TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/13 pp 453–462
DOI: 10.26599 / TST.2018.9010017
Volume 23, Number 4, August 2018

Location- and Relation-Based Clustering on Privacy-Preserving Social Networks

Dan Yin and Yiran Shen*

Abstract: Graph clustering has a long-standing problem in that it is difficult to identify all the groups of vertices that are cohesively connected along their internal edges but only sparsely connected along their external edges. Apart from structural information in social networks, the quality of the location-information clustering has been improved by identifying clusters in the graph that are closely connected and spatially compact. However, in real-world scenarios, the location information of some users may be unavailable for privacy reasons, which renders existing solutions ineffective. In this paper, we investigate the clustering problem of privacy-preserving social networks, and propose an algorithm that uses a prediction-and-clustering approach. First, the location of each invisible user is predicted with a probability distribution. Then, each user is iteratively assigned to different clusters. The experimental results verify the effectiveness and efficiency of our method, and our proposed algorithm exhibits high scalability on large social networks.

Key words: clustering; location prediction; privacy-preserving; social networks

1 Introduction

As a sophisticated data model, the *graph* plays an important role in a broad range of real-world applications. A graph can capture both entities and their relationships in real-world applications. In a graph, each entity in the real-world application is represented as a vertex and an edge connects two vertices in the graph if two corresponding entities are related in the application. Typical examples of graphs include social networks, biological networks, the Web, communication networks, traffic networks, and citation networks. In recent years, graph analytic becomes a research hotspot. Extensive research efforts have been devoted to mining and querying graph data.

Graph clustering, a key operation in graph analytics, detects cohesively connected vertices in a graph. Graph clustering is widely applied in practical scenarios. For example, in social networks, graph clustering is an important method for community detection[1]. In biological networks, clustered vertices can be used to find protein complexes[2]. Other applications include spam detection on the Web[3], designing communication protocols in wireless sensor networks[4], and finding research groups in citation networks[5]. There are many applications, such as cyber-physical social systems[6].

Graph clustering has been the subject of comprehensive study. Representative work includes the modularity-based[7], density-based[8], division[9], and seed-generation[10] methods. The general framework of a graph-clustering algorithm is as follows. First, the cohesiveness of a group of vertices is measured using a pre-defined metric, i.e., modularity or density. Then, a tuned algorithm is applied to peel vertices from (in the division method) or add vertices to (in the seed-generation method) each group to improve the value of the cohesiveness metric. Finally, the divided groups of vertices with optimized cohesiveness values are returned as clusters.

We note that these methods only utilize information

• Dan Yin and Yiran Shen are with College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China. E-mail: yindan@hrbeu.edu.cn; shenyiran@hrbeu.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2017-08-19; accepted: 2017-09-14

regarding the relationship between the vertices in a graph. However, social networks today can provide far richer private information than simply relations between users[11, 12]. For example, in IM software packages such as QQ and WeChat, each user can publish his location. Location information can also make a difference in graph-clustering results. Intuitively, this is because users located near to each other are more likely to form cohesively connected groups. In Fig. 1, we illustrate 12 users and their positions. All users are divided into two clusters where $c_1 = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$ and $c_2 = \{u_9, u_{10}, u_{11}, u_{12}\}$. We find that although the user group $g_1 = \{u_1, u_2, u_3, u_4\}$ is tightly connected to the user group $g_2 = \{u_5, u_6, u_7, u_8\}$, users in g_1 and g_2 are in different cities. When considering the location information, it is much more reasonable to regard g_1 and g_2 as different clusters. In previous work[13], a series of clustering algorithms, which take into account both relationship and location information, have also been proposed for social networks, verified by the authors in Ref. [14], these algorithms can generate more meaningful clustering results, especially in recommendation[15] and promotion[16] applications.

In the clustering methods developed for social networks, a necessary assumption is that the location information of each user is visible. However, this assumption does not hold in many real-world scenarios. For privacy reasons, the location information of some users may be unavailable[17]. For example, in the IM software packages such as QQ and WeChat, each user can choose whether or not to show his position information. Previous methods cannot cluster users without access to complete location information, which leaves a huge gap in the analysis of social networks having only partial location information.

To overcome this challenge, in this study, we investigate the clustering problem in privacy-preserving social networks, and take both the locations and relations of users into consideration. In privacy-preserving social networks, only a fraction of the user location information is available. For users whose location information is deleted before publishing, we propose a novel prediction algorithm that generates the probabilities of different locations. To improve efficiency, we have designed an effective method for selecting the initial cluster centroid. We also introduce a clustering algorithm that works iteratively. In each iteration, we assign each node to one of k clusters to maximally improve the clustering gain.

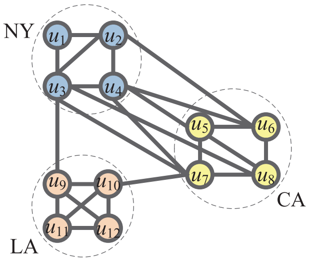


Fig. 1 A clustering example.

The main contributions of this study can be summarized as follows.

(1) The investigation of clustering based on locations and relations in privacy-preserving social networks, which is formally defined as an optimization problem.

(2) The proposal of a probability prediction method with regard to the neighbors' location information, which is effective in predicting the locations of users.

