Knowledge-Based Systems 128 (2017) 59–70

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

CTF-ARA: An adaptive method for POI recommendation based on check-in and temporal features

Yali Si[a,b,c], Fuzhi Zhang[a,b,*], Wenyuan Liu[a,b]

[a] School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
[b] The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao 066004, China
[c] School of Liren, Yanshan University, Qinhuangdao 066004, China

A R T I C L E   I N F O

Article history:
Received 5 October 2016
Revised 9 April 2017
Accepted 24 April 2017
Available online 25 April 2017

Keywords:
Social networks
Recommender systems
Adaptive recommendation algorithm
User activity
Collaborative filtering
Temporal features

A B S T R A C T

Point-of-interest (POI) recommendation in location-based social networks (LBSNs) can solve the problem of information overload by providing personalized recommendation service, which is of great value to both users and businesses. However, the existing POI recommendation methods have not considered the effect of diversity features in check-in data, thus leading to unsatisfactory recommendation results. To address this issue, in this paper we propose an adaptive POI recommendation method (called CTF-ARA) by combining check-in and temporal features with user-based collaborative filtering. We first use probability statistical analysis method to mine user activity and similarity features of check-in behavior, variability and consecutiveness features of temporal factor. Then we use K-means algorithm to divide the users into active users and inactive users, and devise a similar user filtering algorithm based on the proposed features. Finally, we utilize cosine similarity of different time slots smoothing technique to make POI recommendation, which can operate adaptively according to the activity of user. The experimental results on Foursquare and Gowalla datasets show that CTF-ARA can improve precision and recall compared to other POI recommendation methods.

1. Introduction

Location-based social networks (LBSNs) are becoming increasingly popular with the rich multi-dimensional information on social relationship, location, time and content [1], such as Foursquare, Facebook, Twitter [2] and Dianping. Users can post their own check-in information, and benefit from location, friend [3], music, advertisements recommendation and other services of LBSNs, which are important researches to solve the problem of information overloads.

Facing mass information of LBSNs and a large number of points-of-interests (POIs) in life (e.g., restaurants, theaters, and shopping malls), it is difficult for users to choose places where are suitable for their interests. For example, Foursquare has collected more than 65 million place shapes mapping businesses around the world until June 2016 [4]. It causes location information overload to the users, especially when they go to strange cities. To address this issue, POI recommendation has been introduced to LBSNs, which recommends places for users where they have not visited before based on the check-in data [5]. As an emerging application area of recommendation techniques, POI recommendation can not only help users to explore new POIs and enhance the users' experiences, but also help shoppers to send advertisements and improve their business benefits. Therefore, POI recommendation has important practical significance and theoretical value.

To provide POIs recommendation for users, a number of methods have been proposed. These methods combine geographical property [6,7], social connection [8] and temporal property [9] with collaborative filtering [10–12]. Unfortunately, they suffer from low precision and recall due to the following reasons.

(1) Most methods only consider single factor (e.g., time factor, check-in factor, and user factor) in POI recommendation and ignore the relations between factors, which causes poor performance of recommendation.
(2) The existing POI recommendation methods use all users and locations to make recommendation, which leads to unrelated recommendation results to user preference. Furthermore, the POI recommendation algorithms are designed for all users and the diversity between users is not considered, which makes most of users dissatisfy the POI recommendation service.

To address the problems above, we propose an adaptive method, CTF-ARA, to recommend personalized POIs by combining

* Corresponding author.
E-mail addresses: siyali2008@126.com (Y. Si), xjzfz@ysu.edu.cn (F. Zhang).

http://dx.doi.org/10.1016/j.knosys.2017.04.013

features mining and collaborative filtering. The underpinning idea of CTF-ARA is the features extraction of user check-in and time aspects, which can effectively reflect the characteristics in check-in activities. To provide adaptive POI recommendation for users with different activities, we use different schemes of time slots smoothing technology and similar user filter in CTF-ARA algorithm according to users' activities. Moreover, we enhance the personalization of recommendation by introducing multiple techniques. The main contributions of this paper are summarized as follows:

(1) We investigate the difference of users' check-in frequency and extract user activity feature form LBSNs check-in datasets by means of probability statistics analysis method. Moreover, we also extract similarity feature of user check-in behavior, variability and consecutiveness features of temporal factor from historical check-in information, which are important factors for POI recommendation.
(2) To further enhance correlation and personalization of POI recommendation, we adopt four technologies in our recommendation method. User activity clustering is used to divide the users into active users and inactive users. Time slot division with smoothing technology is used to reflect temporal correlation and solve the data sparsity issue. User-based collaborative filtering and similar user filtering mechanism are used to enhance the correlation between users.
(3) We propose an adaptive POI recommendation algorithm based on check-in and temporal features, which can operate adaptively according to the activity of user. Furthermore, we carry out experiments on two large-scale real LBSNs datasets collected from Foursquare and Gowalla to demonstrate its effectiveness.

The structure of this paper is as follows. In Section 2, the related work on time-based POI recommendation is discussed. In Section 3, the user check-in features and temporal features are extracted, and the adaptive recommendation algorithm based on the extracted features and user activity clustering is described. The experimental evaluation is discussed in Section 4, and we conclude the paper in Section 5.

2. Related work

Nowadays recommender systems have a variety of real-world applications (e.g., e-commerce, e-business [13], and LBSNs-based services [4]), and various recommender system softwares have been developed (e.g., Smart BizSeeker [14]). In this section, we mainly give the related work on POI recommendation in LBSNs. Current researches on POI recommendation focus mainly on exploring time factor, geographical influence, social relations based on user check-in information. These factors are usually combined together to achieve better recommendation results.

2.1. Time-based POI recommendation methods

Time factor is combined with recommendation algorithms in order to achieve better performance. According to the technology used, the existing time-based methods for POI recommendation can be classified into three categories: time slot-based methods, periodic time pattern deducing, sequential time factor.

The time slot-based methods utilize the temporal influence to make POI recommendation. In these methods, a day is split into 24 equal discrete time slots according to hours, and check-in data is divided based on check-in time. Gao et al. [15] investigated the temporal properties of user check-in behavior in LBSNs and proposed a location recommendation method. This method is mainly used low-rank matrix factorization to infer users' preference on locations at each time slot. Yuan et al. [16] proposed a time-aware POI recommendation method based on the user-based collaborative filtering and time slot influence. The authors combined the spatial influence with smoothing enhancement technology to improve recommendation performance. Furthermore, they developed a preference propagation algorithm for effective recommendation, which uses geographical-temporal influences aware graph to model check-in records [17]. However, hour-based time slots methods cause the data much sparser, because the density of the check-in dataset is very low. Moreover, they only consider the differences of users' check-in preferences on different time slots, the similarity of user check-in on continuous time slots does not take into account.

To avoid the data loss, some methods use periodic time pattern deducing to make POI recommendation. Zhang et al. [18] proposed a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. They differentiated users' check-in activities on weekdays and weekends, and estimated a time probability density using kernel density estimation. This method not only recommends locations to users, but it also suggests when a user should visit a recommended location. Hosseini et al. [19] chose weekly intervals to improve effectiveness of POI recommenders, and developed a probabilistic model to detect a user's temporal orientation based on temporal weekly alignments of users and POIs. Zhao et al. [20] designed a time indexing scheme according to month, weekday type, and hour slot for successive POI recommendation. Based on a fine-grained modeling of the interactions among time, user, and POI, they proposed a new ranking-based pairwise interaction tensor factorization framework.

The time sequence has also been used to recommend next POIs for users, which is different form time slot-based methods and periodic time pattern deducing methods. Cheng et al. [21] mined sequential patterns from location sequence using a dynamic location-location transition graph, and predicted the probability of a user visiting a location by Addictive Markov Chain. Wei et al. [22] extracted and studied the feature of temporal interval of consecutive check-ins, and proposed a trace-driven model for generating synthetic LBSN datasets capturing the properties of the original datasets. The personalized Markov chain in the check-in sequence is exploited for successive personalized POI recommendation [23]. Unfortunately, sequential time factor methods are used to predict an existing location, and they cannot recommend a new location or locations where a user has not visited before.

2.2. Geography-based POI recommendation methods

Geographical influence is a unique factor in POI recommendation, because the users' check-in behavior can be influenced by geographical features of real world locations. Recent studies use the geographical clustering to develop POI recommendation methods with better performance, which can be classified into the following three categories.

