the conventional approaches.

Our main contributions are summarized as follows.

- **A spatial-temporal topic model to extract users' interest.** We propose a spatial-temporal topic model based on LDA which is well used in NLP (natural language process), to extract users' interest on locations.
- **Comprehensive feature representations.** We comprehensively extract several features including the location interest among similar users, the contextual information of time and current location, and the location preferences of friends in mobile social network, which are helpful for POI prediction.
- **An interest-aware next POI recommendation approach.** We proposed an interest-aware POI prediction model based on supervised learning, which results are well visualized and well explained.

## 2 Related Work

In this section, we review related work about location recommendation models and location recommendation system.



**Fig. 1.** The solution framework.

**Location Recommendation Models:** The trivial model is to recommend the most popular location as the next POI. Obviously, this recommendation is simple and inefficient. In the field of recommendation, Matrix Factorization (MF) method is a traditional way that factorize a user-item matrix into two low rank matrices, which product is the user-item matrix. [10] recommended POI using the conventional collaborative filtering. Zheng et al. [20] constructed location-activity, location-feature, and activity-activity matrices, and used a collective matrix factorization method to mine POI and activities. Tensor Factorization (TF) could add a time dimension to original Matrix Factorization. Zheng et al. [21] also used tensor factorization on the user-location-activity relationship to mine POI and activities. Markov chain is a traditional model that could recommend next POI based on user’s past behavior. Zimdars et al. [22] used temporal data for making recommendations based on Markov chains. Rendle et al. [14] proposed Factorizing Personalized Markov Chains Model which subsumes Matrix Factorization and Markov chain together, learned an own transition matrix for each user, and outperformed both Matrix factorization and unpersonalized Markov chain model. There are also some work based on neural network. Zhang et al. [19] introduced a novel framework based on Recurrent Neural Networks (RNN) which directly models the dependency on users sequential behaviors into the click prediction process. Liu et al. [11] extended RNN and proposed ST-RNN to model continuous time interval and geographical distance for predicting the next location. Theses neural network [11, 17–19], unlike the traditional algorithm, although it could achieve good performance, it is not easy to be explained.

**Location Recommendation Systems:** COMPASS [16] recommended locations to tourist based on weather, traffic conditions, tourist’s profile, shopping list, schedule. MobyRec [15] could recommend restaurants based on users’ preferences. These recommendation systems required user declare their interest such as which kind restaurant to eat. Our recommendation is different from above, while only using check-in log data, users do not need to specify their preferences.

### 3 Problem Formulation

We consider the scenario that people use mobile social network APPs such as Foursquare, Gowalla, etc., to check-in their locations. Assume the trajectories of users’ check-in history are known. For a check-in event of a user, given his

current location, the time stamp, the trajectory, and the friends of the user on the social network, the next POI recommendation problem is to predict the most probable next check-in location of the user. In another word, we want to find a mapping  $f(\text{current location}, \text{current time}, \text{trajectory}, \text{friends}) \rightarrow \text{next POI}$ , which predicts the next check-in POI to match the actual check-in location of the user as well as possible.

## 4 Interest-Aware Next POI Recommendation

### 4.1 Solution Framework

According to the description of next POI recommendation problem, directly taking the historical trajectory as input to predict the next POI is infeasible. Intuitively, we can extract location interests from the historical trajectory, and use them as features to build a prediction model. Location interest refer to the location preference of a user under some context. For example, some users would like to go to the gym after work. Some users tend to visit the bar after dinner. Exploring the location interest can help to build a prediction model for POI recommendation. The framework of the proposed interest-aware next POI recommendation contains the following processes, which are shown in Fig. 1.

- **Data cleaning.** Some of the collected could be unqualified due to some users may disable their GPS or barely check-in with the APPs. In the first step, we filter out the unqualified data which is unsuitable to be used for the prediction model.
- **Extract location interest.** We adopt a temporal-spatial Latent Dirichlet Allocation (LDA) model to extract users’ location interests from their historical trajectory, and form a user-interest matrix.
- **Spectral clustering.** We apply a spectral clustering algorithm to group users into different clusters, where the users in the same cluster have similar location interests.
- **Feature extraction.** We further combine the extracted location interests with a Markov train model to form the interest features and context features representing by a number of transition matrices.
- **Supervised learning for next POI recommendation.** Using the extracted features as input, we propose a supervised learning approach to predict the most probable next POI of the user.