(3) The proposal of a clustering algorithm that maximizes the values of the cluster objective function. An effective initial cluster centroid selection algorithm can help reduce the number of clustering iterations and the clustering strategy can result in faster clustering convergence.

(4) Performance of an extensive evaluation of our proposed clustering approach on real-world data sets, which demonstrates that our method is both efficient and effective. The proposed algorithm also attains high scalability in processing large social networks.

The rest of this paper is organized as follows. In Section 2, we describe related work. In Section 3, we formally define the problem. In Section 4, we propose the clustering algorithms. In Section 5, we describe our experiments. Lastly, in Section 6, we draw our conclusions.

2 Related Work

Graph clustering. Graph clustering is a fundamental operation in graph analytics. The goal of graph clustering is to identify all the groups of vertices that are cohesively connected. Classical solutions for graph clustering are based on link analysis, i.e., they only employ relationship information to identify all clusters. A detailed survey on graph clustering can be found in Ref. [18]. These algorithms often pre-define a metric for measuring the cohesiveness of a group of vertices. Well-known metrics include modularity, as proposed in Ref. [19], density, as proposed in Ref. [20], and the graph cut value, as proposed in Ref. [21]. Based on these metrics, different algorithms have been proposed for grouping vertices to optimize their metric values. The authors in Ref. [22] proposed an algorithm based on multi-way graph partitioning, which divides vertices into different clusters. The authors in Ref. [23] presented an algorithm based on seed generation, where each vertex is iteratively assigned to different clusters based on their distances from different seed vertices. Another category of graph clustering is called spectral clustering[24]. This method utilizes the graph adjacency metric to generate clusters. We note that all these graph-clustering solutions only take relationship information into account in the graph.

Location-based graph clustering. In recent years, research regarding the location privacy problem has attracted the interest of many researchers[25–27]. The location information of users can be obtained by a variety of applications[28–32]. Recent work in clustering social networks has found location information to also be very important. The authors in Ref. [33] proposed a concept called geo-community, which is a group of vertices in a graph that are strongly connected and spatially compact. The authors in Ref. [34] proposed the average linkage (ALK) measure for clustering objects in spatially constrained graphs, and authors in Refs. [35, 36] proposed algorithms for detecting communities with spatial constraints by modularity maximization. In Ref. [37], the authors developed an algorithm for processing large graphs online. However, all of above works make the unreasonable assumption that the location information in the graph is complete. Their techniques cannot handle graphs with incomplete location information in some vertices in the graph.

Privacy-preserving graph analysis. Privacy-preserving graph analysis has become a popular research area in the analysis of graph data[38, 39]. The goal is to query or mine graph data while simultaneously maintaining the privacy of the property and/or structure data of the graph. Privacy-preserving graph analysis has been used to study many problems, i.e., subgraph query[40], vertex reachability[41], inference attacks[42], shortest path[43], spanning tree[44], and graph matching[45]. In this paper, for the first time, we consider the problem of privacy-preserving graph clustering with both relationship and location information.

3 Problem Definition

In this section, we present essential definitions used in clustering problems. First, we define the term of privacy-preserving social networks.

Definition 1 The **privacy-preserving social network** is an undirected graph $G=(V,E,L)$, where $V=V_a\cup V_n$ is the user set satisfying $V_a\cap V_n=\emptyset$, V_a is the set of users whose locations are preserved and V_p is the set of users whose locations are published. L is the locations of V_p, and for $v\in V_p$, $L(v)$ is the location of v. E is the relation between users.

Clustering in privacy-preserving social networks is achieved by partitioning the network G into k disjoined subgraphs $G_i=(V_i,E_i)$, where $V=\sum_{1\leqslant i\leqslant k}V_i$ and $V_i\cap V_j=\emptyset$ for any $i\neq j$. Meanwhile, good clustering in privacy-preserving social networks achieves a good balance between the following two properties: (1) nodes within one subgraph have a high probability of being located at the same place, whereas nodes between clusters are unlikely to be located at the same place; and (2) nodes within one subgraph are close to each other in terms of their social relations, whereas nodes between clusters are connected only loosely to each other.

Definition 2 Cluster location probability. For a subgraph $G_i=(V_i,E_i)$, the cluster location probability of G_i is

$$LocP(G_i)=\max_{l} exp(p_l(v)) \tag{1}$$

where $exp(p_l(v))=\frac{\sum_{v\in V_i}p_l(v)}{|V_i|}$ is the expectation of nodes in location l, $p_l(v)$ is the probability of node v in location l and $|V_i|$ is the number of nodes in G_i. If the location of v is known, there exists $p_l(v)=1$.

Due to the preservation of privacy in some nodes, location information is not complete in social networks. In each cluster, users whose locations are not given may be at different locations. Therefore, we give a probability of $p_l(v)$ to quantify their possible locations. Of all the locations, we choose the one that gives all the users in the cluster the largest sum of $p_l(v)$, which indicates that all the users in the cluster are likely to be at that location.

Definition 3 Cluster density. For a subgraph $G_i=(V_i,E_i)$, the cluster density of G_i is

$$CluD(G_i)=\frac{2|E_i|}{|V_i|(|V_i|-1)} \tag{2}$$

where $|E_i|$ and $|V_i|$ are the number of edges and nodes in G_i, respectively.

There are various ways of measuring the structural closeness of nodes in clusters. Here, we choose a simple method for quantifying the density of clusters, which is the average number of edges in each pair of nodes.