The power-law distribution [16,17,24] is used to model the willingness of users moving from one place to another as a function of their distance, and users are more willing to check in nearby POIs to their current places.

The literatures [6,9,25] modeled the distance between locations visited by the same user as a Gaussian distribution. It mainly includes two latent states: work and home, and two-dimensional time-independent Gaussian distribution is used to model the movement when a user is in the home or work state. Zhao et al. [26] proposed a genetic-based Gaussian mixture model to capture the geographical influence to improve the POI recommendation performance. Liu et al. [27] proposed a general geographical probabilistic factor model which takes various factors into account. Specifically, the model allows to capture the geographical influ-

Table 1
Notations and their descriptions.

Notation	Description
U_{All}	set of all check-in records in a LBSN
U	set of all users in a LBSN
L	set of all point-of-interests in a LBSN
T	set of all time slots
$num_{u,t}$	check-in counts of user u at time slot t
L_u	set of POIs visited by user u
$r_{u,t,l}$	binary check-in value of u visit l at t
$\hat{r}_{u,t,l}$	check-in value of u visiting location l at k consequent time slots
$\tilde{r}_{u,t,l}$	check-in value of u visiting location l at all time slots
$sim_{t,t'}$	similarity of two time slots t and t'
$sim_{u,v}^{(kt)}$	user similarity of k consequent time slots
$sim_{u,v}^{(at)}$	user similarity of all time slots
$p_{u,t,l}^{(kt)}$	recommended probability of u visiting new location l at consequent time slots
$p_{u,t,l}^{(at)}$	recommended probability of u visiting new location l at all time slots

ences on a user's check-in behavior, which is based on the assuming of a Gaussian geographical distribution.

The kernel density estimation model [7,23,28] was proposed to infers users' preferences by exploiting the personalized two-dimensional geographical influence, which used a nonparametric density estimation method to avoid the limitations of setting a common distance function for all users in advance. Because the user's check-in behavior is unique, the geographical influence of locations should be personalized.

2.3. Social relations-based POI recommendation methods

Social relations also influence users in choosing new POIs. Based on the friends in networks and their visited locations, the recommendation methods are devised by integrating social information into the collaborative filtering techniques [8]. Cheng et al. [29] applied the probabilistic matrix factorization with social regularization, and the social constraints ensured latent features of friends keeping in close distance at the latent subspace. However, the results of POI recommendations showed the limited improvement achieved from social influence [23]. The major reason is the cold-start problem that the users have few check-in records or friends, and the friends in LBSNs may share common interest but may not visit the same POIs due to their long distance. In order to solve the data sparsity and cold-start problem, Li et al. [30] defined three types of friends (i.e., social friends, location friends, and neighboring friends) in LBSN, and developed a two-step framework to leverage the information of friends to improve POI recommendation accuracy.

In this paper we focus on the time-based POI recommendation methods. In these methods, more diverse check-in features are not mined and combined with time information, and the methods do not have adaptability for different users with filtering mechanism, leading to low recommendation accuracy.

3. The proposed method

To facilitate the discussions, Table 1 gives the descriptions of notations used in this paper.

To improve the recommendation performance, we propose an adaptive POI recommendation method. Fig. 1 depicts the framework of our adaptive POI recommendation method. As depicted in Fig. 1, the framework mainly includes three parts: LBSNs data, features extraction, and adaptive recommendation. Based on large-scale history check-in datasets in LBSNs, the user check-in and temporal features are extracted using probability statistical analysis method. At the stage of adaptive recommendation, K-means algorithm is used to group users according to the activity feature, similar users are selected by the proposed filtering algorithm, and an adaptive algorithm CTF-ARA is devised to recommend Top-*N* POIs to different users. CTF-ARA introduces two strategies to achieve adaptability, which are consecutive time slots-based method for active users and all time slots-based method for inactive users. The focuses of our method are features extraction and adaptive recommendation, and the details of the framework will be discussed in the following sections.

3.1. Features extraction

The key to our adaptive POI recommendation method is to extract features from LBSNs datasets. In this section, we will extract two kinds of features, namely user check-in features (i.e., activity and similarity) and temporal features (i.e., variability and consecutiveness), using statistical analysis methods.

3.1.1. User activity feature

Users in LBSNs are different in check-in action and preference. Some users are keen to sign in frequently and share their experiences, while the others are not. We define this kind of check-in frequency as user activity feature.

Definition 1 (Activity of User, AoU)**.** The activity of user $u \in U$, AoU_u, is defined as the total check-in number of user u on a LBSN dataset U_{All}, which can be calculated as follows:

$$AoU_u = \sum_{i=1}^{|U_{All}|} r_{ui} \tag{1}$$

where r_{ui} is a binary value, $r_{ui} = 1$ if a record in U_{All} is checked by user u, otherwise $r_{ui} = 0$.

In order to indicate that the users have different activities, we compute each user's activity value based on Foursquare and Gowalla datasets. Fig. 2(a) shows the number of users with different activities on Foursquare. It can be seen that the users' activities differ obviously from each other and the distribution of *AoU* follows a kind of heavy-tailed distribution. The heavy-tailed distribution implies that a small number of individuals make a lot of contributions. In our study, it can be considered that a small number of users conduct a large number of check-in behavior. From Fig. 2(a), it also can be seen that the users whose check-in numbers reach up to $10^2 \sim 10^3$ are too few. Actually, 97 users check in more than 300 records and only 18 users check in more than 500 records in Foursquare dataset, while the total number of users is 2321 in this dataset. In order to further analyze the user activity, we conduct statistics research on the Cumulative Distribution Function (CDF) based on both datasets. As shown in Fig. 2(b), the plot exhibits that about 50% users have checked in less than 40 on Foursquare and 20 on Gowalla dataset respectively, which indicates that the majority of LBSNs users check in occasionally.

Based on Fig. 2 and the above analysis, we can infer that the users can be divided by the level of activity. Namely, there are two kinds of users to be explored: users who check in frequently and users who check in occasionally. Therefore, according to user activity feature, we can use user activity clustering method to divide users and devise the corresponding POI recommendation method to realize adaptively of recommendation.

3.1.2. User similarity feature

Although users have different check-in preferences, some users have common preference when checking the same places. We call users with the same preference as similar users, which can be measured by user similarity feature.

Definition 2 (Similarity of User, SU)**.** For $u, v \in U$, the similarity of users u and v, $SU_{u,v}$, is defined as the number of locations that

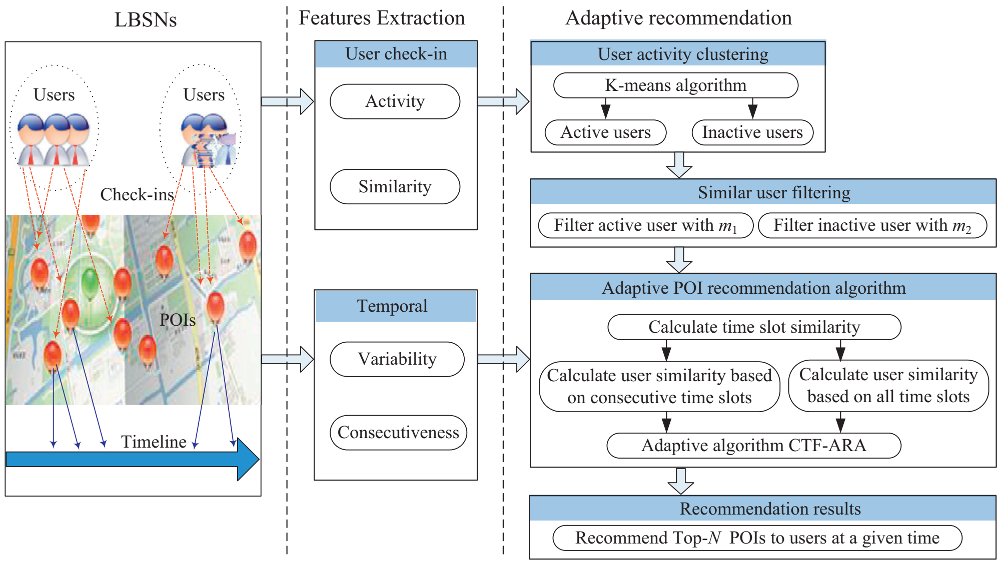


Fig. 1. Adaptive POI recommendation framework.