The details of the framework are explained below.

### 4.2 Data Cleaning

We use the mobile social network dataset from Gowalla [5]. Gowalla is a location-based social networking service where users share their locations by checking-in. The Gowalla dataset contains a total of 6,442,890 check-ins of 196,591 users over the period of Feb. 2009–Oct. 2010.

In our work, we extract the users from the city San Francisco from the dataset. Since some of the users are barely check-in their locations, we filter out the users whose check-in locations is less than 10. Since some of the locations are barely checked, we filter out the locations which have less than 10 check-ins. Before data cleaning, the extracted dataset has 6189 users, 11371 locations, and 152860 check-ins. After filtering out the unqualified users and locations, we obtained a dataset with 1995 users, 3251 locations, and 106,098 check-ins. The average check-in number for each user is 53 times, the minimum is 10, the maximum is 1604, and the median is 22. On average, each location is checked-in 32 times, the minimum is 10, the maximum is 948, and the median is 19.

### 4.3 Extract Location Interests

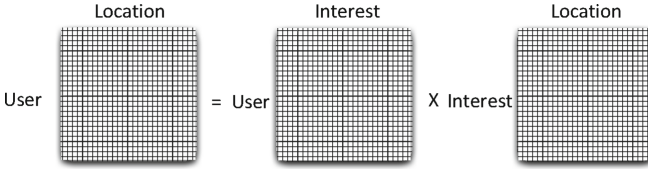
It's evidently that check-in a place at different time means different kind or different degree of location interest. For example, if a user checks-in a library at 10am on weekend, he may want to read books. If a user regularly checks-in the library at 10pm on weekday, he probably works at the library. In order to extract users' interests, both time and location factors should be considered. We propose a temporal-spacial latent Dirichlet Allocation (LDA) model to extract interests. In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that assumes each document is associated with a topic distribution, and each topic is associated with a word distribution. Specifically, we treat the combination time and location of a check-in event as a word, and the historical trajectory of a user can be view as a document. An example is illustrated in Table 1, where time is represented by two values: a Boolean value indicating whether it is weekday or not and an integer representing the hour of the day. We apply the LDA model on all documents of all users, and convert their historical check-in trajectories into interest distribution represented by  $K$  topics. As illustrated in Fig. 2, we obtain a matrix of user-interest distribution to describe the interest of the users. The user-location matrix shows the user distribution on each location, which comes from the counting of the trajectory. The user-interest matrix shows the users' interest distribution, where the number of the  $i$ -th row  $j$ -th column shows how the  $i$ -th user be interested in the  $j$ -th interest topic. The interest-location matrix shows the interest distribution on each location. These latter two matrices come from the LDA process which use Gibbs sampling [6] as inference technique.

To determine the best number  $K$  of the topics of LDA, we use the cross validation method to show the likelihood. We varies the LDA topic number  $K$  from 5 to 95, and draw a figure of likelihood as shown in Fig. 3. It is observed that the maximum likelihood achieved for  $K = 40$ . Therefore 40 topics are chosen in our experiments.

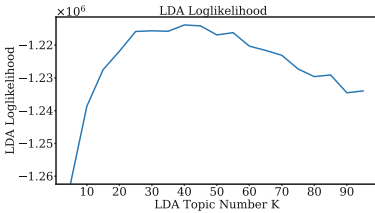
To show the convergence of the LDA model, we show the change of likelihood with the number of iterations in Fig. 4. As shown in the figure, the LDA model converges after 30,000 iterations.

**Table 1.** The example of treating time and location as word.

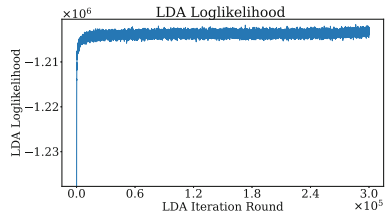
Location	Time	Word
252	2010-06-10T19:52:54Z	(252, 1, 19)
245	2010-08-29T20:20:56Z	(245, 0, 20)
...	...	...



**Fig. 2.** The LDA approach.



**Fig. 3.** Cross validation for K.



**Fig. 4.** Loglikelihood for K = 40.

### 4.4 Spectral Clustering

After obtaining the location interest of individual users, we group the users into clusters according to the similarity of interests. The intuition is that the check-in data of individual could be sparse and grouping similar users together can obtain statistical preferences for next POI recommendation.