Definition 4 Cluster objective function. For a subgraph $G_i=(V_i,E_i)$, the cluster objective function of

G_i is

$$CluObj(G_i) = LocP(G_i)CluD(G_i) \tag{3}$$

The cluster coefficient formula consists of two aspects: the cluster location probability and cluster density. For each cluster, the nodes with a high location probability should be located at the same place and be closely connected to each other. Also, the values of these two aspects range from 0 to 1. So, it is reasonable to consider them together to determine the clustering results. Therefore, we use the product of these two aspects to represent the quality of the cluster.

Definition 5 The **cluster problem** in privacy-preserving social networks is formally defined as follows:

Input: A privacy-preserving social network G, the number of clusters k;

Output: A partition $G = \{G_1, G_2, \ldots, G_k\}$, where $G_i = (V_i, E_i)$ and $V = \sum_{1 \leqslant i \leqslant k} V_i$ and $V_i \cap V_j = \emptyset$ for any $i \neq j$;

subject to the following:

$$\max \sum_{1 \leqslant i \leqslant k} CluObj(G_i) \tag{4}$$

From the problem definition, we can see that the clustering result is a maximum optimization problem, which maximizes the sum of all the cluster objective functions. To solve the cluster problem, we must address the two main issues:

- Location prediction. The location information for some users is sensitive, so this information is deleted before the social data is published. We must accurately predict the unknown locations of some users to cluster them at similar locations.
- The cluster algorithm. Based on the predicted locations, next, we propose an effective cluster algorithm for maximizing the sum of the cluster coefficients, to identify the nodes in each subgraph as having similar locations and close relations.

We will discuss these two issues in the following sections.

4 Algorithm

In this section, we introduce our location-prediction method for clustering. The location of users can be inferred from the locations of their friends. Thus, we propose a location-prediction algorithm based on published social networks. Then, we propose an effective clustering algorithm with two aspects, locations, and relations, to insure that nodes within the same clusters have similar locations and strong connections.

4.1 Location prediction

As shown in Definition 1, for $\forall v \in V_p$ and $\forall l \in L$, there exists $p_l(v) = 0$ or 1. This means that the location of v is certain. On the other hand, for $\forall v \in V_a$ and $\forall l \in L$, the probability of v in location l satisfies $0 \leqslant p_l(v) \leqslant 1$. Thus, the locations of these protected nodes are uncertain. The users have specific probabilities of being at different places. Next, we give two propositions for nodes.

Proposition 1 For $\forall v \in V_p$, $\exists l \in L$, there exists $p_l(v) = 1$, and for $\forall t \in L \backslash l$, there exists $p_l(v) = 0$.

Proposition 2 For $\forall v \in V_a$ and $\forall l \in L$, there exist $0 \leqslant p_l(v) \leqslant 1$ and $\sum_{l \in L} p_l(v) = 1$.

Now, we give the explanation of the main idea on predicting locations for the nodes in V_a. As we know, in social networks, users who are closely connected tend to live near each other or attend the same offline activities. On this basis, we can deduce that the locations of users are related to that of their friends. Users who have relations with each other are likely to be at the same locations, since they are likely to be not only friends in social networks, but may also attend the same events.

In addition, the locations of users are not merely determined by their neighbors. Sometimes, multiple-hop neighbors can also have an effect on the user locations. For example, messages can be propagated through networks. If users learn from their friends that there is an activity somewhere, they may attend that activity. However, this does not necessarily indicate that their friends are at that location; they may simply received the information from their friends.

Next, we present our location-prediction algorithm as shown in Algorithm 1.

This algorithm initializes the list T for recording the location numbers of neighbors and sum for storing the number of neighbors (Line 1). Then for each node v whose location information is published, we set the probability of the location of v to 1 and the probabilities of other locations to 0 (Lines 2–5). For each node v whose location information is preserved, we compute the probability of its locations through the location of its neighbors and 2-hop neighbors (Lines 6–11). Finally, the probability $p_l(v)$ is returned (Line 12). The time complexity of Algorithm 1 is $O(|V|^2)$.

4.2 Clustering algorithm

In our clustering framework, a privacy-preserving social network G is partitioned with respect to both locations and relations. In this aspect, we address the challenges inherent in the clustering process by answering a number of

Algorithm 1 Location prediction algorithm

```
Input: a privacy-preserving social network $G = (V, E, L)$
Output: the probability $p_l(v)$
Initialize list $T$, $sum$
for each $v \in V_p$ do
    $p_{L(v)}(v) = 1$
    for each $l \in L \backslash L(v)$ do
        $p_{L(v)}(v) = 0$
    end for
end for
for each $v \in V_a$ do
    for $v$'s neighbour $u$ in $V_p$ do
        $T.L(u)$++, $sum$++
    end for
    for $v$'s 2-hop neighbour $u$ in $V_p$ do
        $T.L(u)$++, $sum$++
    end for
    $p_l(v) = \frac{T.l}{sum}$
end for
return $p_l(v)$
```

questions. Based on our answers to these questions, we propose a heuristic clustering algorithm for a privacy-preserving social network. In the clustering algorithm, we need to balance the relationship between cluster location probability and cluster density. As such, we must design a method that ensures that the nodes within one cluster have high location probability and density. There are two challenges for the clustering algorithm as follows:

• Selection of the initial k cluster centroids is important for the iteration. A reasonable selection method can ensure fast convergence of the iteration process. As such, we must design a strategy for selecting k initial centroids for the algorithms.