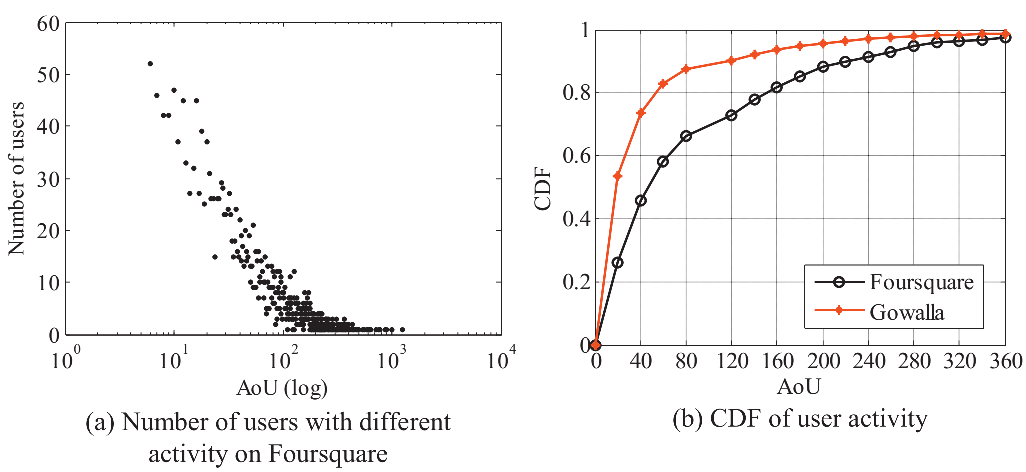


Fig. 2. Statistical results of user activity.

they have checked in common, which can be calculated as follows:

$$SU_{u,v} = \sum_{l \in L} r_{u,l} \cdot r_{v,l} \tag{2}$$

where $r_{u,l}$ is a binary value, $r_{u,l} = 1$ if user u checked location l, otherwise $r_{u,l} = 0$.

Similar users can provide some new candidate locations for a target user, so selecting high similarity users can not only reduce the computational overhead, but also reduce the recommended noise. In order to analyze the user similarity, we plot the cumulative distribution figure of user similarity on Foursquare and Gowalla in Fig. 3. It can be seen form Fig. 3 that the probability of checking one same location is 47.8% on Foursquare and 79.3% on Gowalla, respectively. This indicates that half or more users in whole user dataset have a very low degree of similarity for a target user, because two users have very little correlation if they checked in only one same location in a large number of POIs. Therefore, considering that smaller similarity has less impact on recommendation, we will filter the most dissimilar users in order to increase the accuracy of recommendation.

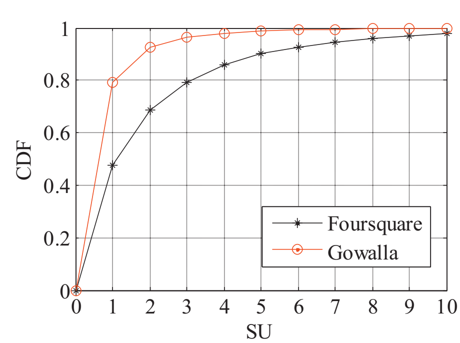


Fig. 3. The cumulative distribution of user similarity.

3.1.3. Temporal variability feature

Temporal information in LBSNs is very important for POIs recommendation at a specific time. The temporal variability means that a user has a distinct check-in preference at different hours of a day. For example, a user is willing to go to restaurant at noon, and he is more likely to go to a theatre at midnight. In this paper, we split a day into 24 equal time slots based on hour, namely

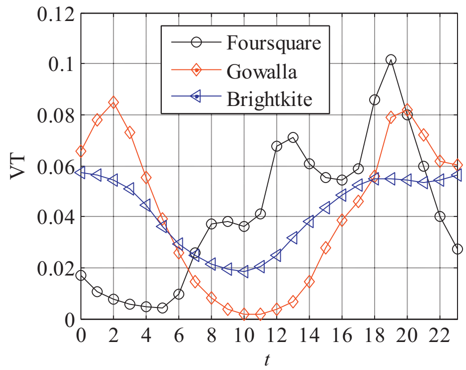


Fig. 4. *VT* values at different times.

$T=\{0, 1,\ldots, 22, 23\}$. Each time slot is denoted by t, $t\in T$ and t is an integer, e.g. 08:36:00 can be denoted as slot $t=8$.

Definition 3 (Variability of Time, VT)**.** The variability of time slot $t\in T$, VT_t, is defined as the ratio of check-in number at time slot t to the total check-in number in dataset U_{All}, which can be calculated as follows:

$$VT_t = \frac{\sum_{u\in U} num_{u,t}}{|U_{All}|} \tag{3}$$

For analysis purposes, the *VT* values on three datasets are computed and depicted in Fig. 4. It can be seen from Fig. 4, the user check-in differs significantly at each time slot, which illustrates that users have different preference at different times. Therefore, we will introduce temporal information to our recommendation algorithm.

3.1.4. Temporal consecutiveness feature

In real life, a user has more similar check-in preferences in consecutive time. For example, users may visit restaurants or other similar locations from 11:00 to 13:00. We can call this feature as temporal consecutiveness.

Definition 4 (Consecutiveness of Time, CT)**.** Let the time slots of user $u\in U$ visiting a same location be $t_{u,l}$ and $t'_{u,l}$, and Δt denote time difference, $\Delta t = t_{u,l} - t'_{u,l}$. Then, consecutiveness of time, $CT_{\Delta t}$, is defined as the ratio of Δt number to the total time difference number, which can be calculated as follows:

$$CT_{\Delta t} = \frac{\sum_{u\in U} num_{u,\Delta t}}{\sum\limits_{\Delta t=-11}^{12} \left(\sum_{u\in U} num_{u,\Delta t}\right)} \tag{4}$$

if $\Delta t>12$, we have $\Delta t=\Delta t-24$; if $\Delta t<-11$, we have $\Delta t=\Delta t+24$.

In order to mine the temporal consecutiveness feature, we plot *CT* values on Foursquare in Fig. 5. As shown in Fig. 5, the highest *CT* value can reach 21.86% at $\Delta t=0$, and *CT* value declines as the absolute value of Δt increases. The statistical results inspire us to use adjacent continuous time slots instead of only one time slot. This smoothing technology can expand the check-in preference and solve the problem of data sparsity, so we will combine temporal consecutiveness with the user activity feature.

3.2. Adaptive recommendation

In this section, we give the details of our adaptive POI recommendation method. We first perform clustering on user activity, then we select similar users according to user similarity feature, and based on which we devise an adaptive POI recommendation algorithm by introducing temporal smoothing technology.

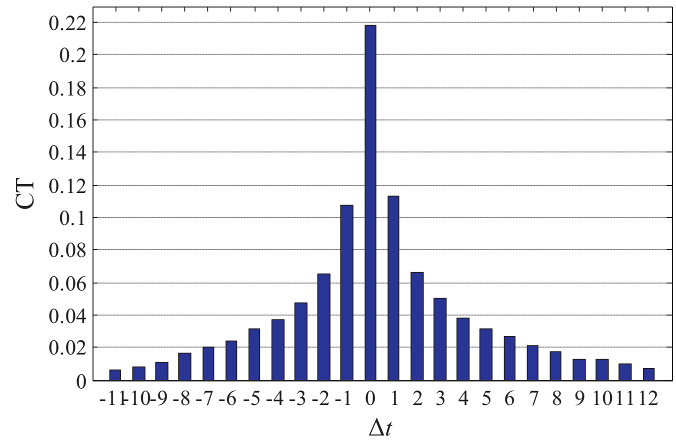


Fig. 5. Figure of temporal consecutiveness.

3.2.1. User activity clustering

To make better POI recommendation results, we first use K-means clustering algorithm to divide the users into two groups: active users and inactive users. The main steps of the user activity clustering algorithm are as follows.

Step 1: For each user $u\in U$, we compute user u's activity value AoU_u, and get a two-dimensional user-check set $UC=\{u_j, AoU_{u_j}|\ j=1, 2,\ldots, |U|\}$.

Step 2: Let clusters number K=2, the min check-in number $C_1(r)=\min\{AoU_{u_j}, j=1, 2,\ldots, |U|\}$ and the max check-in number $C_2(r)=\max\{AoU_{u_j}, j=1, 2,\ldots, |U|\}$ be the two initial cluster centers, r is the iteration number.

