

RESEARCH

Open Access



Friend and POI recommendation based on social trust cluster in location-based social networks

Jinghua Zhu^{1,2†}, Chao Wang¹, Xu Guo¹, Qian Ming¹, Jinbao Li^{1*†} and Yong Liu¹

Abstract

Friend and point-of-interest (POI) recommendation are two primary individual services in location-based social networks (LBSNs). Major social platforms such as Foursquare and Instagram are all capable of recommending friends or POIs to individuals. However, most of these social websites make recommendations only based on similarity, popularity, or geographical influence; social trust among individuals has not been considered in those recommendation system. Recently, trust relationship has been proved to be helpful in collaborative recommendation. In this paper, we first propose algorithm to identify trust clusters and then give a trust prediction method based on these trust clusters. Then we combine the trust value and similarity among individuals to recommend friends to the target user. As for the POI recommendation, we devise a hybrid framework that integrates user preference, geographical influence, and trust relationship to improve the recommendation quality. In order to validate the effectiveness and efficiency of our methods, a series of experiments on two real social networks Foursquare and Instagram are conducted. The experiment results show that the trust cluster-based recommendation approach outperforms the baseline recommendation approaches in precision and recall.

Keywords: LBSN, Friend recommendation, POI recommendation, Collaborative filtering, Trust cluster

1 Introduction

In recent years, social networks have become popular and widespread. There are various of social networks such as location-based social network, mobile social network, vehicular social network, and social sensor cloud system [1–5]. Location-based social networks (LBSNs) are a kind of social networks which include geographical information into shared contents. Many research problems emerge in LBSNs such as information propagation [6–9], privacy, and security [10–14]. Among them, friends and POI recommendation aiming at helping users make friends and find new places are two important services in LBSNs that have attracted a lot of attentions recently [15–18].

However, most of the recommendation methods are based on similarity, popularity, or geographical influence

which are intuitive but consider few of the social trust. Actually, trust relationship has now been proved to be helpful for friends and location recommendation. Collaborative filtering (CF)-based methods use the similarity obtained from the user-location matrix to make recommendations. The basic idea is that user preferences can be inferred according to users who have similar check-in records. But this kind of similarity only reflects the activity similarity among users, not the interest similarity. For example, user A is a tourist and user B is a cook; they both uploaded photos at the same restaurant. The tag of the photo uploaded by A is “travel” and the tag of photo uploaded by B is “food material”. The traditional methods may recommend B to A because they checked in at the same location, but indeed their activities reflected from the photos are different. As for the POI recommendation, a user has higher probability to visit the places recommended by the people he trusts, other than the places recommended by the strangers. It is not necessary for the trusted users to have similar check-in records as the target user. Thus, in addition to check-in similarity, social

*Correspondence: zhujinghua@hlju.edu.cn

†Jinghua Zhu and Jinbao Li contributed equally to this work

¹School of Computer Science and Technology, Heilongjiang University XueFu Road, Harbin 150001, China

Full list of author information is available at the end of the article

trust plays a key role in friends' and POI recommendation. Many trust-based recommendation methods not only can improve the performance of the recommendation but also can guarantee robustness and solve data sparsity and cold start problem.

Figure 1 is an example of LBSNs, showing the check-in activities as well the trust relationships among users. Three different categories of relationships exist in LBSNs: user check-in activities denoted by the directed links from users to locations; trust and distrust relationships among users denoted by directed links from users to users; and geographical links among locations. As can be seen from Fig. 1, u_1 has checked in at l_1, l_2 and l_3 ; l_2 has been visited by u_1 and u_2 ; u_1 trusts in u_3 ; and u_3 trusts in u_4 . Both u_1 and u_4 distrust in u_2 .

We can obtain trust relationship either explicitly from users or implicitly deduce them from users' activities such as follow, review, and rating. Trust predication is the key problem in trust-based recommendation system. Jaccard coefficient or its variants which belongs to memory-based methods are the most popular trust measurement. Model-based approaches utilize both trust and rating by matrix factorization. Data sparsity is the inherent problem of large-scale trust network, and the cost of trust predication via matrix factorization is very high. We present approach to compute trust value by finding co-clusters between trusters and trustees. Co-clusters are groups of both trusters and trustees with similar patterns. Users can trust several user groups and also can be trusted by several user groups according to his social community characteristics. Then, we present FRTC (Friend Recommendation Based on Trust Cluster) algorithm and TSG algorithm for friend and POI recommendation respectively. TSG is a hybrid recommendation algorithm that combines the social trust, user preference similarity, and

geographical influence. To validate the proposed method, several series of experiments on two real LBSN dataset (Foursquare and Instagram) are conducted. The experiments results show that trust relation is helpful for friend and location recommendation; the presented hybrid location recommendation method is considerably significant. Due to that the computation complexity is very low, the algorithms presented in this paper are scalable to the large-scale network.

The remainder of this paper is organized as follows. Section 2 reviews the related work on friend and location recommendation in LBSN. In Section 3, we first introduce the trust co-clustering approach and then introduce the friend recommendation algorithm FRTC by utilizing trust co-clusters. The hybrid POI recommendation algorithm TSG is proposed in Section 4. Several series of experiments are conducted in Section 5 to evaluate the performance of our methods. We give conclusion and future work in Section 6.

2 Related work

2.1 Friend recommendation

The simple way of friend recommendation is according to the number of common friends among users [19]. However, this kind of social graph-based friend recommendation methods may not be appropriate to reflect a user preference in real life. Another way of friend recommendation is depending on similar user preferences. Zhang and Xu [15] designed a friend recommendation framework which can characterize user interest in context and content, as well as combining domain knowledge to improve recommending quality. Lam and Riedl [16] calculated the similarity between users by means of mining users' latent features and proposed a random walking model to get friend recommendation. Guo

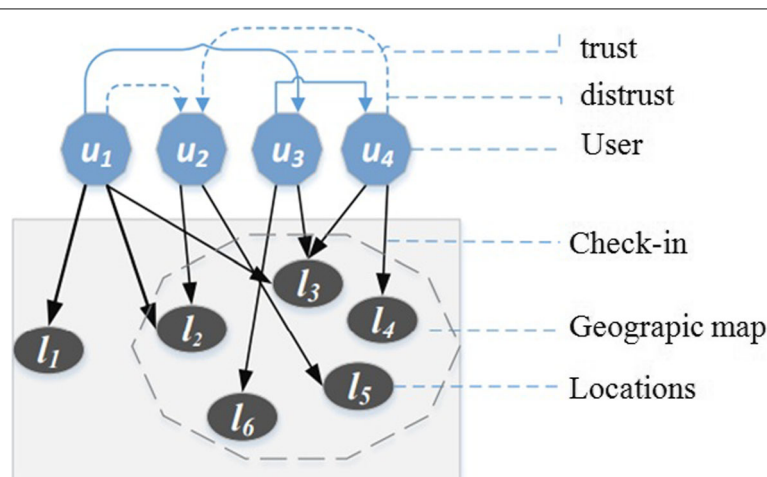


Fig. 1 An example of LBSNs

et al. [17] proposed a semantic-based friend recommendation method to mine friends based on lifestyles instead of social activities. However, the recommendation quality is undesirable because of lack of description of semantic information about user activity preferences.