• Method for assigning nodes to different clusters. In traditional clustering methods, distance measures are used to compute the distances between nodes. In this paper, however, there is no quantifiable values for the distance measures. Determining a way to effectively partition nodes into different clusters is vital for solving this problem.

We propose two algorithms to tackle the clustering problems described above.

First, we propose an algorithm for the selection of the initial centroids. The main concept of this algorithm is as follows: We rank the locations in L to construct a max heap, according to their frequencies. Then, we select a node from V_p that is located at the top of the heap for insertion into the centroid set. It is likely that there are many nodes sharing the same location. We choose the one having the largest degree as the centroid. Then we delete the top of the heap and update the heap. Similarly, we select the second centroid in accordance with the procedure for selecting the first centroid, albeit with a slight difference. This time, we choose the node that is located farthest from the existing centroids, rather than the largest degree.

The initial centroid selection algorithm is presented in Algorithm 2. First, we initialize the variables used in the algorithm (Line 1), which takes $O(1)$ time. Then, we compute a certain number of users at each location on the published data set (Lines 2 and 3). The time cost of this process is $O(|V|)$. Next, we choose the first centroid by selecting the node whose location occurs most frequently. At the same time, this node is also the one with the largest degree (Lines 4 and 5), which consumes $O(|L|)$ time. Then, we construct a max heap h based on the location frequencies (Line 6), which requires $O(|L|)$ time. Next, we iteratively select the $k-1$ centroids. In each iteration, we select the top l of the heap h, and for each node v whose location is l, we compute the shortest paths between a and the other nodes in centroid set C. In this process, we also record the length of the maximum paths, and insert the node into C. Then we delete l from h and update h (Lines 7–15). In each iteration, we select l from top of the heap, which requires $O(1)$ time (Line 8). Then, it takes $O(|E|)$ time to compute the shortest path between v and u, which costs a total of $O(k|V||E|)$ time (Line 13). Then, it costs

Algorithm 2 Initial centroids selection algorithm

```
Input: a privacy-preserving social network $G = (V, E, L)$, the number
       of clusters $k$, the probability of node locations $p_l(v)$
Output: initial $k$ centroids $v_1, v_2, \ldots, v_k$
 1: Initialize the max heap $h$, the frequency list of locations $f$, the
    centroid set $C$
 2: for each $v \in V_p$ do
 3:     $f.L(v)$++
 4: end for
 5: Delete the largest location $l$ from $f$
 6: Select $v$ from $V_p$ satisfying $L(v) = l$ and $v$ has the largest degree
 7: Construct $h$ based on $f$
 8: for $|C| = 1$; $|C| < k$; $|C|$++ do
 9:     Select $l$ from the heap top
10:     for each $v \in V_p$ and $L(v) = l$ do
11:         for each $u \in C$ do
12:             Compute the length $sp$ of the shortest path between $v$ and
                $u$
13:             if $sp > max$ then
14:                 $max = sp$, candidate = $v$
15:             end if
16:         end for
17:     end for
18:     Insert candidate into $C$
19:     Delete $l$ and update $h$
20: end for
21: return $C$
```

$O(1)$ time to insert new centroid into C and $O(|L|)$ time to update h. There are $k-1$ iterations, which finally, returns C (Line 16). To summerize, the time complexity of Algorithm 2 is $O(k|L|+k^2|V||E|)$.

Next, we introduce the clustering algorithm. After selecting the initial k centroids, we arrange the other nodes into different clusters. In each iteration, we assume that node v is put into the cluster G_i, then we compute the cluster gain by adding v into G_i. We select the cluster that has the largest cluster gain after adding v, which is therefore the cluster into which we should put v. Then, we cluster all the nodes in the network. Lastly, we select k centroids from the clusters as the new centroid set. The iteration continues until two adjacent clusters do not have much difference in their optimal function.

Algorithm 3 shows the main procedure of the clustering algorithm. First, we initialize the k centroids as seeds of the k clusters and set the cluster objective function of each cluster to 0 (Line 1), which takes $O(k)$ time. Then, we cluster the networks into k clusters (Lines 2–14). For each node v that has not been put into any cluster, we assume that v is put into cluster G_i, and compute the cluster objective function of G_i. For all k clusters, we select the cluster by adding the v that leads to the largest cluster gain (Lines 3–9). This process takes $O(|V|^2)$ time. Then, we compute the sum of the cluster objective functions (Line 10), which requires $O(k|V|)$ time. Next, we select new centroids from the clusters. For each cluster, we select the node that has a similar location to that of most of the nodes in that cluster as well as a large degree (Lines 11–13), which costs $O(k|V|)$ time. In each iteration, the time complexity is $O(|V|^2+2k|V|)$. The iteration continues until the optimal functions show no obvious change (Line 14). Finally, the clusters are returned (Line 15). Here, we assume the iteration time is t, then the time complexity of Algorithm 3 is $O(t|V|(|V|+2k)$. Since $2k << |V|$, the time complexity of Algorithm 3 is $O(t|V|^2)$.