Step 3: For each point in UC, compute the similar distance with cluster centers, $D(AoU_{u_j},C_k(r))=|AoU_{u_j}-C_k(r)|$, $k=1, 2$, and assign it to the closest cluster to form cluster W_k. If $D(AoU_{u_j},C_k(r))=\min\{D(AoU_{u_j},C_{k1}(r)), k1=1,2\}$, then $AoU_{u_j}\in W_k$.

Step 4: Update each cluster's center by averaging all of the points that have been assigned to it, which is computed as follows.

$$C_k(r+1) = \frac{1}{n}\sum_{j=1}^{n(k)} AoU_{u_j}{}^{(k)} \tag{5}$$

Step 5: The goal of K-means algorithm is to minimize the sum of the squared error over all K clusters, which is computed by Eq. (6), and minimizing this objective function is known to be an NP-hard problem. Iterate between Step 3 and 4 until $E(r+1)$ convergence.

$$E(r+1) = \sum\nolimits_{j=1}^{k} \sum\nolimits_{num_j\in W_k} |AoU_{u_j} - C_k(r+1)|^2 \tag{6}$$

Based on the above steps, the User Activity Clustering Algorithm (UAC) is described as follows.

Algorithm 1

3.2.2. Similar user filtering

In order to choose the most similar users for target users, we devise a **F**iltering **S**imilar **U**sers **A**lgorithm (FSUA) to filter out dissimilar users for different active levels of users based on the user similarity feature. Similar user filtering not only can improve the recommendation accuracy, but also can reduce the computational overhead of user-based collaborative filtering recommendation [31,32]. Given a target user u and all check-in set U_{All}, the main idea of FSUA algorithm is to find the most adjacent similar users for u by judging whether or not similarity value $SU_{u,v}$ is greater than a threshold. Specifically, a user v is similar to the

Algorithm 1 User Activity Clustering Algorithm(UAC).

Input: The user-check dataset UC
Output: The active user set AU and inactive user set IAU
K←2
initialize cluster centers $C_1(r)$ and $C_2(r)$
repeat
  for each user $u_j \in UC$ **do**
    compute $D(AoU_{u_j}, C_k(r))$
    if $D(AoU_{u_j}, C_k(r)) = \min\{D(AoU_{u_j}, C_{k1}(r)),\ k1 = 1,2\}$ **then** $AoU_{u_j} \in W_k$
  end for
  update each cluster's center
until $E(r+1)$ convergence
return two user sets AU and IAU

Algorithm 2 Filtering Similar Users Algorithm (FSUA).

Input: The user check-in dataset U_{All} and the target user u
Output: The similar user set SU
1: **for** $i = 1$ to $|U_{All}|$ **do**
2:   **if** $UId = u$ **then**
3:     add LId to L_u
4:     add UId who check-in LId to LU
5:   **end if**
6: **end for**
7: **for** each $v \in LU$ **do**
8:   compute $SU_{u,v}$
9:   **if** (v is an active user and $SU_{u,v} > m_1$) or (v is an inactive user and $SU_{u,v} > m_2$) **then**
10:     add v to SU
11:   **end if**
12: **end for**
13: **return** SU

target active user u if $SU_{u,v} > m_1$ (m_1 is an integer); otherwise, v is not a similar user and is filtered out. The description of FSUA algorithm is given as follows.

Algorithm 2 mainly includes two parts. The first part, from lines 1 to 6, is to generate target user's location set L_u and user set LU. The second part, from lines 7 to 13, is to compute the similarity $SU_{u,v}$ and generate the similar user set.

3.2.3. Similarity calculation with temporal smoothing technology

Due to the sparsity of LBSNs dataset, the recommendation methods based only on the target time slot suffer from low precision. Therefore, we extend user similarity by consecutive time slots on the basis of cosine similarity and temporal feature. This is not hard to explain. If users always visit the same POI in a continuous period of time, their similarity will be high.

Definition 5 (Similarity of Time Slots, $sim_{t,t'}$)**.** The similarity between two time slots t and t', $sim_{t,t'}$, is the average of similarity values of all users in two time slots, which can be defined as follows:

$$sim_{t,t'} = \frac{\sum_{u\in U} sim^{u}_{t,t'}}{|U|} \tag{7}$$

$$sim^{u}_{t,t'} = \frac{\sum_{l\in L_u}(r_{u,t,l}\cdot r_{u,t',l})}{\sqrt{\sum_{l\in L_u} r^2_{u,t,l}}\sqrt{\sum_{l\in L_u} r^2_{u,t',l}}} \tag{8}$$

where $sim^{u}_{t,t'}$ denotes the similarity between two time slots t and t' for user u. $r_{u,t,l}$ is a binary check-in value, $r_{u,t,l} = 1$ if user u checked location l at time slot t; otherwise $r_{u,t,l} = 0$.

Definition 6 (User similarity based on k consecutive time slots, $sim^{(kt)}_{u,v}$)**.** For user $u \in U$, $v \in SU$, the similarity based on k consecutive time slots between u and v, $sim^{(kt)}_{u,v}$, is defined as follows:

$$sim^{(kt)}_{u,v} = \frac{\sum_{t\in T}\sum_{l\in L}\hat{r}_{u,t,l}\hat{r}_{v,t,l}}{\sqrt{\sum_{t\in T}\sum_{l\in L}\hat{r}^2_{u,t,l}}\sqrt{\sum_{t\in T}\sum_{l\in L}\hat{r}^2_{v,t,l}}} \tag{9}$$

$$\hat{r}_{u,t,l} = \sum_{t'=t-k}^{t+k}\left(\frac{sim_{t,t'}}{\sum_{t'=t-k}^{t+k} sim_{t,t'}}\times r_{u,t',l}\right) \tag{10}$$

where $\hat{r}_{u,t,l}$ denotes a new check-in value based on time slot using Eq. (10), which extends forward and backward time slot by k.

Definition 7 (User similarity based on all time slots, $sim^{(at)}_{u,v}$)**.** For user $u \in U$, $v \in SU$, the similarity based on all time slots between u and v, $sim^{(at)}_{u,v}$, is defined as follows:

$$sim^{(at)}_{u,v} = \frac{\sum_{t\in T}\sum_{l\in L}\tilde{r}_{u,t,l}\tilde{r}_{v,t,l}}{\sqrt{\sum_{t\in T}\sum_{l\in L}\tilde{r}^2_{u,t,l}}\sqrt{\sum_{t\in T}\sum_{l\in L}\tilde{r}^2_{v,t,l}}} \tag{11}$$

$$\tilde{r}_{u,t,l} = \sum_{t'\in T}\left(\frac{sim_{t,t'}}{\sum_{t'\in T} sim_{t,t'}}\times r_{u,t',l}\right) \tag{12}$$

where $\tilde{r}_{u,t,l}$ denotes a check-in value based on all time slots and computed by Eq. (12).

In order to make recommendation adaptively, similarity computation method based on k consecutive time slots is applied to active users and similarity method based on all time slots is applied to inactive users, respectively. The similarity calculation combined with temporal influence will solve the low user similarity problem in POI recommendation algorithm, and the smoothing technology will solve the data sparse problem.

3.2.4. Adaptive POI recommendation algorithm

By combining the user activity, similar user filtering and smooth similarity calculation, we propose a Check-in and Temporal Features based Adaptive Recommendation Algorithm (CTF-ARA). The adaptability is reflected in different recommendation strategies for different active users, and each strategy mainly contains three parts: similar user filtering, the similarity calculation and user-based collaborative filtering with smoothing technique. The main steps of our adaptive POI recommendation are described as follows.

Step 1: all users are divided into active users set AU and inactive users set IAU using user activity clustering algorithm.

Step 2: the dissimilar users whose similarity values are less than a given threshold are filtered out, and the similar user set SU is obtained for a target user.

Step 3: user similarities are computed and POI recommendation is generated. The following two strategies are used in CTF-ARA algorithm.