We adopt the spectral clustering [13] approach to form the user groups. The input is the extracted interests, and the output is the class ID for each user regarding to their interest.

We choose spectral clustering to partition user groups due to the following reasons. On one hand, spectral clustering requires only a matrix of similarity between the users and therefore it is very efficient for clustering sparse data. The traditional clustering algorithm such as K-Means is hard to achieve this. On the other hand, spectral clustering can reduce the dimensions of the input space, therefore the complexity in processing high-dimensional data is better than the traditional clustering algorithms. Further, the result of spectral clustering can be easily visualized, as will be demonstrated later.

First, we specify the input of spectral clustering. After applying the LDA process, we get two matrices: the user-interest matrix  $M^{UI}$  and the interest-location matrix  $M^{IL}$  which show the user-interest distribution and interest-location distribution. The user-interest matrix  $M^{UI}$  is our input of

spectral clustering. The  $i$ -th row of user-interest matrix  $M^{UI}$  is a vector like  $[M_{i,0}^{UI}, M_{i,1}^{UI}, \dots, M_{i,K-2}^{UI}, M_{i,K-1}^{UI}]$  which is used to represent  $i$ -th user's interest. Each  $M_{i,k}^{UI}$  is in the range  $[0, 1]$  representing how the  $i$ -th user interested in the  $k$ -th interest topic. We draw the user-interest matrix as shown in Fig. 5(a).

Second, we explain how spectral clustering works. Free software to implement spectral clustering is available in the open source projects like Scikit-learn [3]. Spectral clustering contains the three steps:

- **(1) Pre-processing.** Construct a similarity graph for all users'  $M_i^{UI}$ . Then build Laplacian matrix  $L$  of the graph.

The adjacency matrix of similarity graph is as following:

$$A_{i,j} = \begin{cases} w_{i,j}, & \text{: weight of edge (i, j)} \\ 1, & \text{: if } i = j \end{cases} \quad (1)$$

The adjacency matrix is constructed using the Gaussian kernel function of the Euclidean distanced  $\|M_i^{UI} - M_j^{UI}\|_2$ .

$$w_{i,j} = \exp(-\gamma * \|M_i^{UI} - M_j^{UI}\|_2^2) \quad (2)$$

where  $\gamma$  is the kernel coefficient, default  $\gamma$  is 1.0

$D$  is the diagonal matrix of degrees.

$$d_i = \sum_{j|(i,j) \in E} w_{i,j} \quad (3)$$

The Laplacian matrix can be calculated by

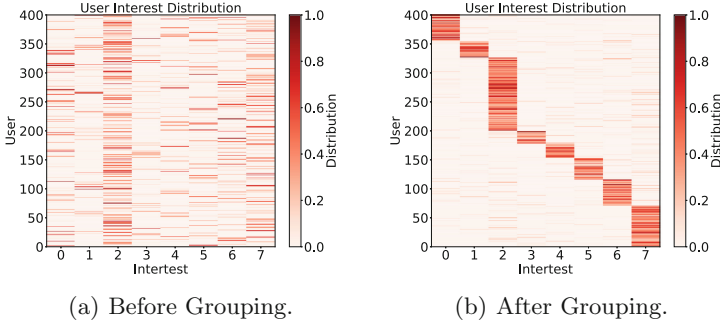
$$L = D - A \quad (4)$$

- **(2) Decomposition.** Build a reduced space from multiple eigenvectors. Here we use the Ng-Jordan-Weiss algorithm for spectral clustering [13]. First, compute eigenvectors of the matrix  $L_{norm}$ .

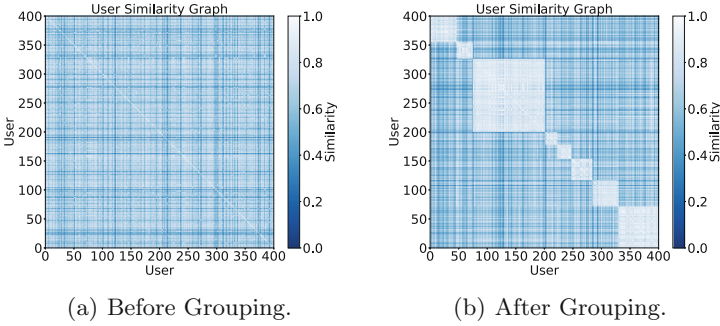
$$L_{norm} = D^{-1/2} L D^{-1/2} \quad (5)$$

Then find the first  $k$  eigenvectors  $\nu_1, \dots, \nu_k$  of the Matrix  $L_{norm}$ . Finally, let  $U \in R^{n \times k}$  be the matrix containing the vector  $\nu_1, \dots, \nu_k$  as columns. In this step, we map each point to a low denominational representation based on eigenvectors.