Algorithm 3 Clustering algorithm

Input: a privacy-preserving social network $G=(V,E,L)$, the number of clusters k, the probability of node locations $p_l(v)$, the initial centroids C

Output: k clusters $G_1,G_2,\ldots,G_k$

```
1: Initialize $k$ centroids to $G_1,G_2,\ldots,G_k$, $CluObj(G_i)=0$, $V_r=V\backslash C$
2: repeat
3:     for each $v \in V_r$ do
4:         for each $G_i$ do
5:             Compute $CluObj'(G_i)=LocP(G_i\cup v)CluD(G_i\cup v)$
6:             if $CluObj'(G_i)-CluObj(G_i)>max$ then
7:                 $candidate=i$
8:             end if
9:         end for
10:        Insert $v$ into cluster $candidate$
11:        Delete $v$ from $V_r$
12:    end for
13:    Compute $\sum_{1\leqslant i\leqslant k} CluObj'(G_i)$
14:    for each $G_i$ do
15:        $t=target_l \max_{v\in G_i} exp(p_l(v))$
16:        Select the node with the largest $p_t(v)$ and degree as the next centroid
17:    end for
18: until $|CluObj'(G_i)-CluObj(G_i)|<\delta$
19: return $G_1,G_2,\ldots,G_k$
```

5 Experimental Study

In this section, we describe an extensive series of experiments we performed to evaluate the performance of our algorithm on real graph datasets. All experiments were conducted on a 2.8 GHz Intel PC with 4 GB main memory, running Windows 7. All algorithms were implemented in Java and compiled using a Myeclipse 8.0 compiler.

5.1 Datasets

Sina microblog datasets. We crawled a dataset from the Sina microblog. By extracting the personal information of users, we obtained location information. The dataset is a network of 2162 users, 77891 followers, and 72367 fans. By analyzing the dataset, we found that 69% of users provided their location details. We established four network subsets to test the results of the algorithms, which are shown in Table 1.

NLPIR microblog datasets. From the Natural Language Processing and Information Retrieval (NLPIR) microblog corpus, which can be downloaded from http://www.nlpir.org/, we extracted a dataset with 9940 users and 107 979 relations. To test the performances of the algorithms on datasets of different sizes, we constructed four dataset subsets. We assigned locations to these users from a specific location set. Table 2 shows the numbers of users and relations.

5.2 Effectiveness evaluation

Accuracy of location prediction. We evaluated the effectiveness of the location-prediction algorithm on the

Table 1 Datasets of Sina microblog.

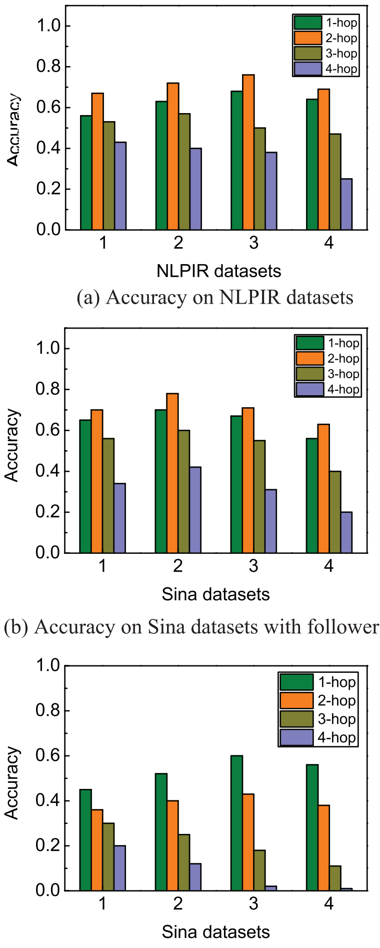
ID	Users	Followers	Fans
1	500	28283	16115
2	1000	54520	29220
3	1500	76732	49331
4	2000	102117	69012

Table 2 Datasets of NLPIR microblog.

ID	Users	Relations
1	2748	12121
2	4025	61881
3	5194	77567
4	9940	107979

two types of datasets. Figure 2 is a comparison of the accuracies for different numbers of hops between neighbors on the NLPIR and Sina datasets. We took into account the different number of hops between neighbors when predicting the locations of users.

Figure 2a shows the results of our accuracy evaluations on NPLIR datasets. The x-axis represents four data subsets with increasing size, and the y-axis represents the accuracy percentage of the prediction algorithm. From the results, we can see that the accuracy increases with the size of the dataset. This trend persists from dataset 1 to 3. The accuracy reaches a peak on dataset 3 of the NPLIR, where it reaches almost 0.77 with a 2-hop selection. As shown in the figure, the algorithm with 2-hop neighbors yields the highest accuracy, followed by the 1-hop selection. In contrast, 3-hop and 4-hop selections are much less accurate in the four datasets.



(a) Accuracy on NLPIR datasets

(b) Accuracy on Sina datasets with followers

(c) Accuracy on Sina datasets with fans

Fig. 2 Accuracy of location prediction.

When we tested the accuracy of location prediction on the Sina datasets, we define both followers and fans as user neighbors. Figures 2b and 2c show the experimental results. As shown in Fig. 2b, when taking the followers as user neighbors, the 1-hop and 2-hop selections achieve the best accuracy, followed by the 3-hop selection. The 4-hop selection performs worst. We can see that the prediction accuracy on the second Sina dataset is the highest, with 0.79 in the 2-hop neighbor selection. The prediction accuracy increases from datasets 1 to 2 and then begins to decrease from datasets 2 to 4. From the figure, we can see that more hops lead to poorer prediction accuracy. This is because friends with only distant relations may have little in common.