Strategy 1: For active users in AU, who have enough check-ins and similar users, we calculate the similarity $sim^{(kt)}_{u,v}$ between the target user u and similar user v based on k consecutive time slots. The recommendation probabilistic value $p^{(kt)}_{u,t,l}$ of user u will visit location l at time t is calculated as follows:

$$p^{(kt)}_{u,t,l} = \frac{\sum_{v\in SU}\left(sim^{(kt)}_{u,v}\cdot\sum_{t'=t-k}^{t+k} r_{v,t',l}\, sim_{t',t}\right)}{\sum_{v\in SU} sim^{(kt)}_{u,v}} \tag{13}$$

Strategy 2: For inactive users in IAU, who have less check-ins and similar users, we calculate the similarity $sim^{(at)}_{u,v}$ between the target user u and similar user v based on all time slots. The recommendation probabilistic value $p^{(at)}_{u,t,l}$ of user u will visit l at time t is calculated as follows:

$$p^{(at)}_{u,t,l} = \frac{\sum_{v\in SU}\left(sim^{(at)}_{u,v}\cdot\sum_{t'\in T} r_{v,t',l}\, sim_{t',t}\right)}{\sum_{v\in SU} sim^{(at)}_{u,v}} \tag{14}$$

Based on the above steps, the proposed CTF-ARA algorithm is described as follows.

Algorithm 3

Algorithm 3 CTF-ARA.

Input: The user check-in dataset UC, the target user u and time slot t
Output: The recommended Top-N POIs
1: call Algorithm 1 to get active user set AU and inactive user set IAU
2: call Algorithm 2 to get similar user set SU of u
3: **if** $u \in AU$ **then** // Strategy 1
4: **for** each $l \in L$ and $t' \in [t\text{-}k,\ t+k]$ **do**
5: compute $sim_{t,t'}$ using Eq. (7) and compute $\hat{r}_{u,t,l}$ using Eq. (9)
6: **end for**
7: **for** each $v \in SU$ and $l \in L$ and $t \in T$ **do**
8: compute $sim_{u,v}^{(kt)}$ using Eq. (10)
9: **end for**
10: **for** each $v \in SU$ and $t' \in [t\text{-}k,\ t+k]$ **do**
11: computer $p_{u,t,l}^{(kt)}$ using Eq. (13)
12: **end for**
13: **else** // Strategy 2
14: **for** each $l \in L$ and $t' \in T$ **do**
15: compute $sim_{t,t'}$ using Eq. (7) and compute $\tilde{r}_{u,t,l}$ using Eq. (11)
16: **end for**
17: **for** each $v \in SU$ and $l \in L$ and $t \in T$ **do**
18: compute $sim_{u,v}^{(at)}$ using Eq. (12)
19: **end for**
20: **for** each $v \in SU$ and $t' \in T$ **do**
21: computer $p_{u,t,l}^{(at)}$ using Eq. (14)
22: **end for**
23: **end if**
24: sort($p_{u,t,l}^{(kt)}$) or sort($p_{u,t,l}^{(at)}$)
25: **return** Top-N POIs

Table 2
Statistics of the overall datasets.

Items of datasets	Foursquare	Gowalla
Number of check-ins	194,108	456,988
Number of locations	5,596	24,250
Number of users	2,321	10,162
Check-in density	14.9×10^{-3}	1.85×10^{-3}

Algorithm 3 first divides the users and constructs similar user set of target user (Lines 1–2). Then, according to the target user's activity, recommendation probabilistic values of candidate POIs are computed by Strategy 1 for active users (Lines 3–12) or Strategy 2 for inactive users (Lines 13–23). Finally, the probability values are sorted and the top-N POIs are returned as recommendation results (Lines 24–25).

4. Experiments and analysis

4.1. Experimental datasets and setting

We select two large-scale LBSNs datasets as our experimental data, namely Foursquare dataset (.txt file is 11.8 M) and Gowalla dataset (.txt file is 25.7 M), which are used in [16]. The statistics of two datasets are shown in Table 2. Each check-in record in the datasets includes a user ID, a location ID, a location's latitude, a location's longitude, a time and a day ID. For fair comparison with the recommendation algorithms, we randomly mark 16% of each user's visited POIs as testing data to evaluate the effectiveness of the recommendation methods, and the rest 84% check-ins are marked as training data. The random selection is conducted five times separately, and the average results are reported finally. Moreover, all locations in the testing data are not in the training data for each user, because POI recommendation algorithms should recommend new locations for users where they have not visited before.

In the experiments of algorithm performance comparison and parameters adjustment, we use active users set and inactive users set divided from overall datasets. All users in each dataset are divided into active users or inactive users by Algorithm 1, and the data statistics of active users and inactive users are shown in Table 3. The proposed CTF-ARA algorithm is run to recommend POIs adaptively on the results of user activity clustering.

Table 3
Data statistics of active users and inactive users.

Items of datasets	Foursquare	Gowalla
User number in inactive cluster	2,034	9,938
User number in active cluster	287	224
Final cluster center of inactive users	53	35
Final cluster center of active users	300	501
Runtime of user activity clustering (second)	0.16	0.34
Check-in number of inactive users	116,568	294,812
Check-in number of active users	77,540	162,806

4.2. Evaluation metrics

We use precision and recall metrics to evaluate the effectiveness of the proposed method. The precision metric measures how many POIs in the Top-N recommended POIs corresponding to the POIs in the testing data, and the recall metric measures how many POIs in the testing set are returned as Top-N recommended POIs, where N is the number of recommendation results. In this paper, temporal influence is added to POI recommendation algorithm, so the overall precision and recall are calculated by averaging the precision and recall values at all time slots, respectively. The precision and recall metrics are calculated as follows:

$$precision = \frac{1}{24} \sum\nolimits_{t \in T} \left(\frac{\sum\nolimits_{u \in U} |T_{u,t} \cap R_{u,t}|}{\sum\nolimits_{u \in U} |R_{u,t}|} \right) \tag{15}$$

$$recall = \frac{1}{24} \sum\nolimits_{t \in T} \left(\frac{\sum\nolimits_{u \in U} |T_{u,t} \cap R_{u,t}|}{\sum\nolimits_{u \in U} |T_{u,t}|} \right) \tag{16}$$

where $T_{u,t}$ denotes the set of corresponding ground truth POIs in the testing data, $R_{u,t}$ is the set of Top-N recommended POIs.

4.3. Recommendation methods

We evaluate six recommendation methods in Table 4 based on user check-in features and temporal features. In Table 4, U denotes the user-based collaborative filtering (CF) method without using any feature. LRT denotes the location recommendation method based on matrix factorization and temporal effects. UTE denotes user-based CF algorithm with time influence proposed in [16]. GTAG-BPP denotes the breadth-first preference propagation algorithm with geographical-temporal influences aware graph [17]. FSUA and CTF-ARA are two algorithms proposed in this paper. Specifically, we use FSUA combined with UTE to evaluate the improvement brought by considering similar user filtering. CTF-ARA is an adaptive POI recommendation algorithm based on check-in and temporal features.

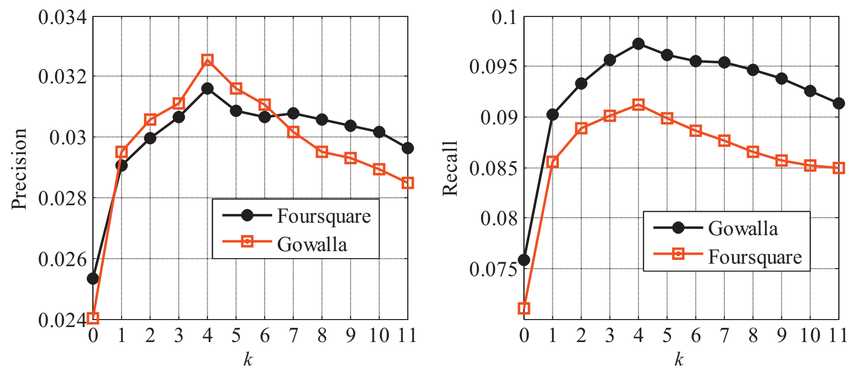
4.4. Experimental results

4.4.1. Impact of time slots' lengths on recommendation

In CTF-ARA algorithm, the parameter k is the absolute value of time difference between time slot t and the adjacent time slot t'. Because the reasonable continuous time slots can reduce the noise of other time slots and improve the accuracy of recommendation, we use $2k+1$ continuous time slots to compute similarity for POI recommendation. Fig. 6 shows the influence of parameter k on Top-5 recommendation for active users on Foursquare and Gowalla datasets ($m_1 = 0$). As shown in Fig. 6, the best precision and recall of CTF-ARA are achieved when $k = 4$ on two datasets. Specifically, on the Gowalla dataset, the precision of CTF-ARA algorithm

Table 4
Recommendation algorithms for comparison.

