Social Network Anonymization via Local-perturbing Approach

Peng Liu[1, 2], Huanjie Wang[1], Shan Lin[1], Xianxian Li[1, 2]

[1] Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, China
[2] School of Computer Science and Engineering, Beihang University, China
liupeng@gxnu.edu.cn, Whj.6040@163.com, lin-sam@foxmail.com, lixx@gxnu.edu.cn

Abstract

Social networks provide a large amount of social network data, which is collected, studied and distributed for various purposes. Because social network data usually contains sensitive personal information, it needs to be anonymized before publication. Many data anonymization methods have been proposed to protect the privacy of individuals; but most methods were proposed for general purposes and suffer the problem of excessive information loss when they are used for specific purposes. In this paper, we focus on the problem of improving data utility when applying privacy-preserving methods to the original data for protection privacy. We propose two novel local-perturbing methods: one is based on the *k*-anonymity model; the other is based on a randomization model. Both methods can achieve the same privacy levels as *k*-anonymity model while minimizing the impact on community structure. We evaluate the performance of our methods by testing three real-world datasets. Experimental results show that both methods loss less community structure information compared to existing methods.

Keywords: Social networks, Privacy protection, Anonymization, Community structure

1. Introduction

In recent years, social network services have steadily and rapidly grown. Social network services include a diverse set of social network platforms such as LinkedIn and Twitter but also encompass user interaction networks like email, chat, and blog applications. The data from social networking services is valuable for academic researchers from a variety of field such as sociology and information science [10]. To use social network data effectively, the data owner usually share it with different data miners. However, unlike most scientific data, these data contains personal and sensitive information about individuals, such as a person's name, age, address, personal relationships, and interests. Sharing raw data breaches consumer privacy laws. To fulfill the needs of data sharing, many anonymization methods have been developed to protect the privacy of individuals. However, most of them are proposed for general purpose, that is, the privacy-preserving process does not take into account the purpose of using the data. In general, data owners and data reception are fully communicated when sharing data. If we consider the using purpose of the data when designing of privacy protection methods, we will be able to retain better data utility.

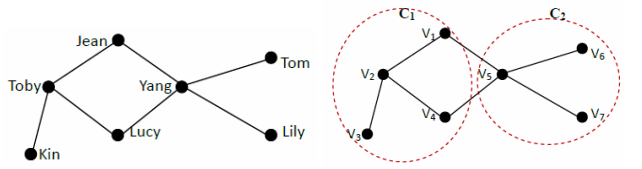
In this paper, we assume that the purpose of data sharing is to conduct community-related analysis and propose local-perturbing methods to anonymize social network data for preventing individuals from being re-identified. We study the problem of restricting the privacy-preserving method into local group in the original data to improve output data utility. Then we propose two novel local-perturbing methods: one is based on a *k*-anonymity model, with the other based on a randomization model. Both methods achieve comparable performance with the *k*-anonymity model while minimizing the impact on community structure.

2.2 Motivation

To protect the privacy of individuals, simple anonymization methods remove identity information from each node, such as names and IDs. This process is insufficient as described by many research studies [12]. Example 1 illustrates a privacy attack on a naive anonymized social network.

Example 1. A typical social network is shown in Figure 1(a). The social network's naive data anonymization graph is illustrated in Figure 1(b). The data anonymization method is characterized by removing the names of every participant. Nevertheles, an adversary could re-identify the victim via neighborhood attack [18] or a complex structural attack [3]. In this example, we assume that the adversary knows that the victim Kin has one friend, and his friend has three friends. With such background information, the adversary may easily infer that Kin is the node V_3 in Figure 1(b).

*Corresponding Author: Xianxian Li; E-mail: lixx@gxnu.edu.cn
DOI: 10.3966/160792642018011901024



(a) (b)

Figure 1. (a) Raw social network; (b) Naive anonymized social network

One of the effective ways to protect the privacy of individuals in Example 1 is through the use of *k*-anonymity [3-4], or randomization [21]. The *k*-anonymity model divides all nodes into several groups. Each group has at least *k* indistinguishable nodes.The randomization model adds noise data to the original data, and protects the privacy via uncertainty.