Social trust has now been considered useful in collaborative recommendation. Guo et al. [18] proposed to recommend friends by considering multiple factors such as user semantic, activity preferences, and social trust. Jamali and Ester [20] designed a light-weighted privacy-preserving friend recommendation scheme by utilizing both social attribute and trust relationship to establish a multi-hop trust chain between strangers. Due to that the number of explicit trust relationship is very small, the trust relation data is usually very sparse.

In this work, we propose to use co-clustering technique for trust value prediction when recommending friends. Co-clustering approaches have been used in the collaborative filtering process. However, most of the research focus on product recommendation [21, 22] or community detection [23]; we are interested in friend recommendation according to trust value prediction with co-clusters.

2.2 POI recommendation

Collaborative filtering (CF) is the most popular recommendation technique, Zhang and Xu [15] used the co-rating-based similarity measurement for predicting ratings of unknown locations. Social influence can also be used for POI recommendation. Wang et al. [24] proposed a trust-based probabilistic recommendation model for social networks. A matrix factorization model was proposed for recommendation which considered trust and distrust relation to improve the recommendation accuracy [25]. Ye et al. [26] designed a power law distribution model to describe geographical influence on user check-in behaviors, and improved the POI recommendation quality by including geographical influence into CF method. Lian et al. [27] proposed a combinative recommendation

framework which integrates the users' physical check-in characters and social influence into the recommendation model. In addition to check-in activities, context information can also be used to improve the recommendation quality [28]. Gao et al. [29] studied the content information such as user preferences, location characteristics, and emotion distinctions in POI recommendation system. Lee and Ma [30] exploit distrust information and investigate their propagation effects. They combine the k -nearest neighbors and the matrix factorization methods to maximize the advantages of both rating and trust information.

In the above work, the social trust is derived directly or indirectly from the trust network; however, the trust network is usually very sparse. We first deal with friend recommendation problem by using co-clustering approach and then propose a hybrid POI recommendation method by incorporating trust, geographical influence, and user preference.

3 Social trust-based friend recommendation method

In this section, we present our trust co-clustering-based friend recommendation algorithm for LBSN.

3.1 Trust network and matrix

Given trust matrix T , the rows correspond to trusters and the columns correspond to trustees. If the (i,j) th element of T takes on the value $t_{ij} = 1$, this indicates that user i trusts user j and user j is trusted by user i . $t_{ij} = 0$ indicates that user i might trust user j or user j might be trusted by user i . Our objective is to recommend friends that the target user is likely to trust.

Example: The trust network is shown in Fig. 2 (The left is the explicit trust and the right is the implicit trust; the solid line indicates the direct trust; the dotted line represents the 2-hop indirect trust). The trust matrix T can be derived from the trust network.

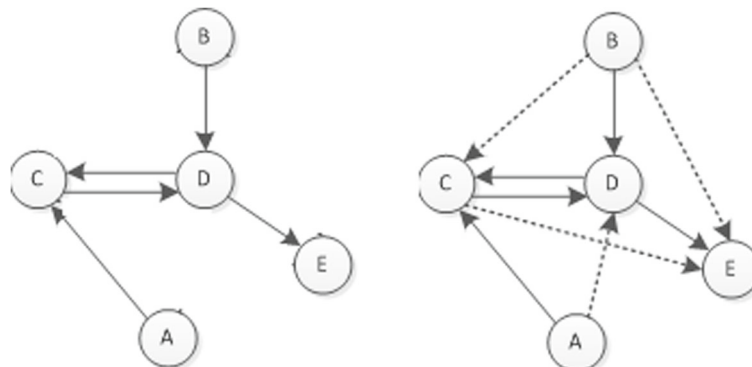


Fig. 2 An example of trust network

$$T = \begin{matrix} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (1)$$

In order to solve the matrix sparsity, we assume that the vertices in the path whose path length does not exceed two are the follow users of the starting vertex in the transit path of the user relationship. For example, in the left part of Fig. 2 one of the paths starting from A is A-C-D-E, which indicates that A trust D and E indirectly. Due to that the path length from A to E is beyond two, E does not serve as the following user of A. We can see from the right part of Fig. 2 that there is no dotted line from A to E. We obtain the implicit trust through the transitive closure of the trust matrix. The implicit trust matrix is as the following:

$$T' = \begin{matrix} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (2)$$

The goal of trust co-clustering is to divide the matrix T' into several clusters. Both the trusters and the trustees can belong to multiple clusters. There is at least one truster and one trustee in one cluster. We will introduce the specific co-clustering method in the following subsection.

3.2 Trust co-clustering algorithm

We choose fuzzy c-means (FCM) as the trust co-clustering algorithm. FCM is a widely used clustering algorithm which uses membership to determine the degree of user affiliation with each cluster. The result of the clustering of trust network is represented by affiliation matrix of users and clusters $A_{n \times c}$; the value of $A_{ij} \in [0, 1]$ denotes the probability that user i belongs to cluster j . The goal of FCM is to minimize the following cost function J :

$$J(A_{ij}, c_j) = \sum_{j=1}^c \sum_{i=1}^n (A_{ij})^p d(x_i, c_j)^2, \text{ where } \sum_{j=1}^c A_{ij} = 1. \quad (3)$$

In the above equation, A_{ij} represents the probability that user i belongs to cluster j . c_j is the center of the j th cluster. p is a free parameter that controls the degree of mixing of different clusters. When $p = 0$, each user belongs to only one cluster; when $p > 0$, each user is allowed to belong to multiple clusters. Here we set $p = 2$. $d(x_i, c_j)$ is the Euclidean distance from the cluster center of the i th user to the j th cluster. We divide the user relationship matrix

R into c fuzzy clusters and find the cluster center of each cluster to minimize the global cost function.

In the initialization step of FCM, the affiliation matrix is initialized with random numbers between (0,1). The sum of the affiliation of each user to all clusters is one. The necessary condition for minimizing the global cost function is as the following:

$$c_j = \frac{\sum_{i=1}^{n+m} (A_{ij})^p x_i}{\sum_{i=1}^n (A_{ij})^p} \quad (4)$$

$$A_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d(x_i, c_j)}{d(x_i, c_k)} \right)^{2/(p-1)}} \quad (5)$$

FCM algorithm generates trust clusters in a way of iterative process. The performance of FCM depends on the initial cluster center, so we run FCM multiple times. Each time, a different cluster center is selected to initialize the algorithm. The algorithm stops when the change of cost function J from the previous iteration is less than the threshold q . The trust aggregation result is shown in Fig. 3 As can be seen, $u_1 \sim u_3$ belong to the same trust cluster. $u_3 \sim u_6$ belong to another trust cluster. One user can belong to multiple clusters, for example, u_3 belongs to two different trust clusters in Fig. 3. u_7 does not belong to any cluster.