Algorithms	Check-in feature	Time feature	Description
U	√		User-based CF [16]
LRT		√	Matrix factorization recommendation algorithm [15]
UTE		√	CF with temporal smoothing [16]
UTE+FSUA	√	√	Adding similar user filtering to UTE ([16]+ Section 3.2.2)
GTAG-BPP		√	breadth-first preference propagation algorithm with geographical-temporal influences [17]
CTF-ARA	√	√	Check-in and temporal features based adaptive recommendation algorithm (Section 3.2.4)



(a) Precision (b) Recall

Fig. 6. Influence of parameter k on Top-5 recommendation.

on nine adjacent time slots improves by around 35.49% compared with that only on the target time slot, and the recall improves by around 28.27% at same condition. Therefore, we take $k=4$ to compute similarity in this paper. This means the target time t is the center of nine adjacent time slots. The possible reason is that users are more willing to visit correlative locations in most adjacent time slots, and the performance will decline with the time passing. This is consistent with the analysis of temporal consecutiveness in Section 3.1.4.

4.4.2. Impact of m_1 and m_2 parameters on recommendation

In similar user filtering, parameter m_1 represents the filter threshold for active users, and parameter m_2 represents the filter threshold for inactive users. We tune the values of m_1 and m_2 to compare the performance of CTF-ARA on Foursquare and Gowalla datasets ($k=4$), and the precision and recall for different m_1 and m_2 values are shown in Fig. 7.

As shown in Fig. 7(a), the precision increases with the increasing of m_1 and reaches its peak when $m_1=2$. After that, the precision on two datasets decreases gradually as m_1 increases. The similar result for recall can be obtained in Fig. 7(b). The results indicate that the precision and recall for active users can be improved by using similar user filter mechanism. The reason is that active users have a lot of check-in records and similar users and similar user filtering in CTF-ARA can produce better performance.

For inactive users, we tune m_2 on Foursquare, and plot the Top-N precision and recall of CTF-ARA. As shown in Fig. 7(c) and (d), the precision and recall decrease with the increasing of m_2. This means that filtering can not improve the recommendation performance for inactive users. The reason for this phenomenon is that inactive users have few check-in records. Moreover, filtering mechanism will make the data more sparser, thus degrading the recommendation performance. Therefore, we do not perform filter processing for inactive users.

4.4.3. Experimental results and analysis

We conduct three groups of experiments to compare and analyse the recommendation results on Foursquare and Gowalla. They are experiments of different recommendation methods for active users, inactive users, and all users, respectively.

4.4.3.1. Analysis of active users' experimental results. The first group of experiments is to evaluate the accuracy of different recommendation methods for active users. Fig. 8 shows the Top-5, Top-10, and Top-20 precisions and recalls of six algorithms, respectively. As shown in Fig. 8(a) and (b), the Top-5 precision of CTF-ARA reaches the highest value 3.16% on Foursquare, which is increased by 6.76% and 1.52% respectively compared with UTE and GTAG-BPP. The Top-20 recall of CTF-ARA reaches the highest value 19.18%, which is improved by 4.14% compared with GTAG-BPP. It can be seen from Fig. 8(c) and (d) that the CTF-ARA algorithm still has better precisions and recalls than other algorithms. These results indicate that algorithms incorporating time factor (e.g., LRT, UTE, UTE+FSUA, GTAG-BPP, and CTF-ARA) are better than basic U algorithm, and also illustrate that the CTF-ARA algorithm is superior to other algorithms in terms of precision and recall for active user on both datasets. The reason is that CTF-ARA combines time information, user activity, and similar user filtering together.

It can also be seen from Fig. 8 that UTE+FSUA is superior to UTE algorithm. Take Top-5 for example, the precision of UTE+FSUA increases by 4.73% and 15.44% than UTE on Foursquare and Gowalla, respectively. The reason is that UTE+FSUA is incorporated with similar user filtering processing for active users. Therefore, the similarities of users are improved so as to reduce the influence of dissimilar users on the recommendation, and the precision and recall of UTE+FSUA are improved.

4.4.3.2. Analysis of inactive users' experimental results. The second group of experiments is to evaluate the accuracy of different recommendation methods for inactive users on Foursquare and Gowalla. Fig. 9 shows the Top-N precisions and recalls of different algorithms. As shown in Fig. 9(a) and (b), the recommendation precision of CTF-ARA on Foursquare dataset reaches the highest value 2.13% on Top-5, and the highest recall value is 18.93% on Top-20 of CTF-ARA. The similar results can be seen on Gowalla from Fig. 9(c) and (d), the Top-5 precision of CTF-ARA is 2.75% and the Top-

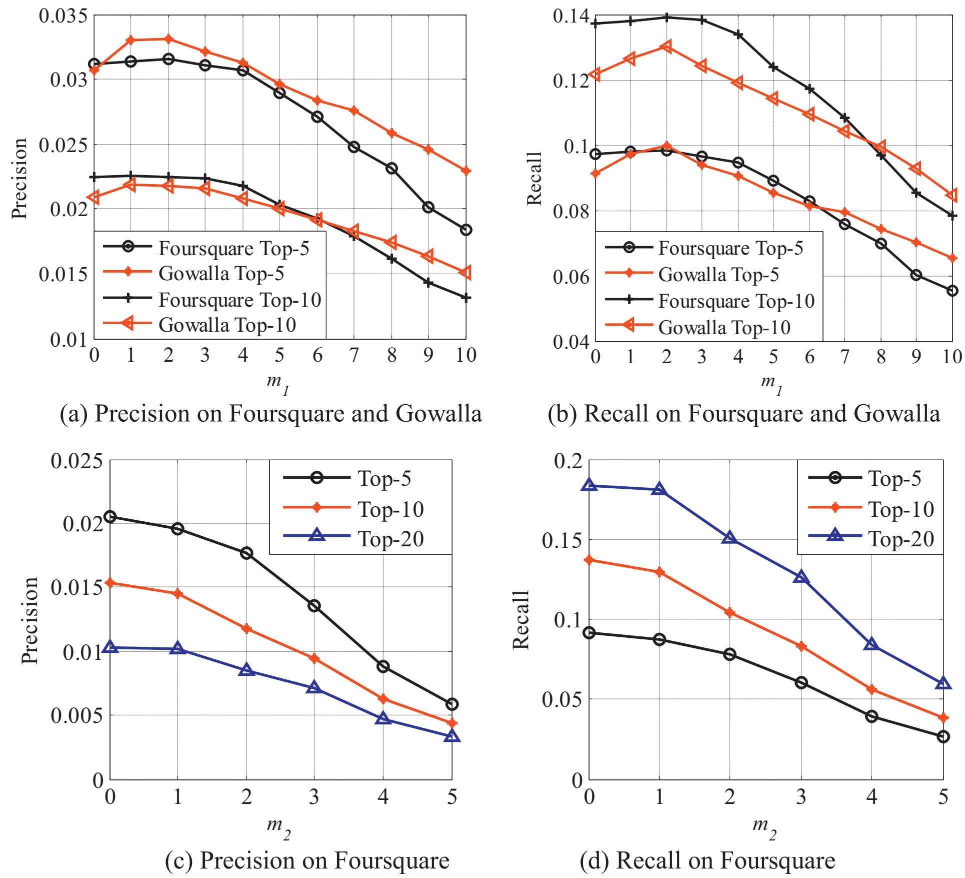


Fig. 7. Influence of m_1 and m_2 on recommendation.