- **(3) Grouping.** A classical clustering algorithm (here is k-means) is applied to matrix  $U$  to partition the users. The output of spectral clustering assigns each user  $i$  with a class ID  $InterestClassID_i$ . Figure 5(b) shows the user-interest matrix after spectral clustering with the number of clusters setting to 40. It is shown that the users are grouped together with similar interest distributions. Figure 6 shows the similarity between user before and after spectral clustering. It clearly shows that similar users are grouped together forming clusters.



**Fig. 5.** Visualize user interest graph before and after grouping.



**Fig. 6.** Similarity graph before and after grouping.

### 4.5 Feature Extraction

Based on the obtained user interests and cluster IDs, we can form different types of features including interest features and context features, which will be used to build a prediction model. The feature extraction process including the following steps.

**Step 1:** We build a Markov transition matrix  $MC^{LOC}$  based on user trajectory, where each element  $MC_{i,j}^{LOC}$ , in the matrix means the transition probability of a user transiting from POI  $i$  to POI  $j$ .

The user’s trajectory can be represented by a sequence of POI IDs. We build a weighted graph  $G(V, E)$ , where  $V$  is the set of nodes indicated by POI IDs,  $E$  is the set of edges. If a user moves from a node to another, there is an edge between them. Initially the weights on the edge  $\omega'(E) = 0$  at the beginning. If we find a transit from POI  $ID_i$  to POI  $ID_j$ , let  $\omega'_{ID_i, ID_j} = \omega'_{ID_i, ID_j} + 1$ . After scanning the whole trajectory, we obtain  $G(V, E)$  with weight  $\omega'$  on every edges. Then we normalize the weights  $\omega'$  to  $\omega$ : for each

POI  $ID_i$ , set  $\omega_{ID_i, ID_j} = \frac{\omega'_{ID_i, ID_j}}{\sum_{j=1}^{j=POI\ number} \omega'_{ID_i, ID_j}}$ . We get the Matrix  $MC_{i,j}^{LOC}$ ,



where  $MC_{i,j}^{LOC}$  =  $\omega_{ID_i, ID_j}$  are the weights between two edges that satisfies  $\sum_{j=1}^{j=POI\ number} \omega_{ID_i, ID_j} = 1$ . Generally speaking,  $MC^{LOC}$  is a 2-dimension matrix where each row sums to 1.

**Step 2:** Now based on the spectral clustering result, we build a Markov transition matrix  $MC^{Interest}$ , where  $MC_{c,i,j}^{Interest}$  means the transition probability of a user transiting from POI  $i$  to POI  $j$  on condition that the user's cluster ID satisfies  $ClassID = c$ .

The construction of matrix  $MC^{Interest}$  is similar to that of  $MC^{LOC}$ , except that the trajectories are chosen from the set of users belonging to the same cluster  $c$ .

We may find the situation that some users in  $ClassID_c$  never go from  $POI_i$  to anywhere. In this situation, in order to get locations to recommend, we set  $MC_{c,i}^{Interest} = MC_i^{LOC}$ . That means copying a whole row in  $MC_i^{LOC}$  to  $MC_{c,i}^{Interest}$ . It helps to fill the blank data. That also means that when it comes a new data with property  $c$  and current location  $i$  that we do not know which next POI to recommend (because the property  $c$  and current location  $i$  never occurred before), we can ignore the property  $c$  and simply use the current POI  $i$  to recommend next POI.

**Step 3:** Further, we extract the context features (such as time, friends, etc.) represented by Markov transition matrices. Besides interest, we could apply the similar method in Step 2 to obtain different kinds of context features such as: weekday, hour, log (friends number), and so on. Each feature could be represented by a transition matrix under the property condition.

The transition matrices we got will be used to predict the next POI location to be recommended.

#### 4.6 Supervised Learning for Next POI Recommendation

After feature extraction, we obtain the interest features and context features represented by a number of transition matrices. Using the features as input, we can build a supervised learning model to predict the next POI. The process works as follows.