Both the *k*-anonymity and randomization models focus on keeping general properties unchanged while protecting the privacy of individuals. Some properties, such as community structure, are import for science research in the fields such as sociology and anthropology. These properties also benefit practical applications such as recommendation system and public administration. Without considering community structure information, the boundaries of the original community structure will likely become blurry after the *k*-anonymity reconstruction process [2-3]. We use the following example to illustrate the drawbacks of the *k*-anonymity methods.

Example 2. The original data shown in Figure 1 has 2 communities $\{C_1, C_2\}$ and 2 edges between them, where C_1= $\{V_1, V_2, V_3, V_4\}$, C_2= $\{V_5, V_6, V_7\}$. Figure 2 shows its anonymized output using the privacy method proposed by A. Campan [3]. All nodes are grouped into two super nodes, cl_1 and cl_2, based on neighbor similarities with parameter k=3. Because most graph analysis algorithms can only process atomic nodes and edges, the *k*-anonymity super nodes usually need to be reconstructed before analysis. Figure 20 shows a possible result of reconstructing the data in Figure 30. The number of edges connecting the communities C_1 and C_2 in Figure 20 becomes four, which blurs the boundary of the communities. There is a considerable difference between the original and anonymized graph. This is likely due to the difference in community structure information.

To address the problems mentioned above, we propose novel local-perturbing approaches, that can achieve the same privacy requirement of the *k*-anonymity, while preserving high utility of the community structure. In addition, we extend the local-perturbing approach presented in conference paper [19], and propose a local-perturbing approach that uses the randomization-based privacy-preserving model.

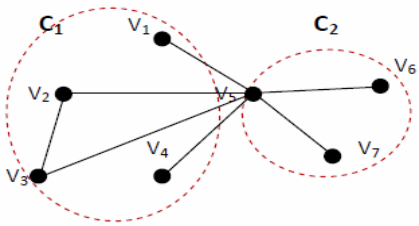


Figure 2. A reconstructed graph

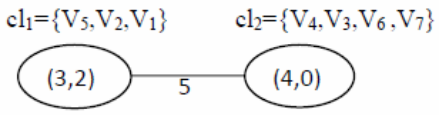


Figure 3. A 3-anonymity graph

1.2 Contributions

The contributions of this paper are summarized as follows.

(1) We study problem of restricting the privacy-preserving methods into local groups of the original data to improve output data utility. Then we localize two types of privacy models: the *k*-anonymity and randomization.

(2) By considering the community structure in the clustering and perturbation procedures, our local-perturbing methods have comparable privacy levels with *k*-anonymity model while minimizing the impact on community structure.

The remaining parts of the paper are organized as follows. Section 2 defines the privacy problem in social networks. Section 3 presents our two privacy-preserving methods. Section 4 examines our solution using real world datasets. Section 5 discusses related work and Section 6 concludes this paper.

2 Problem Definition

In this paper, we model a social network as an undirected graph G= (V, E), where V is a set of nodes, and $E \in V \times V$ is a set of edges. Each node indicates an individual in the social network. An edge between two nodes represents their relationship. Only binary relationships are allowed in our model.

2.1 The Privacy Model

Adversaries usually rely on background knowledge to re-identify individuals and then learn their sensitive information from published social network data. In this paper, we assume that an adversary knows sub-graph information about the target victim, and wants to re-identify the node of the targeted victim in the published data. We first define the scenario of a privacy breach.

Definition 1 (Privacy breach). Let G be a social network and G' be the published anonymization data of G. A privacy breach occurs when adversaries can successfully map a target user to a node in G' with a

level of confidence higher than 1/*k*.

2.2 Problem Statement

In this paper, we assume that the data publishers are trusted and that the social network users are willing to provide their detailed information to them. To protect individual privacy when sharing the data, the data publishers want to design a sanitization technique to transform the original data into an anonymized version. The problem statement of this research is:

Given a social network *G* without labels, and a privacy requirement *k*. The problem of anonymization of social network for community structure is to transform *G* to a local-perturbing social network *G*' that does not breach privacy requirements, while retaining as much community structure as possible.

2.3 Relevant Definitions

In this section we provide some relevant definitions that will emerge in the description of our work.

The nodes in the social network tend to form closely-knit groups, and the connections within the group are dense; whereas, connections with the rest of the network are sparse. These groups are also known as communities.