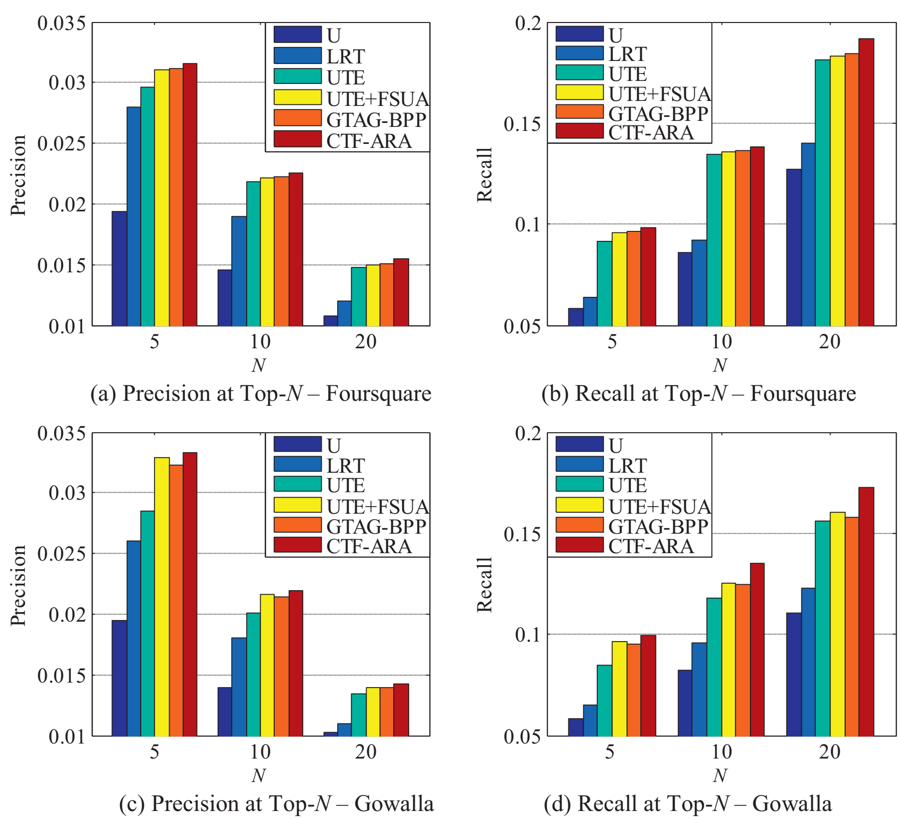


Fig. 8. Performance of recommendation algorithms for active users.

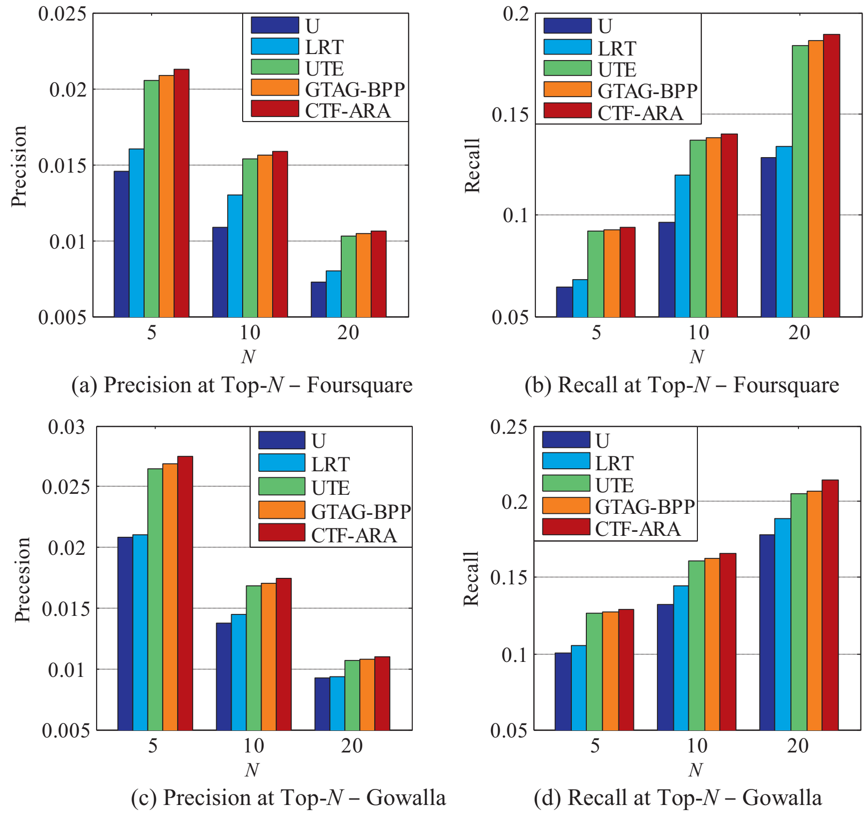


Fig. 9. Performance of recommendation algorithms for inactive users.

20 recall is 21.4%. These results show the superiority of CTF-ARA in POI recommendation for inactive users.

From Figs. 8 and 9, we can see that under the same Top-*N*, the precision of active users is better than that of inactive users for CTF-ARA algorithm on both datasets, which indicates that CTF-ARA is more effective for active users. The reason is that active users have a lot of check-in information. So, the similar user filtering and time slots techniques in CTF-ARA algorithm can improve the recommendation precision significantly. However, the Top-*N* recall values of CTF-ARA are almost the same for active users and inactive users on Foursquare, and the recall of inactive users is higher than that of active users for CTF-ARA on Gowalla. The reason is that both datasets have sparser data, especially for Gowalla, which causes more inactive users and less active users.

4.4.3.3. Analysis of all users' experimental results. The third group of experiments is to evaluate the accuracy of different recommendation methods for all users. Fig. 10 shows the Top-*N* precision and recall of six algorithms on overall Foursquare and Gowalla datasets, respectively. As shown in Fig. 10, the precision and recall values of CTF-ARA are best under the same conditions. Take Gowalla for an example, compared with U, LRT, UTE, UTE+FSUA and GTAG-BPP, Top-5 precision of CTF-ARA increases by 63.27%, 33.27%, 19.36%, 3.12%, and 2.96%, respectively. Top-20 recall of CTF-ARA increases by 54.23%, 23.91%, 18.27%, 7.47%, and 5.8%, respectively. These results indicate that CTF-ARA performs well on two datasets, because POIs are recommended to different users adaptively by the most favorable strategy.

Among the above three group experiments, CTF-ARA always achieves the best results in precision and recall at different Top-*N* on both datasets. The reason is that CTF-ARA is incorporated with check-in and temporal features. Therefore, the personalization and relativity of target users are improved greatly so as to reduce the influence of dissimilar information on the recommendation results, and the accuracy of CTF-ARA is improved.

4.4.4. Comparison of running time

To compare the time consumption of five POI recommendation algorithms, we conduct experiments on two datasets and calculate the running time of five algorithms, respectively. The running time of each recommendation algorithm includes two parts: user similarity calculation time and predictive time. We conduct experiments on Top-5 recommendation for all users, and calculate the user similarity calculation time and predictive time, respectively. Then, we calculate the average user similarity calculation time and the average predictive time for each algorithm. Table 5 shows the comparison of running time for five recommendation algorithms. It needs to be explained that the user similarity calculation time in our CTF-ARA algorithm also includes time for selecting similar users. The time spent in selecting similar users is 6 and 7 milliseconds on Foursquare and Gowalla datasets, respectively.

As shown in Table 5, the average user similarity calculation time of CTF-ARA algorithm is smaller than that of U, LRT, UTE, and GTAG-BPP algorithms on two datasets. Moreover, the average predictive time of CTF-ARA algorithm is also smaller than that of U, LRT, UTE, and GTAG-BPP algorithms on two datasets. This is because GTAG-BPP algorithm adds the geographical factor to temporal computation and U, LRT, and UTE algorithms are executed for all users and all locations. By contrast, CTF-ARA algorithm selects similar users and their visited POIs to compute similarity and make recommendation. Therefore, the running efficiency of CTF-ARA algorithm is the best among five POI recommendation algorithms.

5. Conclusions and future work

In this paper, we present an adaptive method to recommend POIs based on check-in and temporal features. By using proba-

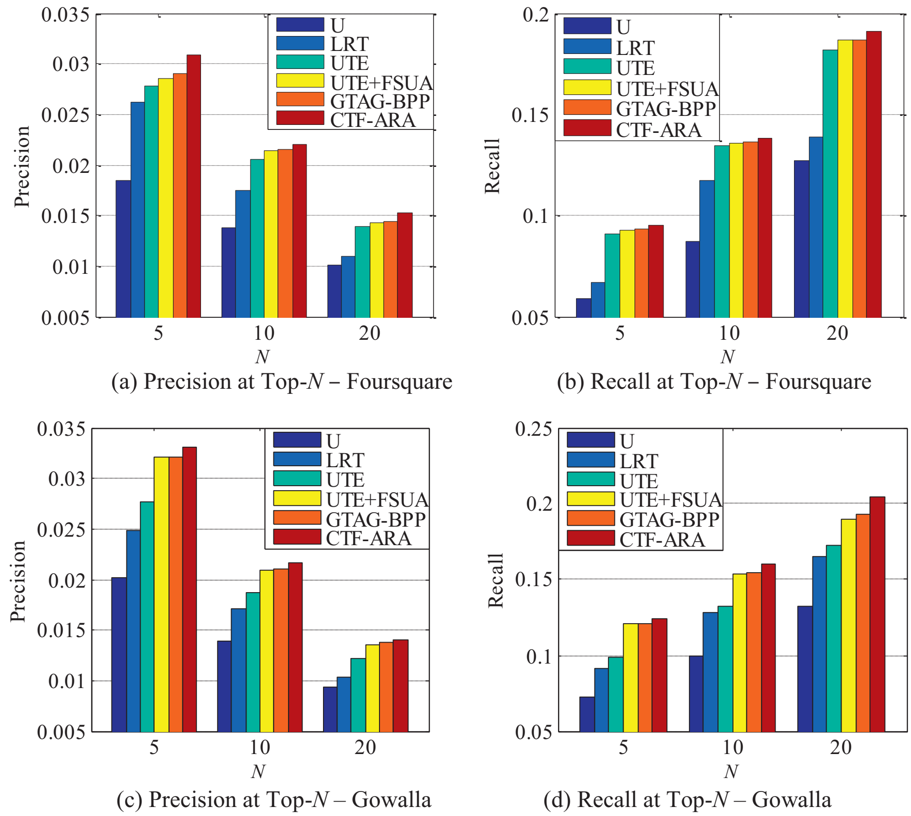


Fig. 10. Performance of recommendation algorithms for all users.