The main steps of the FCM algorithm are shown in the Algorithm 1.

The input parameter T of FCM is trust matrix; q is the threshold to control the iteration times. c is the number of co-clusters. A is the affiliation matrix. Lines 1–2 initialize the iteration time and affiliation matrix. Lines 4–13 generate trust clusters in a way of iterative process. In

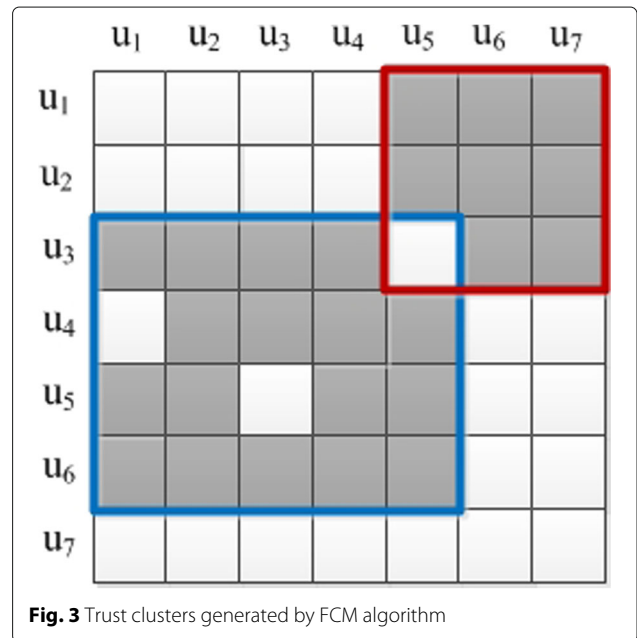


Fig. 3 Trust clusters generated by FCM algorithm

Algorithm 1 Fuzzy-C-Means (T, q, c, A)

```

1:  $t = 1$ 
2:  $A = A^0$ 
3: repeat
4:   for  $j = 1$  to  $c$  do
5:     Compute the center of the  $j$ th cluster  $c_j^t$  according
       to(4)
6:   end for
7:   for  $i = 1$  to  $n$  do
8:     for  $j = 1$  to  $c$  do
9:       Compute Euclidean distance  $d(x_i, c_j^t)$  and  $A_{ij}^t$ 
       according to (5)
10:    end for
11:  end for
12:  Compute cost function  $J^t$  according to (3)
13:   $t++$ 
14: until  $|J^{t-1} - J^{t-2}| \leq q$ 

```

the iterative process, we first compute the cluster center for every cluster and then compute the distance and the affiliation value to each cluster for each user. Finally, we compute the cost function J and the cost change from the previous iteration. The algorithm stops when the cost change less than threshold q .

The time cost of FCM not only depends on the number of samples n and the number of co-clusters c , but also depends on the threshold q . The time cost of lines 4–6 is $O(c)$; the cost of lines 7–10 is $O(cn)$; suppose that the iteration time is m , so the total time cost of FCM is $O(mcn)$.

3.3 Trust value prediction

A_{ij} represents the affiliation degree of user x_i and c_j , and p is a free parameter that controls the degree of mixing of different clusters. When $p = 0$, user only belongs to one cluster; when $p > 0$, user is allowed to belong to multiple clusters. $d(x_i, c_j)$ is the Euclidean distance between the user x_i to cluster c_j . We divide the trust matrix $T_{n \times n}$ into c fuzzy clusters and find the clustering center of each cluster to minimize the global cost function.

Assume that we have K trust co-clusters generated by the clustering algorithm. Two K -dimensional co-cluster affiliation vectors \mathbf{f}_u and \mathbf{f}_v are used to denote the affiliation of truster u and trustee v with co-cluster. The entry values of the vectors are constrained to be nonnegative. $[\mathbf{f}]_c$ denotes the c -th entry of \mathbf{f} . The absolute value of $[\mathbf{f}_u]_c$ corresponds to the affiliation strength of u with co-cluster c ; the larger it is, the stronger the affiliation. $[\mathbf{f}_u]_c = 0$ indicates that truster u has no affiliation with co-cluster c .

If truster u and trustee v appear in the same co-cluster c , co-cluster generates a positive example with probability $1 - e^{-[\mathbf{f}_u]_c[\mathbf{f}_v]_c}$. We assume that co-clusters generate

positive example independently, so $1 - P[t_{uv} = 1] = \prod_c e^{-[\mathbf{f}_u]_c[\mathbf{f}_v]_c} = e^{-\langle \mathbf{f}_u, \mathbf{f}_v \rangle}$; here $\langle \mathbf{f}_u, \mathbf{f}_v \rangle = \sum_c [\mathbf{f}_u]_c [\mathbf{f}_v]_c$ represents the inner product.

Thus, we have the following model:

$$t_{u,v}^+ = P[t_{uv} = 1] = 1 - e^{-\langle \mathbf{f}_u, \mathbf{f}_v \rangle} \tag{6}$$

We find the most likely factors \mathbf{f}_u and \mathbf{f}_v to matrix T by maximizing the likelihood to fit the model parameters.

$$L = \prod_{(u,v):t_{uv}=1} (1 - e^{-\langle \mathbf{f}_u, \mathbf{f}_v \rangle}) \prod_{(u,v):t_{uv}=0} e^{-\langle \mathbf{f}_u, \mathbf{f}_v \rangle} \tag{7}$$

We use single-step projected gradient descent with backtracking search method to solve the above likelihood approximately. \mathbf{f}_u and \mathbf{f}_v are iteratively updated until convergence.

Take Fig. 3 as an example, if we want to predict the trust value from u_3 to u_5 , because there are two trust clusters in the network; the affiliation vectors are \mathbf{f}_3 and \mathbf{f}_5 . After iterative updating, we get $\mathbf{f}_3 = [0.85, 0.15]$ and $\mathbf{f}_5 = [0.77, 0.23]$. Then, the trust value $t_{35} = 0.49$ can be computed according to formula (6) as shown in Fig. 4

3.4 Distrust value prediction

The trust model that integrates distrust is an effective way to deal with user’s dishonesty and malicious attacks, so we consider distrust relationship among users in our trust model. Measuring distrust propagation involves the evaluation of both trust and distrust links. For example, in order to propagate distrust from user u to user v , we first