Figure 2c shows the prediction accuracy results when considering user fans, in which we can see that the accuracy is much lower than in Fig. 2b. The best accuracy is close to 0.6 in the third dataset with the 1-hop selection. In contract with the previous two figures, the 1-hop selection performs best here. This is because in most instances, the locations of fans may have nothing to do with the users, which can result in low prediction accuracy. Thus, the accuracy of 3-hop and 4-hop selections drops dramatically. When the size of dataset increases, the accuracy of the 4-hop selection is very low.

Cluster quality. In our experiments, we used density and objective function values to evaluate the quality of the clusters. The objective function values are given in Definition 5, and the density is defined as follows:

$$density(\{V_i\}_{i=1}^{k}) = \sum_{i=1}^{k} \frac{|\{(v_p,v_q)|v_p,v_q \in V_i,(v_p,v_q)\in E\}|}{|E|} \tag{5}$$

Figure 3 shows a comparison of the cluster density of datasets of different sizes when we set $k = 6,8,10,12$, where the x-axis represents the k values and the y-axis the density. When k increases from 6 to 12, the cluster density of the first dataset decreases from 0.48 to 0.2. This is because the size of the dataset is small, and increasing the number of clusters makes the clusters smaller. Partitioning the networks into more clusters leads to the nodes within the cluster not being well connected, so the density decreases. In contrast, the cluster density of the fourth dataset increases when the value of k increases. Since the network is large, the nodes can be closely connected when

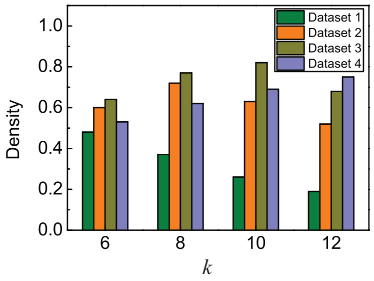


Fig. 3 Cluster density comparison in NPLIR datasets.

the number of clusters is small, whereas, when the number of clusters becomes large, the nodes can be partitioned into more groups, which can make the nodes more cohesive.

Figure 4 shows a comparison of the cluster objective functions for dataset of different sizes when we set $k = 6, 8, 10, 12$, where the x-axis represents the k values and the y-axis the objective function values. As shown in the figure, the objective function values increase as the value of k increases in all the datasets. The objective function values in the second dataset are larger than those in the other datasets. When $k = 12$, this value reaches about 0.78, due to the high density of the network and the high accuracy of the location prediction on the network. The clustering results on the third dataset follow those of the second. When $k = 12$, the function value reaches about 0.63.

5.3 Efficiency evaluation

In this experiment, we tested the efficiency of the proposed algorithms. Figure 5 shows the runtime for two methods on the NLPIR datasets. The Cen-Clu method uses the initial cluster centroid algorithm to choose the initial cluster centroids and the Ran-Clu method randomly selects the initial cluster centroids. Figures 5a and 5b show the runtimes when $k = 8$ and 10, respectively. As we can see in the two figures, the initial cluster centroid selection algorithm can reduce the clustering runtime on the four datasets. When the size of the datasets increases, the runtime increases. In Fig. 5a, the runtime on the selection of the initial cluster centroid of the first three datasets is less than 70 s. When the size of the dataset increases to about 117 919 (the fourth dataset), the algorithm requires about 220 s to cluster the network. In Fig. 5b, the runtime for the selection of the initial cluster centroid in the first three datasets is less than 160 s. The algorithm takes about 350 s on the largest dataset, which shows the good scalability of the algorithm.

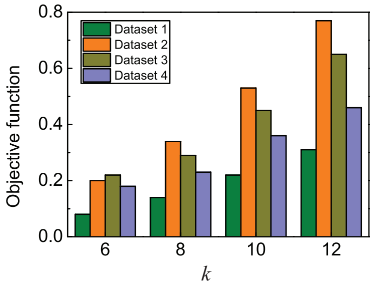
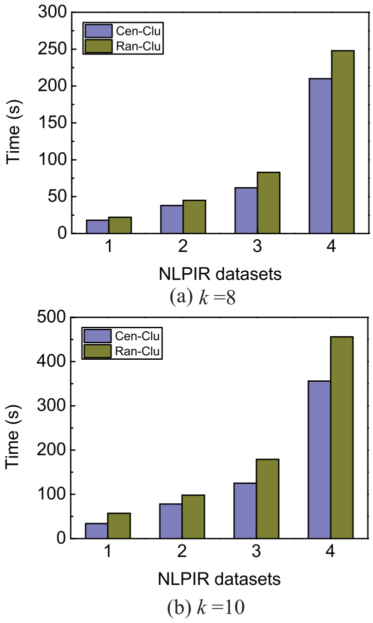


Fig. 4 Cluster objective function in NPLIR datasets.



(a) $k = 8$

(b) $k = 10$

Fig. 5 Clustering efficiency.

6 Conclusion

In this paper, we investigated the clustering problem in privacy-preserving social networks. In this setting, the location information of only a fraction of users is visible for reasons of privacy. We proposed an algorithm to tackle this problem that uses two key techniques: a novel approach using probability to predict the location of each invisible user and an iterative method by which each user is assigned to the appropriate cluster. We verified the effectiveness and efficiency of the proposed algorithm in a series of extensive experiments. Moreover, our algorithm is scalable for processing large social networks. From the figure, we can see that the density of the second dataset grows from $k = 6$ to 10 and then decreases. The peak for the second dataset is about 0.8 when $k = 10$. Similarly, the density of the third data rises from $k = 6$ to 8 and then begins to decrease, having reached its peak of almost 0.72 at $k = 8$.