Table 5
Comparison of running time for five POI recommendation algorithms (ms).

Algorithms	The average user similarity calculation time		The average predictive time	
	Foursquare	Gowalla	Foursquare	Gowalla
U	178	197	0.124	0.532
LRT	237	251	0.176	0.657
UTE	196	205	0.143	0.574
GTAG-BPP	274	292	0.215	0.869
CTF-ARA	151	165	0.113	0.516

bility statistics method, we extract four features from historical check-in data in LBSNs, which are user activity, similarity, temporal variability and consecutiveness. Based on the proposed features, we devise an adaptive POI recommendation algorithm according to the users' activities grouped by K-means algorithm. In particular, we filter users with different levels according to user activity and exploit different length of time slots smoothing technology to solve data sparsity issue. The main innovations are realization of adaptivity in our recommendation method by dividing user activity and improvement of recommendation accuracy by combining user check-in and temporal features. Experimental results on two datasets show that the proposed algorithm outperforms the state-of-art recommendation algorithms in terms of precision and recall metrics.

Although the recommendation performance of our method is improved to some extent, there is still room for improvement. The high sparsity of LBSN dataset is the fundamental reason that leads to low POI recommendation performance. In our future work, we will explore the combination of user activity and other factors, such as spatial features, categories information, and social relations. Furthermore, we will develop new POI recommendation algorithms by means of data mining and smoothing technology in order to achieve better recommendation performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.61379116, No.61300193, No.61272466), the Natural Science Foundation of Hebei Province, China (No.F2013203124, No.F2015203046, No.F2015501105).

References

[1] H. Gao, J. Tang, X. Hu, H. Liu, Modeling temporal effects of human mobile behavior on location-based social networks, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 1673–1678.
[2] Q. Yuan, G. Cong, K.Q. Zhao, Z.Y. Ma, A.X. Sun, Who, where, when, and what: a nonparametric Bayesian approach to context-aware recommendation and search for twitter users, ACM Trans. Inf. Syst. 33 (1) (2015) 1–33.
[3] R. Pálovics, A.A. Benczúr, L. Kocsis, T. Kiss, E. Frigó, Exploiting temporal influence in online recommendation, in: Proceedings of the 8th ACM International Conference on Recommender Systems, 2014, pp. 273–280.
[4] S. Zhao, I. King, M.R. Lyu, A survey of point-of-interest recommendation in location-based social networks, Comput. Res. Repository (2016) 1–30 abs/1607.00647.
[5] B. Liu, H. Xiong, Point-of-interest recommendation in location based social networks with topic and location awareness, in: Proceedings of the International Conference on Sustainable Design and Manufacturing, 2013, pp. 396–404.

[6] B. Liu, Y. Fu, Z. Yao, X. Hui, Learning geographical preferences for point-of-interest recommendation, in: Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2013, pp. 1043–1051.
[7] J. Zhang, C. Chow, CoRe: exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations, Inf. Sci. 293 (2015) 163–181.
[8] H. Gao, J. Tang, H. Liu, gSCorr: modeling geo-social correlations for new check--ins on location-based social networks, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 1582–1586.
[9] H. Gao, J. Tang, H. Liu, in: Mobile Location Prediction in Spatio-Temporal Context, 2012, pp. 44–47.
[10] H. Yin, B. Cui, L. Chen, Z. Hu, C. Zhang, Modeling location-based user rating profiles for personalized recommendation, ACM Trans. Knowl. Discovery Data 9 (3) (2015) 19:1–19:41.
[11] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.
[12] M. Chen, F. Li, G. Yu, D. Yang, Extreme learning machine based point-of-interest recommendation in location-based social networks, in: Proceedings of the International Conference on Extreme Learning Machine, 2016, pp. 249–261.
[13] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.
[14] J. Lu, Q. Shambour, Y. Xu, Q. Lin, G. Zhang, A web-based personalized business partner recommendation system using fuzzy semantic techniques, Comput. Intell. 29 (1) (2013) 37–69.
[15] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceedings of the 7th ACM International Conference on Recommender Systems, 2013, pp. 93–100.
[16] Q. Yuan, G. Cong, Z. Ma, A. Sun, N.M. Thalmann, Time-aware point-of-interest recommendation, in: Proceedings of the 36th international ACM International Conference on Research and Development in Information Retrieval (SIGIR), 2013, pp. 363–372.
[17] Q. Yuan, G. Cong, A. Sun, Graph-based point-of-interest recommendation with geographical and temporal influences, in: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, 2014, pp. 659–668.
[18] J.D. Zhang, C.Y. Chow, TICRec: a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations, IEEE Trans. Serv. Comput. 9 (4) (2016) 633–646.
[19] S. Hosseini, L.T. Li, Point-of-interest recommendation using temporal orientations of users and locations, in: Proceedings of the International Conference on Database Systems for Advanced Applications, Springer International Publishing, 2016, pp. 330–347.
[20] S. Zhao, T. Zhao, H. Yang, M.R. Lyu, I. King, STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation, in: Proceedings of the 30th AAAI International Conference on Artificial Intelligence, 2016, pp. 315–321.
[21] C. Cheng, H. Yang, M.R. Lyu, I. King, Where you like to go next: successive point-of-interest recommendation, in: Proceedings of the 23rd International Conference on Artificial Intelligence, 2013, pp. 2605–2611.
[22] W. Wei, X. Zhu, Q. Li, LBSNSim: analyzing and modeling location-based social networks, in: Proceedings of the 33rd IEEE International Conference on Computer (INFOCOM), 2014, pp. 1680–1688.
[23] J.D. Zhang, C.Y. Chow, Y. Li, LORE: exploiting sequential influence for location recommendations, in: Proceedings of the 22nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL), 2014, pp. 1–10.
[24] Y. Mao, P. Yin, W. Lee, D. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in: Proceedings of the 34th ACM SIGIR International Conference on Research and Development in Information Retrieval, 2011, pp. 325–334.
[25] E. Cho, S.A. Myers, J. Leskovec, 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, 2011, pp. 1082–1090.
[26] S. Zhao, I. King, M.R. Lyu, Capturing geographical influence in POI recommendations, in: International Conference on Neural Information Processing, 2013, pp. 530–537.
[27] B. Liu, H. Xiong, S. Papadimitriou, Y.J. Fu, Z.J. Yao, A general geographical probabilistic factor model for point of interest recommendation, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1167–1179.
[28] J.D. Zhang, C.Y. Chow, Point-of-interest recommendations in location-based social networks, Sigspatial Spec. 7 (3) (2016) 26–33.
[29] C. Cheng, H. Yang, I. King, M.R. Lyu, Fused matrix factorization with geographical and social influence in location-based social networks, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence, 2012, pp. 17–23.
[30] H. Li, Y. Ge, R. Hong, H. Zhu, Point-of-Interest recommendations: learning potential check-ins from friends, in: Proceedings of the ACM SIGKDD International Conference, 2016, pp. 975–984.
[31] H. Ma, On measuring social friend interest similarities in recommender systems, in: Proceedings of the 37th ACM International Conference on Research & Development in Information Retrieval, 2014, pp. 465–474.
[32] H.G. Rong, S.X. Huo, C.H. Hu, User similarity-based collaborative filtering recommendation algorithm, J. Commun. 35 (2) (2014) 16–24.