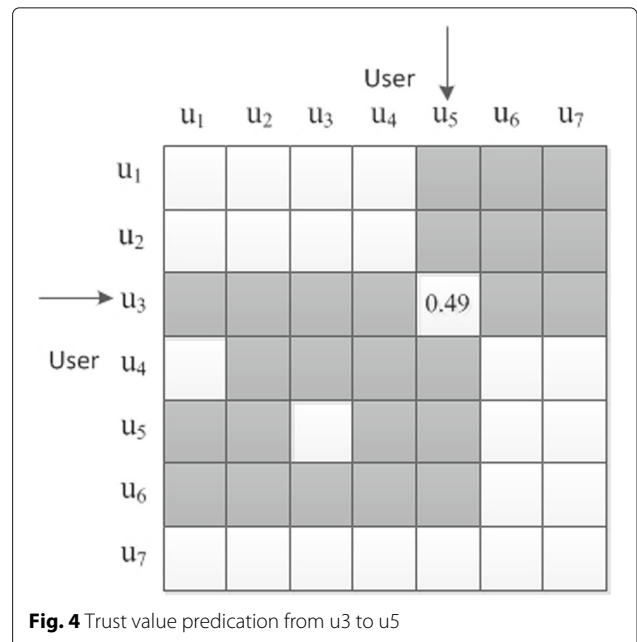


Fig. 4 Trust value prediction from u_3 to u_5

select the qualified paths and then adopt the Jaccard coefficient to calculate the distrust value of each link as below:

$$t_{u,v}^- = \min_{x \in O(u)} t_{u,x}^- \cdot \frac{|O(u^-) \cap I(x^+)|}{|O(u^-)|} - \max_{x \in O(u)} t_{u,x}^- \cdot \frac{|O(u^+) \cap I(x^+)|}{|O(u^+)|} \quad (8)$$

In the above equation, the first term represents the distrust aggregation over all paths that a distrust link followed by a trust link, and the second term represents the trust over all paths that a trust link followed by a trust link. $O(u)$, $O(u^-)$, $O(u^+)$ denote the sets of direct neighbors, trust neighbors, and distrust neighbors of user u , respectively. $I(x^+)$ is the set of users who trust user x . The function \min in the first term provides the minimal value on the links of user u to his distrusted neighbors, and the function \max in the second term provides the maximal value meaning to the highest trust relationship. Thus, a negative $t_{u,v}^-$ enables a distrust propagation from u to v . The final trust value t_{uv} from user u to user v is the difference of the trust value $t_{u,v}^+$ based on co-clusters and the distrust value $t_{u,v}^-$ based on Jaccard coefficient:

$$t_{u,v} = t_{u,v}^+ + t_{u,v}^- \quad (9)$$

3.5 Friend recommendation

Friend recommendation algorithm mainly consists of the computation of user check-in similarity and user trust value. Due to the fact that user check-in history can reflect user preference, our friend algorithm first builds the user-checkin matrix; the matrix value of v_{ij} represents the times that user i has checked in at location j .

$$ULM = \begin{matrix} & L1 & L2 & .. & Lj & .. & Lm \\ \begin{matrix} u_1 \\ u_2 \\ .. \\ u_i \\ .. \\ u_n \end{matrix} & \begin{pmatrix} v_{11} & v_{12} & .. & v_{1j} & .. & v_{1m} \\ v_{21} & v_{22} & .. & v_{2j} & .. & v_{2m} \\ .. & .. & .. & .. & .. & .. \\ v_{i1} & v_{i2} & .. & .. & .. & v_{im} \\ .. & .. & .. & .. & .. & .. \\ v_{n1} & v_{n2} & .. & .. & .. & v_{nm} \end{pmatrix} \end{matrix} \quad (10)$$

We compute the cosine similarity s_{uv} between u and v :

$$s_{u,v} = \frac{\sum_{l_j \in L} c_{u,j} c_{v,j}}{\sqrt{\sum_{l_j \in L} c_{u,j}^2} \cdot \sqrt{\sum_{l_j \in L} c_{v,j}^2}} \quad (11)$$

We can rank the candidate users by combining the trust value and similarity weight in a linear combination way:

$$r_{u,v} = \lambda t_{u,v} + (1 - \lambda) s_{u,v} \quad (12)$$

In the above equation, λ is a number between zero and one which is used to adjust the weight of trust relation and

similarity for friend recommendation. In our experiments, we vary the value of λ to verify the effect of trust relation for friend recommendation. The larger λ indicates that the trust relation has an important role in the friend recommendation; the smaller λ indicates that the similarity has an important role in recommendation. We set the value of λ to be 0.5 which implies that both trust relation and similarity has the same influence in friend recommendation. The main steps of the friends recommendation algorithm is shown in the Algorithm 2.

Algorithm 2 FRTC(u)

- 1: Generate trust clusters using FCM algorithm.
 - 2: **for** all $v! = u$ **do**
 - 3: Compute trust value $t_{u,v}$ from u to v
 - 4: Compute similarity $s_{u,v}$ between u and v
 - 5: Compute the rank value $r_{u,v}$ of the candidate friends
 - 6: **end for**
 - 7: return top-k users as friends to user u .
-

4 Hybrid POI recommendation method

4.1 Similarity-based recommendation

As we know that users who are similar usually have common location interests, thus traditional CF techniques make POI recommendation by inferring the potential location interests of the target user according to his similar users' interests. We use U and L to denote user set and POI set; c_{ij} presents the check-in of user $u_i \in U$ at POI $l_j \in L$. $c_{ij} = 1$ means that u_i has checked in at l_j ; otherwise, we have no record of u_i at l_j . We can utilize the similar users check-in records to infer the target user's location preference. The check-in probability of how likely the user would visit an unvisited location is denoted as $c_{i,j}$:

$$c_{i,j} = \frac{\sum_k s_{i,k} c_{k,j}}{\sum_k s_{i,k}} \quad (13)$$

here $s_{i,k}$ represents similarity between user i and k .

4.2 Geographical influence-based POI recommendation

Usually, near things have closer relations than distant things. On the one hand, people would like to go to nearby locations around their living and work places. On the other hand, it is very possible that people will visit the nearby locations of their interested locations, even if they are far away from them.

We use the naive Bayesian model to infer the probability that people will visit a POI under the geographical influence. Given user u_i and his POI set L_i , the probability of u_i check-in at locations in L_i is decided by the pair-wise distances of POI in L_i as follows:

$$Pr[L_i] = \prod_{l_m, l_n \in L_i \wedge m \neq n} Pr[d(l_m, l_n)] \quad (14)$$

Here, $d(l_m, l_n)$ represents the Euclidean distance between l_m and l_n whose probability is independent and follows power law distribution. Given the candidate POI l_j , the target user u_i and the target user's visited POI set L_i ; the probability for u_i to visit l_j can be computed as the following:

$$Pr[l_j|L_i] = \prod_{l_y \in L_i} Pr[d(l_j, l_y)] \quad (15)$$

4.3 Trust-based POI recommendation

Trust-aware POI recommendation methods have been proved to be able to improve recommendation accuracy, provide robustness for attack, and solve the sparsity and cold start problem.

Under the influence of trust, the likelihood of user u_i visiting l_j can be computed as the following:

$$c_{i,j} = \frac{\sum_{u_k \in F_i} t_{i,k} \cdot c_{k,j}}{\sum_{u_k \in F_i} t_{i,k}} \quad (16)$$

F_i is the trust friends set of user i which can be obtained by the friend recommendation method presented in this paper, $t_{i,k}$ represents the trust value from user i to user k , and $c_{k,j}$ represents the check-in probability of user k at location j .