Acknowledgment

This work was partially supported by the National

Natural Science Foundation of China (Nos. 61602129, 61702132, and 61702133), the Natural Science Foundation of Heilongjiang Province (Nos. QC2017069 and QC2017071), Fundamental Research Funds for the Central Universities (Nos. HEUCFJ170602 and HEUCFJ160601), the China Postdoctoral Science Foundation (No. 166875), and Heilongjiang Postdoctoral Fund (No. LBH-Z16042).

References

[1] S. Fortunato, Community detection in graphs, *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[2] A. D. King, N. Pržulj, and I. Jurisica, Protein complex prediction via cost-based clustering, *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, 2004.

[3] M. Sasaki and H. Shinnou, Spam detection using text clustering, in *Proc. 2005 Int. Conf. Cyberworlds*, Singapore, 2005, p. 4.

[4] S. Bandyopadhyay and E. J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in *Proc. 22^{th} Annu. Joint Conf. IEEE Computer and Communications Societies*, San Francisco, CA, USA, 2003, pp. 1713–1723.

[5] H. D. White, B. Wellman, and N. Nazer, Does citation reflect social structure?: Longitudinal evidence from the "Globenet" interdisciplinary research group, *J. Am. Soc. Inf. Sci. Technol.*, vol. 55, no. 2, pp. 111–126, 2004.

[6] X. Zheng, Z. P. Cai, J. G. Yu, C. K. Wang, and Y. S. Li, Follow but no track: Privacy preserved profile publishing in cyber-physical social systems, *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1868–1878, 2017.

[7] H. Shiokawa, Y. Fujiwara, and M. Onizuka, Fast algorithm for modularity-based graph clustering, in *Proc. 27^{th} AAAI Conf. on Artificial Intelligence*, Bellevue, DC, USA, 2013, pp. 1170–1176.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. W. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proc. 2^{nd} Int. Conf. Knowledge Discovery and Data Mining*, Portland, OR, USA, 1996, pp. 226–231.

[9] M. C. V. Nascimento and A. C. P. L. F. De Carvalho, Spectral methods for graph clustering—A survey, *Eur. J. Oper. Res.*, vol. 211, no. 2, pp. 221–231, 2011.

[10] W. Chen, Y. F. Yuan, and L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in *Proc. 2010 IEEE Int. Conf. Data Mining*, Sydney, Australia, 2010, pp. 88–97.

[11] L. C. Zhang, Z. P. Cai, and X. M. Wang, FakeMask: A novel privacy preserving approach for smartphones, *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 2, pp. 335–348, 2016.

[12] M. Han, J. Li, Z. P. Cai, and Q. L. Han, Privacy reserved influence maximization in GPS-enabled cyber-physical and online social networks, in *Proc. 2016 IEEE Int. Conf. Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, Atlanta, GA, USA, 2016, pp. 284–292.

[13] Q. N. Li, Y. Zheng, X. Xie, Y. K. Chen, W. Y. Liu, and W. Y. Ma, Mining user similarity based on location history, in *Proc. 16^{th} ACM SIGSPATIAL Int. Conf. on Advances in geographic Information Systems*, Irvine, CA, USA, 2008, p. 34.

[14] A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[15] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, Personalized recommendation in social tagging systems using hierarchical clustering, in *Proc. 2008 ACM Conf. Recommender Systems*, Lausanne, Switzerland, 2008, pp. 259–266.

[16] Y. Shi, Y. Peng, W. X. Xu, and X. W. Tang, Data mining via multiple criteria linear programming: Applications in credit card portfolio management, *Int. J. Inf. Technol. Decis. Making.*, vol. 1, no. 1, pp. 131–151, 2002.

[17] M. Han, J. B. Wang, M. Y. Yan, C. Y. Ai, Z. J. Duan, and Z. Hong. Near-complete privacy protection: Cognitive optimal strategy in location-based services, *Procedia Comput. Sci.*, vol. 129, pp. 298–304, 2018.

[18] C. C. Aggarwal and H. X. Wang, A survey of clustering algorithms for graph data, in *Managing and Mining Graph Data*, C.C. Aggarwal and H. X. Wang, eds. Springer, 2010, pp. 275–301.

[19] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.

[20] J. Abello, M. G. C. Resende, and S. Sudarsky, Massive quasi-clique detection, in *Proc. 5^{th} American Symposium Cancun LATIN* 2002: *Theoretical Informatics*, Mexico, 2002, pp. 598–612.

[21] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York, NY, USA: W H Freeman and Co, 1979, p. 90.

[23] M. J. Rattigan, M. Maier, and D. Jensen, Graph clustering with network structure indices, in *Proc. 24^{th} Int. Conf. Machine Learning*, Corvalis, OR, USA, 2007, pp. 783–790.

[24] F. R. K. Chung, *Spectral Graph Theory—CBMS Regional Conferance Series in Mathematics*. Providence, RI, USA: American Mathematical Society, 1997.