Definition 2 (community in social network). Let $G=(V, E)$ be a social network with a set of communities $C=\{C_1, C_2, \ldots, C_m\}$, where $C_i \cap C_j = \phi$ for all $1 \le i \ne j \le m$. For each $C_o \in C$, the density of the internal connection is higher than the external connection.

In this paper, we choose a classical community-detecting algorithm, the Girvan and Newman algorithm [7], to discover community structure. The algorithm uses the modularity optimization method to explore the community structure. The modularity of a social network is defined as follows [6, 12]:

$$Q=\sum_{c=1}^{n}[\frac{l_c}{m}-(\frac{d_c}{2m})^2] \quad (1)$$

where n is the number of communities, l_c is the total number of edges in the community c, d_c is the sum of the degrees of nodes in the community c, and m is the number of edges in G.

The *k*-anonymity-based local-perturbing method includes two steps, clustering and reconstruction. Some relevant concepts used in the process of anonymization are formally defined as follows.

Definition 3 (*k*-cluster social network). Let $G=(V, E)$ be a social network, where k is the threshold specified by social network data publishers. For a given clustering $CL=\{cl_1, cl_2, \ldots, cl_n\}$ of V, the corresponding social network is denoted as G_{cl} where $cl_t \cap cl_c = \phi$ for all $1 \le t \ne c \le n$, and $|cl_i| \ge k$ for $1 \le i \le n$.

In the clustering process, all nodes are divided into clusters based on similarity criteria. In other words the nodes in each cluster are as similar as possible. We use the distance between nodes as the measurement of similarity and define the distance between two nodes as

Definition 4 (the distance between nodes). The distance between two nodes (V_i, V_j) is:

$$\text{dist}(V_i, V_j)=\frac{|\{V_k \mid V_k \in (adj[V_i] \oplus adj[V_j], V_k \ne V_i \ne V_j)\}|}{n-2} \quad (2)$$

where n is the number of nodes in a graph. The reason that n is reduced by 2 in the denominator is that we exclude V_i and V_j from the set. The symbol $adj\ [V_i]$ denotes the set of neighbors of a node V_i. For example, $adj\ [V_2\ ]$ in Figure 1(b) is $\{V_1, V_3, V_4\}$. Thus, the distance between V_1 and V_2 in Figure 1(b) is $dist(V_1,V_2)=\frac{|\{V_2,V_5\} \oplus \{V_1,V_3,V_4\}/(\{V_1\} \cup \{V_2\})|}{5}=\frac{3}{5}$.

Next, we define the distance between a node and a cluster [3].

Definition 5 (the distance between a node and a cluster). The distance between a node V_p and a cluster cl_q is:

$$dist(V_p, cl_q)=\frac{\sum_{V_j \in cl_q} dist(V_p, V_j)}{|cl_q|}. \quad (3)$$

3 Local-perturbing Methods

In this section, we introduce two local-perturbing methods to preserve privacy in data publishing. One is based on the *k*-anonymity model and the other is based on the randomization model. Both methods can transform the original social network *G* into a locally perturbed graph *G'* that preserves privacy while providing better data utility for community analysis.

3.1 The Local Perturbation Based on *k*-anonymity

Because most of the previous data anonymization methods only consider data privacy and ignore data application, we consider application requirements and preserve the key data property for the community analysis related applications by using a local-perturbing technique.

3.1.1 Cluster for Social Networks

The first step of the *k*-anonymity-based method is to transform *G* into a *k*-cluster graph G_{cl}. The pseudocode of the method is shown in Table 1.

In the *K*-cluster algorithm, line 1 sets the variable *CL* to store the clustering result, and also initializes the intermediate variables *i*. Lines 2 to 8 sequentially divide the nodes in set *V* into *i* clusters of *k*-size until *V* contains fewer nodes than *k*. Line 3 generates a new