4.4 Hybrid POI recommendation

In this section, we integrate the user similarity, geographic influence, and trust influence into a hybrid POI recommendation system. $S_{i,j}$ is the check-in probability of user i at location j , and we use $S_{i,j}^s$, $S_{i,j}^g$ and $S_{i,j}^t$ to represent the check-in probability of user i at location j based on user similarity, geographical influence, and trust respectively. Then we combine these three probabilities in a linear way:

$$S_{i,j} = (1 - \alpha - \beta)S_{i,j}^s + \alpha S_{i,j}^g + \beta S_{i,j}^t \quad (17)$$

Here, α and β ($0 \leq \alpha + \beta \leq 1$) are weighting parameters which adjust the relative importance of three different influence factors. $\alpha = 1$ means that check-in probability mainly depends on geographical influence; $\beta = 1$ indicates that probability depends on trust influence.

Check-in probability: In order to estimate the check-in probability $S_{i,j}$, we first compute the check-in probability $S_{i,j}^s$, $S_{i,j}^g$, and $S_{i,j}^t$, respectively. Accordingly, we compute the check-in probability $p_{i,j}^s$, $p_{i,j}^g$ and $p_{i,j}^t$ to obtain $S_{i,j}^s$, $S_{i,j}^g$, and $S_{i,j}^t$. According to the Eq. (13), we have:

$$p_{i,j}^s = \frac{\sum_{u_k} s_{i,k} \times c_{k,j}}{\sum_{u_k} s_{i,k}} \quad (18)$$

Similarly, according to Eq. (15), we have:

$$p_{i,j}^g = Pr[l_j|L_i] = \prod_{l_y \in L_i} Pr[d(l_j, l_y)] \quad (19)$$

Based on the relationship of trust and distrust and the effect of its propagation, we have:

$$p_{i,j}^t = \frac{\sum_{u_k \in F_i} t_{i,k} \times c_{k,j}}{\sum_{u_k \in F_i} t_{i,k}} \quad (20)$$

Finally, the corresponding scores can be computed as the following:

$$S_{i,j}^s = \frac{p_{i,j}^s}{Z_i^s}, \text{ where } Z_i^s = \max_{l_j \in L-L_i} \{p_{i,j}^s\}$$

$$S_{i,j}^g = \frac{p_{i,j}^g}{Z_i^g}, \text{ where } Z_i^g = \max_{l_j \in L-L_i} \{p_{i,j}^g\}$$

$$S_{i,j}^t = \frac{p_{i,j}^t}{Z_i^t}, \text{ where } Z_i^t = \max_{l_j \in L-L_i} \{p_{i,j}^t\}$$

According to the hybrid framework above, the specific steps for generating the recommendation are as follows:

Algorithm 3 Hybrid POI Recommendation (TSG)

- 1: Compute the check-in probability $p_{i,j}^s$ based on user similarity according to (19).
 - 2: Compute the check-in probability $p_{i,j}^g$ based on geographical influence according to (20).
 - 3: Compute the check-in probability $p_{i,j}^t$ based on trust influence according to (21).
 - 4: Linear combine the above three check-in probabilities according to (18).
 - 5: Rank the candidate POIs according to the final check-in probability.
 - 6: Recommend the top-k POIs to users.
-

5 Experiments

In order to reveal the effectiveness of the proposed friend and POI recommendation methods, extensive experiments are designed and conducted to evaluate the performance of the proposed recommendation methods by comparing them with several state-of-the-art recommendation techniques.

Datasets: This paper uses two representative LBSNs Foursquare and Gowalla's datasets to conduct the experiment. The user profile, check-in activities, trust relation, and POI information are all included in the dataset. To reduce the sparsity of data, experiments were conducted on the data of two popular cities, New York and Austin. In addition, the user with less than 10 check-in records and fewer than 5 friends and the POI with less than 5 check-in records were not used in experiments. After processing the dataset, Foursquare's dataset includes 9800 users, 4626 POIs, and 45711 check-in records. Gowalla's

dataset includes 3112 users, 3298 POIs, and 27149 check-in records. Eighty percent data are randomly chosen as the training set, the remaining 20% as test data.

Performance metrics: The friend and POI recommendation algorithms aim to recommend a list of friends and POIs to the target user. We evaluate the performance of our friend recommendation algorithm by comparing it with the existing friend recommendation method. In the POI recommendation algorithm, we calculate the probability of the target user check-in each candidate position, and the POI with the highest score is recommended to the target user. Recall ratio and precision ratio are used to evaluate the friend recommendation quality.

$$Pre@k = \frac{|U_{re} \cap U_{uT@friend}|}{k} \quad (21)$$

$$Recall@k = \frac{|U_{re} \cap U_{uT@friend}|}{|U_{uT@friend}|} \quad (22)$$

where $U_{uT@friend}$ are the real friends of the target user in LBSN.

We use the same performance metrics to evaluate POI recommendation methods. How many POIs previously marked off in the preprocessing step recovered in the returned POI recommendations is meaningful to in our evaluation.

$$Precision@N = \frac{|POI_{recommend} \cap POI_{recovered}|}{N} \quad (23)$$

$$Recall@N = \frac{|POI_{recommend} \cap POI_{recovered}|}{|POI_{recovered}|} \quad (24)$$

where $POI_{recovered}$ are the recovered POIs and $POI_{recommend}$ are the recommend POIs. Precision@N represents the ratio of recovered POIs to the N recommended POIs, and Recall@N represents the ratio of recovered POIs to the set of POIs deleted in preprocessing. We test the performance when $N=5, 10, 20$ with 5 as the default value.

Baseline Algorithms: As for friend recommendation, we evaluated the recommendation performance of our social trust co-cluster-based algorithm (FRTC) by comparing it with the other three baseline friend recommendation algorithms:

(1) Common Friend Recommendation algorithm (CFR) [31] recommends friends to the target user according to the number of common friends.

(2) Preference Similarity Recommendation algorithm (PSR) [32] recommends friends according to user preference similarity.

(3) Friend Recommend based on Preference Coverage (FRPC) algorithm [33] focuses on information coverage of user's preference demands.

In our hybrid POI recommendation algorithm, we consider trust relationship (T), user similarity (S), and geographical influence from POIs (G), so our algorithm is denoted as TSG. In TSG algorithm, we set $0 \leq \alpha + \beta \leq 1$. We run TSG algorithm on Foursquare dataset several rounds with different value of α and β and find that the recommendation quality is the best when $\alpha = 0.1, \beta = 0.2$. And we also compare the performance of TSG with the following existing algorithms.

(1) User-based (U): the traditional CF recommendation algorithm based on user similarity [15]. This is a special case of USG; α and β are set to be zero. That means only user preference similarity is considered for POI recommendation.

(2) Geographic Distance (GD): a special case of USG where $\alpha = 1$ and $\beta = 0$. GD considers the importance of geographical influence in the user's choice of POI [27]. GD recommends POI according to the distance factor.