[25] Y. Liang, Z. P. Cai, Q. L. Han, and Y. S. Li, Location privacy leakage through sensory data, *Secur. Commun. Netw.*, vol. 2017, p. 7576307, 2017.

[26] X. Zheng, Z. P. Cai, J. Z. Li, and H. Gao, Location-privacy-aware review publication mechanism for local business service systems, in *Proc. 2017 IEEE Conf. Computer Communications*, Atlanta, GA, USA, 2017, pp. 1–9.

[27] J. Li, Z. P. Cai, M. Y. Yan, and Y. S. Li, Using crowd, sourced data in location-based social networks to explore influence maximization, in *Proc. 35th Annu. IEEE Int. Conf. Computer Communications*, San Francisco, CA, USA, 2016, pp. 1–9.

[28] Z. B. He, Z. P. Cai, J. G. Yu, X. M. Wang, Y. C. Sun, and Y. S. Li, Cost-efficient strategies for restraining rumor spreading in mobile social networks, *IEEE Trans. Vehic. Technol.*, vol. 66, no. 3, pp. 2789–2800, 2017.

[29] T. Qiu, R. X. Qiao, and D. O. Wu, EABS: An event-aware backpressure scheduling scheme for emergency internet of things, *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 72–84, 2018.

[30] T. Y. Song, N. Capurso, X. Z. Cheng, J. G. Yu, B. Chen, and W. Zhao, Enhancing GPS with lane-level navigation to facilitate highway driving, *IEEE Trans. Vehic. Technol.*, vol. 66, no. 6, pp. 4579–4591, 2017.

[31] J. J. Wang, Y. L. Han, and X. Y. Yang, An efficient location privacy protection scheme based on the Chinese remainder theorem, *Tsinghua Sci. Technol.*, vol. 21, no. 3, pp. 260–269, 2016.

[32] Y. Wang, D. B. Xu, and F. Li, Providing location-aware location privacy protection for mobile location-based services, *Tsinghua Sci. Technol.*, vol. 21, no. 3, pp. 243–259, 2016.

[33] M. Barthélemy, Spatial networks, *Phys. Rep.*, vol. 499, nos. 1–3, pp. 1–101, 2011.

[34] D. Guo, Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP), *Int. J. Geogr. Inf. Sci.*, vol. 22, no. 7, pp. 801–823, 2008.

[35] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte, Uncovering space-independent communities in spatial networks, *Proc. Natl. Acad. Sci. USA*, vol. 108, no. 19, pp. 7663–7668, 2011.

[36] Y. Chen, J. Xu, and M. Z. Xu, Finding community structure in spatially constrained complex networks, *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 6, pp. 889–911, 2015.

[37] Y. X. Fang, R. Cheng, X. D. Li, S. Q. Luo, and J. F. Hu, Effective community search over large spatial graphs, *Proc. VLDB Endow.*, vol. 10, no. 6, pp. 709–720, 2017.

[38] Z. B. He, Z. P. Cai, and J. G. Yu, Latent-data privacy preserving with customized data utility for social network data, *IEEE Trans. Vehic. Technol.*, vol. 67, no. 1, pp. 665–673, 2018.

[39] M. Han, Q. L. Han, L. J. Li, J. Li, and Y. S. Li, Maximizing influence in sensed heterogenous social network with privacy preservation, *Int J. Sensor Netw.*, doi: 10.1504/IJSNET.2017.10007412.

[40] N. Cao, Z. Y. Yang, C. Wang, K. Ren, and W. J. Lou, Privacy-preserving query over encrypted graph-structured data in cloud computing, in *Proc. 31st Int. Conf. Distributed Computing Systems*, Minneapolis, MN, USA, 2011, pp. 393–402.

[41] X. Y. He, J. Vaidya, B. Shafiq, N. Adam, and X. D. Lin, Reachability analysis in privacy-preserving perturbed graphs, in *Proc. 2010 IEEE/WIC/ACM Int. Conf. Web Intelligence and Intelligent Agent Technology*, Toronto, ON, Canada, 2010, pp. 691–694.

[42] Z. P. Cai, Z. B. He, X. Guan, and Y. S. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, *IEEE Trans. Depend. Secure Comput.*, doi:10.1109/TDSC.2016.2613521.

[43] K. Mouratidis and M. L. Yiu, Shortest path computation with no information leakage, *Proc. VLDB Endow.*, vol. 5, no. 8, pp. 692–703, 2012.

[44] P. Damaschke, Degree-preserving spanning trees in small-degree graphs, *Discrete Math.*, vol. 222, nos. 1–3, pp. 51–60, 2000.

[45] A. Jeckmans, Q. Tang, and P. Hartel, Privacy-preserving profile matching using the social graph, in *Proc. 2011 Int. Conf. Computational Aspects of Social Networks*, Salamanca, Spain, 2011, pp. 42–47.

Yiran Shen received the PhD degree in computer science and engineering from the University of New South Wales in 2014. He is an associate professor in the College of Computer Science and Technology, Harbin Engineering University (HEU). He was a SMART Scholar at the Singapore-MIT Alliance for Research and Technology before he joined HEU. His current research interests are wearable/mobile computing, wireless sensor networks, and compressive sensing applications.

Dan Yin received the PhD degree from Harbin Institute of Technology in 2016. Dr. Yin is currently an assistant professor at Harbin Engineering University. Her research areas focus on big data, graph mining, and privacy preservation.