Table 1. The *K*-Cluster( ) algorithm

Algorithm 1. *K*-Cluster(*G*, *k*)
Input: Social network *G*=(*V*, *E*), *V* in descending order of degree and the threshold *k*.
Output: A *k*-cluster graph G_{cl} and number of clusters.
1: $CL = \phi$; i=1;
2:**while** |*V*|>=*k*
3: V^i_{seed}=*V* [0]; cl_i={ V^i_{seed} }; *V*=*V*-cl_i; //select the seed node for cl_i;
4: **while** |cl_i|<*k*
5: FindBestNode(*V*, cl_i);
6: **end while**
7: $CL = CL \cup \{cl_i\}$; i++;
8: **end while**
9: **if** $V \neq \phi$
10: $cl_i \leftarrow V$;//store *V* in cl_i;
11: **for each** *v* in cl_i
12: FineBestCluster(*v*, *CL*);
13: *V*=*V*-{*v*};
14: **end for**
15:**end if**

Table 2. The function of FindBestNode( )

Function1. FindBestNode(*V*, cl_i)
1:**for each** node V_p in *V* :
2: compute the distance dist (V_p, cl_i);
3:**end for**
4:store the nodes with the smallest distance in set *CanN*;
5:**if** |*CanN*|>1
6: **for each** node V_q in *CanN*://traverse the list
7: **if** V_q and V^i_{seed} are in the same community
8: add the V_q to cl_i, *V*=*V*-{V_q};
9: **return**;
10: **end if**
11: **if** no node is in the same community as V^i_{seed}
12: add the last node V_q to cl_i;
13: *V* = *V* –{ V_q };
14: **return**;
15: **end if**
16:**else**
17: add the only element V_q in *CanN* to cl_i;
18: *V*=*V*-{V_q};

cluster cl_i with the node in current *V* that has the maximum degree at each step, and then removes that node from *V*. Lines 4 to 6 use the subroutine FindBestNode(*V*, cl_i) to find and add the suitable nodes to the current group in turn until the cluster size is *k*. Line 7 records the current cluster result in *CL* and increases the group number *i* by 1. If the number of elements contained in the set of nodes *V* is not a multiple of *k*, it is necessary to use lines 9 to 14 to process the remaining nodes. Line 9 tests whether the set *V* is empty. If not empty, lines 10 to 14 assign nodes to the appropriate cluster.

Due to the power law degree distribution [5], it is likely that more than one node have the same degree, which results in multiple nodes have the same distance from the current cluster. We are faced with the question of how to choose the appropriate nodes from the candidates of the current cluster, that have a minimal impact on the community structure. Different selections lead to different results. Thus, we devise a heuristic subroutine FindBestNode() for selecting approprite nodes, as shown in Table 2.

In detail, Lines 1 to 3 use Equation (3) to calculate the distance from each alternative point V_p to the cluster cl_i. Line 4 selects the nodes with the minimum distance and stores them in *CanN*. Note that there may be multiple nodes with the same minimum distance. Line 5 tests whether *CanN* contains only one element. If the set *CanN* has only one element, lines 16-18 are processed. Line 17 adds the unique node to the community cl_i. Line 18 removes this node from set *V*. If the set *CanN* has more than one element, lines 6-15 are processed. The main processing step is selecting the node which is in the same community as V^i_{seed} to join cl_i first. If there is no such node, the function selects the last node in the set *V* to join cl_i.

When the number of nodes in *G* is not a multiple of *k*, it is possible that the number of nodes in current *V* is less than *k*. Then, we should find the best cluster for each of them. The specific process is described as the FindBestCluster() function, shown in Table 3.

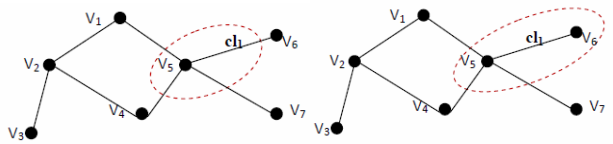
Table 3. The function of FindBestCluster( )

Function2. FindBestCluster(*v*, *CL*)
1:*mincl*=dist(V_m, cl_0);
2:**for each** cluster cl_n in *CL*:
4: **if** V_m and V^n_{seed} are in the same community
5: *Bestcl*=cl_n; Break;
6: **else**
7: *dist*=dist(V_m, cl_n);
8: **if** *dist*<mincl
9: *mincl*=*dist*;
10: *Bestcl*=cl_n;
11: **end if**
12: **end if**
13:**end for**
14:add V_m to cl_n;

We use an example to explain how to generate *cl* in Agorithm 1.