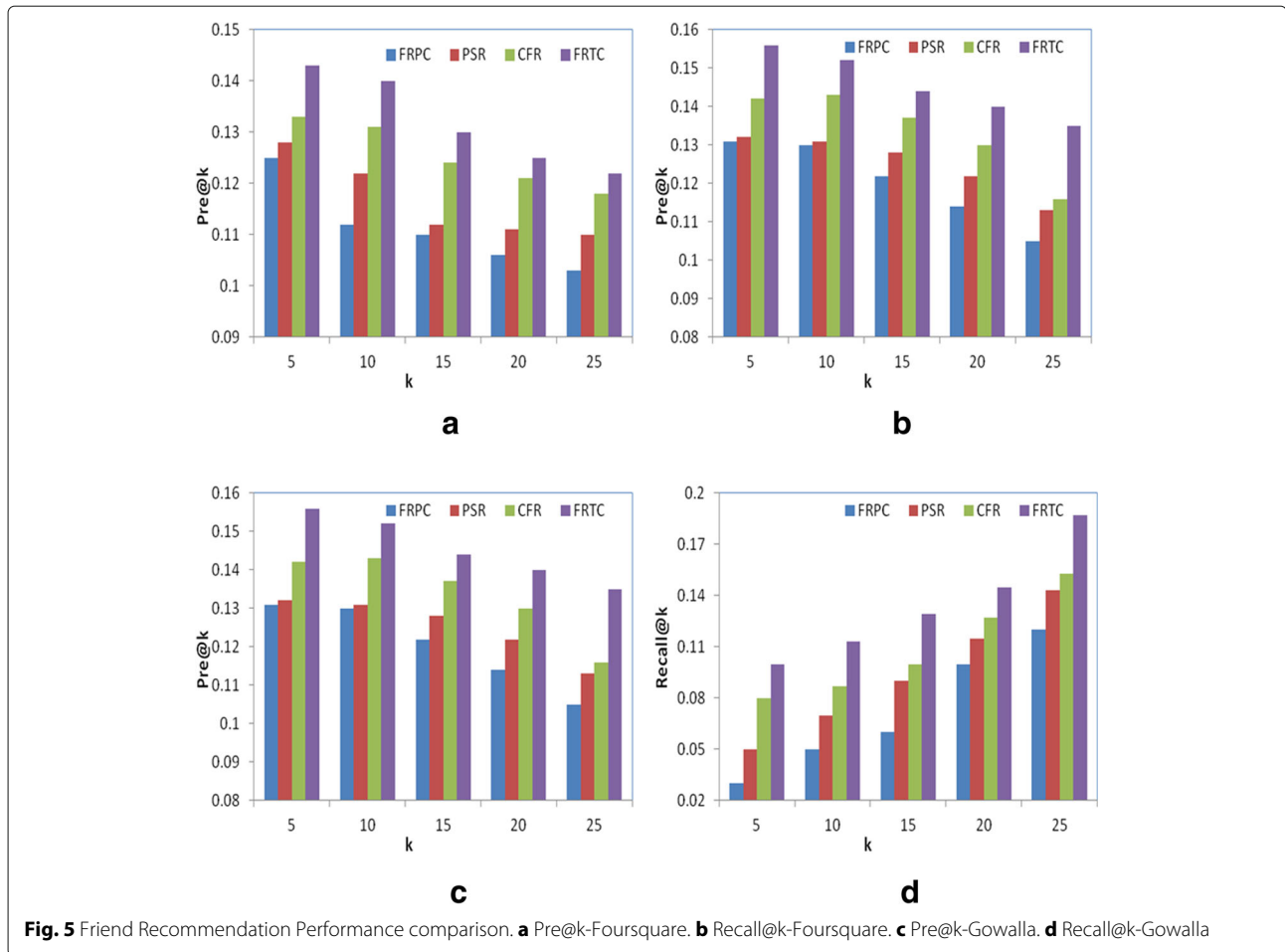
(3) User Similarity and Social trust (USG): algorithm that considers user similarity and trust [30] where $0 < \alpha < 1$ and $\beta = 0$.

Figure 6a–d show the recommendation precision and recall ratio of the four algorithms. Fig. 6a and b show the recommendation precision and recall on Foursquare dataset; Fig. 6c and d show the precision and recall on Gowalla dataset.

6 Results and discussion

We compare our Friend Recommend based on Trust Cluster (FRTC) algorithm with the above algorithms in Foursquare and Gowalla. We randomly select 200 users in the LBSNs as the target users. In Fig. 5 (a)–(d), our algorithm FRTC has the significant advantage in recommendation precision and recall compared with CFR, PSR, and FRPC. The x -axis represents the number of recommended friends and the y -axis represents the precision and recall respectively. CFR makes recommendation by utilizing network topology, PSR and FRPC consider the preference coverage, and none of them utilize the trust relationship among users. Our FRTC algorithm first computes the trust co-clusters of users and then predicts the trust values among users according to these trust co-clusters. Finally, FRTC combines the trust value and user similarity to recommend top-k users to the target user. As can be seen from Fig. 5, the recommendation quality of FRTC is better than the other three algorithms.

We evaluate the POI recommendation performance of our TSG algorithm and compare with the existing three algorithms. As can be seen in Fig. 6, our hybrid recommendation algorithm TSG can get a better performance than the other three algorithms on both of the two datasets. And as N increases, the accuracy of all algorithms decreases and the recall of all algorithms increases. However, TSG algorithm proposed in this paper always



shows the best recommendation quality. This verifies that trust relationship can improve the accuracy and reliability of the recommendation.