**Step 1: Construction of training set.** Here is how the training set is constructed. Based on the transition matrices regarding interest and context such as location, time, and friends, we choose the POIs with maximum transition probability as candidates for POI recommendation. Specifically, given the user ID, cluster ID  $c$ , and current POI  $i$ , we set  $Loc_{loc} = Max(MC_i^{LOC})$ , which is the location with maximum transition probability according to location context. We set  $Loc_{interest} = Max(MC_{c,i}^{Interest})$ , which is the location with maximum transition probability under interest group  $c$ . Similar, we obtain  $Loc_{time}$ ,  $Loc_{friend}$  from the transition matrices regarding the context of time and friends.

Using the candidate locations as features, we take the actual next POIs of the user as labels, a training set can be formed with the format like  $[CurLoc, Loc_{loc}, Loc_{interest}, Loc_{time}, Loc_{friend}] => [nextPOI]$ .

**Step 2: Learning.** We adopt the random forest [4,8] algorithm to train a prediction model. Random Forest is an ensemble classifier that consists of many decision trees. It outputs the class that is the mode of the class’s output by individual trees. Here we choose no limit to the leaf node and tree depth. Learning random forest is fast, and could produces a highly accurate classifier.

## 5 Performance Evaluation

In this section, we conduct experiments based on the Gowalla dataset [5] to evaluate the performance of the proposed approach.

### 5.1 Baseline Algorithms

We compare the proposed approach with four traditional POI recommendation algorithms.

- Top. Choose the most popular locations as next POI for each user.
- Random. Choose random locations as next POI for each user.
- Markov Chain (MC) [22]. Based on the current location, choose the most likely locations from Markov transition matrix as next POI for each user.
- Factorizing Personalized Markov Chain (FPMC) [14]: It is a sequential prediction method which based on personalized Markov chains.

### 5.2 Default System Parameters

The default parameters are as follows. The number of decision trees in random forest is 10. The number of check-ins for each user and each location is at least 10 times. The LDA topic number is 40. The spectral clustering parameters  $N$  is 160. The training set is set to 80% of the dataset.

**Table 2.** Performance comparison.

	Recall	Precision	F1-score
Interest-aware	<b>0.101697</b>	0.192546	<b>0.125091</b>
FPMC	0.087795	<b>0.287373</b>	0.122852
MC	0.088313	0.211404	0.116741
Top	0.000484	0.003119	0.000720
Random	0.014722	0.000217	0.000427

**Table 3.** The impact of tree number.

	Recall	Precision	F1-score
Tree = 10	0.101697	0.192546	0.125091
Tree = 20	<b>0.101791</b>	<b>0.192579</b>	<b>0.125138</b>
Tree = 30	0.101744	0.192555	0.125107
Tree = 40	0.101744	0.192555	0.125107

**Table 4.** The impact of LDA parameter K

	Recall	Precision	F1-score
K = 20	0.102545	0.190029	0.125066
K = 40	0.101744	0.192555	0.125107
K = 60	0.102403	<b>0.195227</b>	0.126255
K = 80	<b>0.102828</b>	0.193236	<b>0.126832</b>

### 5.3 Comparison with Baselines

We adopt the commonly used performance metrics including precision, recall, and F1-score for performance evaluation. The definitions of the performance metrics can be found in [7]. We compare the performance of the proposed interest-aware approach with the baselines, which results are shown in Table 2. According to the table, the proposed approach performance much better than the baselines. The Random and Top approaches perform poor in precision and recall since they ignore the personalized location preference and contextual information. The MC approach works good in precision, but it performs poor in recall. FPMC which extent MC, also performs poor in recall too. As F1-score evaluate both recall and precision, our approach outperform the FPMC. By combining location interest and contextual information for prediction, the proposed approach achieves better performance than the baselines.

### 5.4 Sensitive Analysis

Next we conduct sensitive analysis by showing the performance under different system parameters.

**The Impact of the Number of Decision Trees.** The random forest algorithm relies on a set of decision trees for prediction, and the number of decision trees is an important system parameter for the model. In the experiments, we varies the number of decision trees from 10 to 40, and the results are shown in Table 3. The number of decision trees has effect on the performance. The best performance achieved when  $Tree = 20$ . While bigger tree number means more memory cost ( $O(Tree)$ ), the improvement is not so much. It is not worthy use bigger tree number for a little performance improvement.