Example 3 Consider *k*=3 in Figure 4(a). The list of *V* after sorting (by degree) is {V_5, V_2, V_1, V_4, V_3, V_6, V_7}. A new cluster cl_1 starts with the seed node V_5 which has the largest degree. Then, V_5 is removed from list *V*. We calculate the distance between cl_1 and the nodes in *V*, and get the minimum dist(V_p, cl_i)= dist(V_2, cl_1)= dist(V_6, cl_1)= dist(V_7, cl_1)= 3/5. We can then easily get the candidate set *CanN*={V_2, V_6, V_7}. Then, we access

CanN in turn. V_2 is not in the same community as V_5, but the next node V_6 is in the community. Therefore, we stop the search and add V_6 into cl_1. We then remove V_6 from list V. We repeat the steps in the algorithm until the number of nodes in cl_1 is 3.



(a) (b)

Figure 4. The illustration of generating *cl*

3.1.2 Reconstruction

To protect a user's privacy and analyze data effectively, the *k*-cluster social network data needs to be reconstructed before releasing. This process will bring uncertainty to the reconstruction of the graph, which is worse for data analyzers to achieve accurate community structure information. Here, we reconstruct the *k*-cluster graph by randomly regenerating edges in each cluster uniformly and making sure that the number of intra-cluster edges in each cluster is the same as the original. Note that the number of inter-cluster edges remains the same as the original.

For each cluster in the *k*-cluster social network graph, we clean all the edges first and regenerate edges among nodes in the current cluster with uniform probability until the number of edges is the same as the original cluster.

A uniform probability distribution is used when selecting any node pair to regenerate edges in each cluster during the reconstructing process. The probability distribution guarantees that each node has an equal likelihood of being chosen, in other words, the nodes in a cluster are indistinguishable. In addition, the size of each cluster is bigger than *k*, therefore, the probability for an adversary to re-identify any node in the anonymized social network *G*' is no more than 1/*k*. From the details above, we can safely assume that our local-perturbing approach can achieve the same privacy performance as *k*-anonymity.

3.2 The Local Perturbation Based on Randomization

The randomization-based method uses perturbation to protect privacy. It adds random noise to the original social network data by adding or deleting edges randomly, and protects the data against the re-identification risk in a probabilistic manner. However, existing random methods do not consider the inner attribute of the data, and lead to much information loss. In this paper, we restrict the random perturbation to the local groups of the graph, and devise a local-perturbing method for privacy-preserving data sharing.

To restrict the perturbation to the local, we use the fast GN community-detecting algorithm to divide the original data into communities [20, 22]. Then, we adjust the results by combining the communities that contain less than *k* nodes. We use the perturbing method proposed by Ying to protect the privacy of each community [21]. The pseudocode of the local-perturbing method based on randomization is illustrated in Table 4.

Table 4. The Localrandomization( ) algorithm

Algorithm 2. LocalRandomization(*G*, *k*)
Input: Social network *G*=(*V*, *E*) and the threshold *k*
Output: The anonymized graph *G'*
1: *comm*=FastGN(*G*);
2: CombineCom(*comm*, *G*);
3: **for each** community c_i in *comm*
4: *m*=CalEdgesNum(k, c_i);
5: RandomPerturb(*m*, c_i);
6: **end for**
7: Reconstruct the edges between communities;

The fuction FastGN() returns a list of communities with maximum modularity. The function Combine Com(), in line 2, merges the communities that contain nodes less than *k*. The pseudo code is illustrated in Table 5. The function CalEdgesNum(), in line 3, uses formula (4) of paper [21] to calculate the number of perturbing edges *m*. Line 6 uses the randomization method proposed by Ying [21] to perturb the community c_i with parameter *m*. Line 7 reconstructs the edges between communities.

$$J(m) = \min_{\alpha \in \Omega} \tau_r[\alpha \mid E(\hat{d})] \geq 1 \quad \textbf{(4)}$$

Table 5. The function of CombineCom( )

Function 3. CombineCom(*comm*, *G*)
1 **for each** community c_i in *comm*
2: **if** Size(c_i)<*k*
3: find its directly connecting community list *dc*;
4: merge c_i with the community that the result has the maximum modularity;
5: **end if**
6:**end for**

4 Experimental Evaluations

To evaluate our local-perturbing methods, we compared our methods with the *SaNGreeA-uniform* privacy-preserving method. The *SaNGreeA-uniform* method was first proposed by Campan in [3], and extended in [2] by determining the impact of the method on community structure. The main idea of the *SaNGreeA-uniform* is generalization: the nodes in the original data generalize to super nodes; the edges between super nodes generalize to edges with weight.