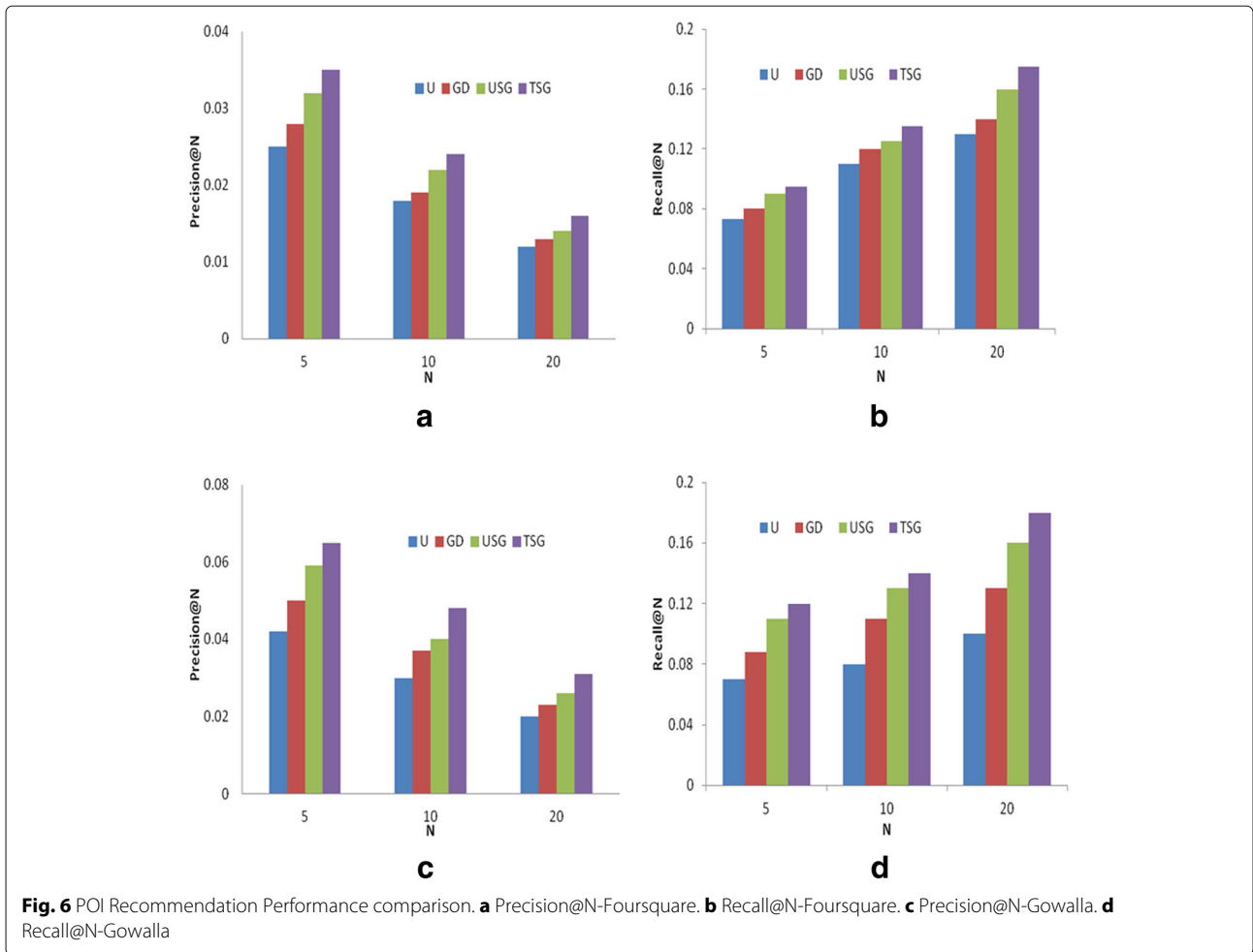
In Fig. 7, we compare the performance of algorithms which consider trust and distrust. TSG only considers trust relationship, and TSG-all considers both trust and distrust relationship at the same time. From Fig. 7, we can see that TSG-all has better accuracy and recall compared with TSG. This implies that distrust can improve recommendation quality. Therefore, more potential user information such as distrust can be mined to improve recommendation accuracy.

Figure 8 compares the running time of the proposed hybrid algorithm with the other recommendation algorithm. As can be seen from Fig. 8, TSG algorithm with trust relationship spends much more time than the traditional recommendation algorithm with single factor in both datasets. It is because that the hybrid recommendation needs to calculate the probability according to similarity, geographic influence, and trust relationship and linearly combines them to obtain the final recommendation result. However, the three recommendation systems are

relatively independent and can run in parallel, so the increase in time overhead is not very large.

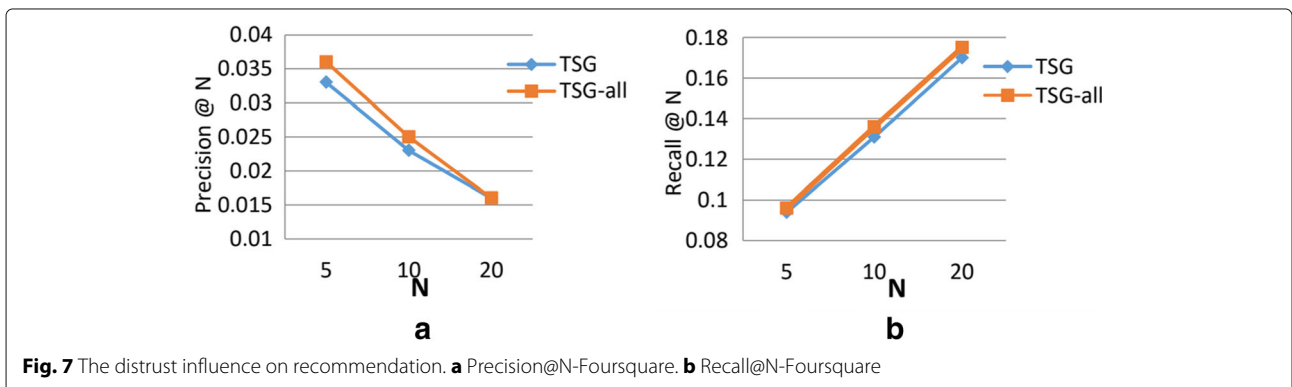
7 Conclusion and future work

This paper studies friend and POI recommendation problem in location-based social networks. Most of the existing recommendation methods do not consider trust relationship among users, and thus, they cannot obtain high-quality recommendation result. In order to improve the recommendation quality as well as time efficiency, in this paper, we study how to utilize trust relationship for friend and POI recommendation. First, we introduce trust co-cluster concept and exploit FCM algorithm to generate trust clusters among users. Then, we predicate the trust values among users in an iterative way. We combine the trust value and similarity among users to recommend friends to the target user. Finally, as for the POI recommendation, we present a hybrid approach that integrates social trust, preference, and geographical influence into the recommendation system. In order to evaluate the performance of our trust-based recommendation method, experiments are conducted on real LBSNs datasets. The



experiment results show that compared with the existing friend and POI recommendation algorithms, our trust co-cluster-based friend recommendation algorithm and hybrid POI recommendation algorithm are more accurate and time efficient.

As for the future work, we plan to incorporate more semantic information into the clustering process to further improve the quality of friend and location recommendations. We will also consider the privacy protection problem in the friend and POI recommendation process.



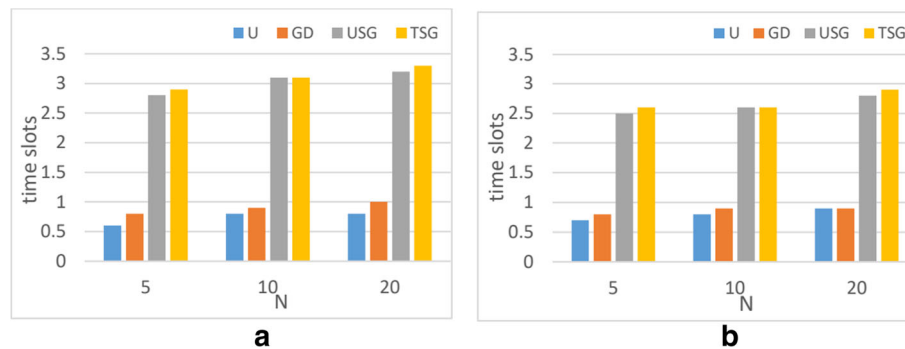


Fig. 8 Time efficiency comparison. **a** Foursquare. **b** Gowalla

Abbreviations

Abbreviations CF: Collaborative filtering; FCM: Fuzzy c-Means; LBSN: Location-based social network; POI: Point of interest

Acknowledgements

The authors thank the reviewers and editors for giving the helpful comments and suggestions, which help to improve the quality of the manuscript.