**Table 5.** The impact of parameter N.

	Recall	Precision	F1-score
N = 40	0.098021	0.186868	0.121204
N = 80	0.099717	0.187972	0.122556
N = 160	<b>0.101697</b>	0.192546	<b>0.125091</b>
N = 320	0.101037	<b>0.193670</b>	0.124512

**Table 6.** The impact of train set size.

	Recall	Precision	F1-score
Train = 60%	0.094910	0.179373	0.117071
Train = 70%	0.099749	0.186864	0.122339
Train = 80%	0.101697	0.192546	0.125091
Train = 90%	<b>0.102733</b>	<b>0.198377</b>	<b>0.126699</b>

**The Impact of the Number of Topics in LDA.** The number of topics  $K$  is an important parameter for the LDA model. In the experiments, we varies  $K$  from 20 to 80 and the results are shown in Table 4. The performance raise with the number of topics  $K$  raise. More topic number  $K$ , means more information about users, also means more memory and computation cost (both  $O(K)$ ), while the performance improve not too much.

**The Impact of the Number of Clusters.** The number of clusters  $N$  is an important parameter for the spectral clustering algorithm. In the experiments, we vary  $N$  from 40 to 320, and the results are shown in Table 5. The performance raise with  $N$  varies from 40 to 160, and performance fall with  $N$  varies from 160 to 320. Smaller  $N$  means to recommend user some POIs from grouped similar users, which is not too personalized. While bigger  $N$  means to recommend user more personalized POI, meanwhile some POIs from grouped similar users could not be recommend. At the point  $N = 160$ , the model performs best.

**The Impact of Train Set Size.** We change the size of training set from 60% to 90%, and the results are shown in Table 6. The performance raise with the percentage of training data raise. More training set achieved better perform because of more information about users.

## 6 Conclusion

In this paper, we address the next POI recommendation for mobile social networks, and propose a novel interest-aware POI prediction model to solve the problem. We develop a spatial-temporal topic model based on LDA that takes the historical POI check-in records as input to extract users' location interests. We further combine location interest with contextual features to construct a prediction model based on supervised learning for next POI recommendation. We conduct experiments on the Gowalla dataset, which show that the proposed POI prediction approach outperforms the conventional approaches.

## References

1. <https://foursquare.com/>
2. <https://www.lifewire.com/location-apps-for-user-reviews-tips-3485920>
3. <http://scikit-learn.org/stable/modules/clustering.html>
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1082–1090. ACM (2011)
6. Darling, W.M.: A theoretical and practical implementation tutorial on topic modeling and Gibbs sampling. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: HuMan Language Technologies*, pp. 642–647 (2011)

7. Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and  $F$ -score, with implication for evaluation. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 345–359. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-31865-1\\_25](https://doi.org/10.1007/978-3-540-31865-1_25)
8. Ho, T.K.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition 1995, vol. 1, pp. 278–282. IEEE (1995)
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
10. Levandoski, J.J., Sarwat, M., Eldawy, A., Mokbel, M.F.: LARS: a location-aware recommender system. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 450–461. IEEE (2012)
11. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: AAAI, pp. 194–200 (2016)
12. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
13. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856 (2002)
14. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820. ACM (2010)
15. Ricci, F., Nguyen, Q.N.: Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intell. Syst.* **22**(3), 22–29 (2007)
16. van Setten, M., Pokraev, S., Koolwaaij, J.: Context-aware recommendations in the mobile tourist application COMPASS. In: De Bra, P.M.E., Nejd, W. (eds.) AH 2004. LNCS, vol. 3137, pp. 235–244. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-27780-4\\_27](https://doi.org/10.1007/978-3-540-27780-4_27)
17. Xing, S., Liu, F., Zhao, X., Li, T.: Points-of-interest recommendation based on convolution matrix factorization. *Appl. Intell.* **47**, 1–12 (2017)
18. Yang, C., Bai, L., Zhang, C., Yuan, Q., Han, J.: Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245–1254. ACM (2017)
19. Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B., Liu, T.Y.: Sequential click prediction for sponsored search with recurrent neural networks. In: AAAI, vol. 14, pp. 1369–1375 (2014)
20. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with GPS history data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1029–1038. ACM (2010)
21. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: Collaborative filtering meets mobile recommendation: a user-centered approach. In: AAAI, vol. 10, pp. 236–241 (2010)
22. Zimdars, A., Chickering, D.M., Meek, C.: Using temporal data for making recommendations. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp. 580–588. Morgan Kaufmann Publishers Inc. (2001)