We choose the *SaNGreeA-uniform* method as a comparison mainly for two reasons: (1) The generalization of the *SaNGreeA-uniform* is related to the localization of this paper; (2) The author also studies the effect of the *SaNGreeA-uniform* on community structure in subsequent paper [2]. We tested several utility measurements to show how well the published data preserves the structural information of the original data.

4.1 Datasets

We evaluate the data utility on three real-life datasets: WebKB dataset, Citation dataset and Cora dataset.

WebKB(http://linqs.umiacs.umd.edu/projects//projects/lbc/index.html) consists of 877 websites coming from 4 universities and 1608 relationships between them.

Citation (http://www.datatang.com/data/17310) is a citation graph dataset. It consists of 2555 papers and 6101 citation relationships. This dataset is a directed multigraph, collected by an academic researcher of Tsinghua University and published in a Web site named Datatang.

Cora(http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html) is a directed graph dataset which contains 2708 nodes and 5429 edges.

The three data sets are all in text format, using the binary tuple $<v_1, v_2>$ to denote the edge between node v_1 and node v_2. The tuple of each edge occupies a line in the data file.

We implemented our two local-perturbing algorithms with Python (2.7.X). We use the open source software package Network X to store and manipulate the graph data. We also use the open source package SciPy to perform probability-related calculations.

4.2 Data Utility Measurements

We use jaccard similarity and the change of community ΔQ to compare community preservation between the initial graph and anonymized graph.

The Jaccard similarity is a statistic used for comparing the similarity and diversity of sample sets. We use it as a criterion for the difference of two communities, and define it as

$$J_i(C_i) = \frac{|C_i \bigcap C'_j|}{|C_i \bigcup C'_j|}, i \in [1,n], j \in [1,m], \quad \textbf{(5)}$$

where C_i is a set of nodes of each community in original graph G and C'_i is the corresponding set of nodes in the anonymized graph. To evaluate the similarity between the initial graph and anonymized graph, we sum all the differences of communities and define the similarity measurement as the arithmetic mean of $J(C_i)$.

$$J(G,G') = \frac{\sum_{i=1}^{n} J_i}{n}, i \in [1,n]. \quad \textbf{(6)}$$

Modularity is one of the measures to indicate community properties of networks. We use the change of modularity to test the community information change after being anonymized. Intuitively, the greater the result, the more the community becomes blurry, that is, the community information of the original social network do not get preserved.

$$\Delta Q = Q - Q' \quad \textbf{(7)}$$

In addition, we use a general measurement Clustering Coefficient **(CC)** to evaluate the impact of our local-perturbing methods. This measurement represents the degree to which the vertices in a graph tend to be clustered together. The CC of a vertex v is given by the proportion of connections between the vertices within its neighborhood divided by the number of connections that could possibly exist between them. This is calculated using

$$C_v = 2T(v)/d_v(d_v-1), \quad \textbf{(8)}$$

where $T(v)$ is the number of triangles through vertex v and d_v is the degree of v.

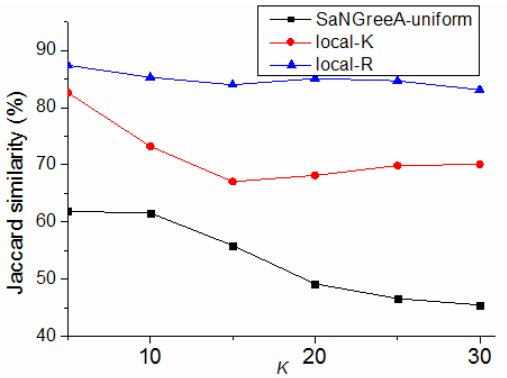
We conducted the experiments on a workstation with 32GB RAM and Xeon E5-2630 CPU. We examed the impacts of privacy-preserving methods on three datasets by changing k values from 5 to 30. For convenience, we use the notation *local-K* to represent our local-perturbing method based on k-anonymity and local-R to represent our local-perturbing method based randomization. For the *SaNGreeA-uniform* and *local-k* methods, the value of k indicates that the attacker cannot re-identify the target node within k nodes. For the local-*R* method, the value of k indicates that the probability that the attacker can re-identify the target node is not more than $1/k$. In this setting, the three privacy-preserving methods are comparable.