Funding

This work is supported in part by the National Natural Science Foundation of China (CN) (61602159, 61100048, 61370222), the Natural Science Foundation of Heilongjiang Province (F2016034), the Education Department of Heilongjiang Province (12531498).

Availability data and materials

The data can be made available for scientific purpose only by sending an e-mail to zhujinghua@hlju.edu.cn.

Authors' contributions

JZ and JL constructed the theory. JZ is the main writer of this paper. YL modified the sentences and improved the readability of the manuscript. CW and XG completed the experiment and analyzed the experimental results. They wrote the program of the algorithm and put forward some suggestions for improvement. All authors read and approved the final manuscript.

Authors' information

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Computer Science and Technology, Heilongjiang University XueFu Road, Harbin 150001, China. ²Heilongjiang Province Key Lab of Database and Parallel Computing XueFu Road, Harbin 150001, China.

Received: 11 September 2018 Accepted: 28 February 2019

Published online: 05 April 2019

References

- S. Zhao, I. King, M.R. Lyu, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*. A Survey of Point-of-interest Recommendation in Location-based Social Networks (Springer, New York, 2016)
- H. Zhou, M. Ruan, C. Zhu, V.C.M. Leung, S. Xu, C. Huang, A time-ordered aggregation model-based centrality metric for mobile social networks. *IEEE Access*. **6**(1), 25588–25599 (2018)
- C. Huang, Y. Chen, S. Xu, H. Zhou, The vehicular social network (VSN)-based sharing of downloaded geo data using the credit-based clustering scheme. *IEEE Access*. **6**(1), 2018
- H. Zhou, H. Wang, X. Chen, X. Li, S. Xu, Data offloading techniques through vehicular ad hoc networks: a survey *IEEE Access*. **6**(1), 65250–65259 (2018)
- C. Zhu, V.C.M. Leung, J.J.P.C. Rodrigues, L. Shu, L. Wang, H. Zhou, Social sensor cloud: framework, greenness, issues, and outlook. *IEEE Network*. **32**(5), 100–105 (2018)
- J. Zhu, Y. Liu, X. Yin, et al, A New Structure Hole-based Algorithm For Influence Maximization in Large Online Social Networks. *IEEE Access*. **5**, 23405–23412 (2017)
- Z. He, Z. Cai, Wang X, in *Proceedings of International Conference on Distributed Computing IEEE Systems*. Modeling Propagation Dynamics and Developing Optimized Countermeasures for Rumor Spreading in Online Social Networks, (2015), pp. 205–214
- M. Han, Y. Li, J. Li, et al, Maximizing influence in sensed heterogeneous social network with privacy preservation. *Int. J. Sensor Netw.* **1**(1), 1 (2017)
- J. Li, Z. Cai, M. Yan, Y. Li, in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*. Using Crowdsourced Data in Location-based Social Networks to Explore Influence Maximization, (San Francisco, 2016)
- J. Li, Z. Cai, J. Wang, et al, Truthful Incentive Mechanisms for Geographical Position Conflicting Mobile Crowdsensing Systems. *IEEE Trans. Comput. Soc. Syst.*, 1–11 (2018)
- Z. Cai, Z. He, X. Guan, Y. Li, Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks. *IEEE Trans. Dependable Secure Comput.* **15**(4), 577–590 (2018)
- X. Zheng, Z. Cai, Y. Li, in *IEEE Communications Magazine*. Data Linkage in Smart IoT Systems: A Consideration from Privacy Perspective (ACM, Piscataway, 2017), pp. 51–58
- X. Zheng, Z. Cai, J. Li, H. Gao, in *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*. Location-Privacy-Aware. Review Publication Mechanism for Local Business Service Systems, (Atlanta, 2017), pp. 1–9
- Z. Cai, X. Zheng, A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical Systems. *IEEE Trans. Netw. Sci. Eng.* **PP**(99), 1–1 (2018)
- D. Zhang, C. Xu, in *Web-Age Information Management*. A Collaborative Filtering Recommendation System by Unifying User Similarity and Item Similarity (Springer, Heidelberg, 2012), pp. 175–184
- SK. Lam, J. Riedl, in *International Conference on World Wide Web*. Shilling recommender systems for fun and profit (ACM, New York, 2004), pp. 393–402
- G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowl-Based Syst.* **57**(2), 57–68 (2014)
- G. Guo, in *ACM Conference on Recommender Systems*. Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems, (2013), pp. 451–454
- Y. Zheng, *Location-Based Social Networks: Users*. (Springer, New York, 2011)
- M. Jamali, M. Ester, in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. TrustWalker: a random walk model for combining trust-based and item-based recommendation (ACM, Paris, 2009), pp. 397–406
- C. Cheng, H. Yang, I. King, M. R. Lyu, in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Fused matrix factorization with geographical and social influence in location-based social networks (AAAI Press, Toronto, 2012), pp. 17–23

22. S. Zhao, I. King, M. R. Lyu, in *International Conference on Neural Information Processing*. Capturing geographical influence in poi recommendations (Springer, Berlin, 2013), pp. 530–537
23. B. Liu, Y. Fu, Z. Yao, H. Xiong, in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Learning geographical preferences for point-of-interest recommendation, (Chicago, 2013), pp. 1043–1051
24. Y. Wang, G. Yin, Z. Cai, et al, A trust-based probabilistic recommendation model for social networks. *J. Netw. Comput. Appl.* **55**, 59–67 (2015)
25. R. Forsati, M. Mahdavi, M. Shamsfard, et al., Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans. Inf. Syst.* **32**(4), 17 (2014)
26. M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. Exploiting geographical influence for collaborative point-of-interest recommendation, (Beijing, 2011), pp. 325–334
27. D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, Y. Rui, in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation, (New York, 2014), pp. 831–840
28. Y. Liang, Z. Cai, J. Yu, Q. Han, Y. Li, Deep learning based inference of private information using embedded sensors in smart devices. *IEEE Netw.* **32**(4), 8–14 (2018)
29. H. Gao, J. Tang, X. Hu, H. Liu, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Content-aware point of interest recommendation on location-based social networks (AAAI Press, Texas, 2015), pp. 1721–1727
30. W. P. Lee, C. Y. Ma, *Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks*. (Elsevier Science Publishers B. V., North-Holland, 2016)
31. J. Chen, W. Geyer, C. Dugan, et al, in *Sigchi Conference on Human Factors in Computing Systems*. Make new friends, but keep the old: recommending people on social networking sites (ACM, Boston, 2009), pp. 201–210
32. C. J. Bao, Y. Zheng, M. F. Mokbel, in *International Conference on Advances in Geographic Information Systems*. Location-based and preference-aware recommendation using sparse geo-social networking data (ACM, New York, 2012), pp. 199–208
33. F. Yu, N. Che, Z. Li, et al, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Friend Recommendation Considering Preference Coverage in Location-Based Social Networks (Springer, Cham, 2017)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