4.3 Results and Analysis

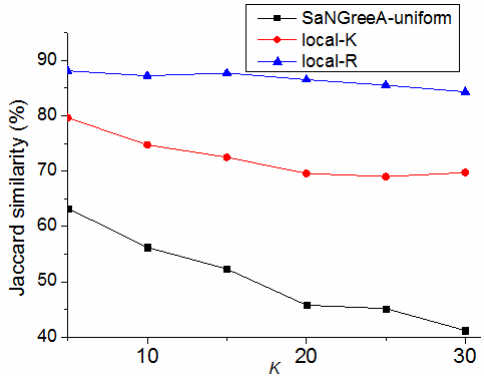
We first tested how well the published graph represents the original graph. We use *Jaccard similarity* and the change of modularityΔQ to measure the changes.

Figure 5(a), (b), and (c) show the Jaccard similarity of the three datasets respectively. The vertical axis represents the Jaccard similarity of each method in terms of k. The horizontal axis represents the value of k. The legend of *local-k* represents our local-perturbing method based on k-anonymity and the local-R represents the randomization-based method. As shown in the Figure 5(a) to (c), the Jaccard similarity decreases as k increases because stronger privacy protection requires more perturbation, which will generate a difference with the original graph. The results also show that for all the three datasets, our two

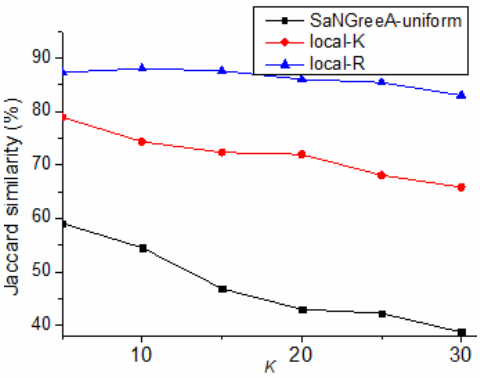
local-perturbing methods perform better than the *SaNGreeA-uniform* algorithm.



(a)WebKB



(b)Citation

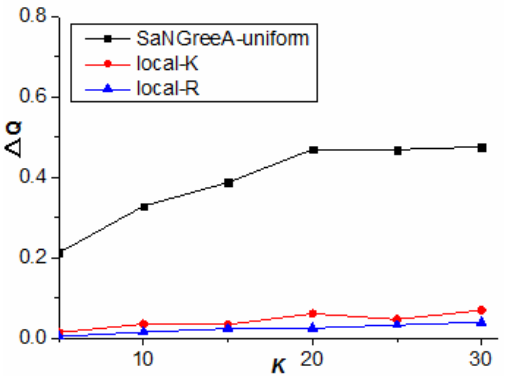


(c)Cora

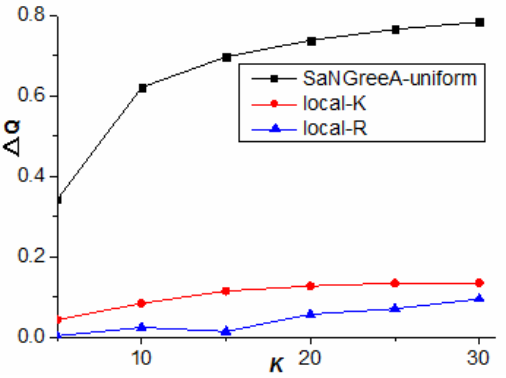
Figure 5. Jaccard similarity for different *k*

Figure 6 (a), (b), and (c) show the change of modularity ΔQ of the three datasets respectively. From the figures, we can see that the boundaries between communities from the original social network become more blurry with the increase of the values of *k*. Our two methods have less of an impact than the *SaNGreeA-uniform* method. The randomization-based method performs the best because it impacts the modularity only when it merges the communities containing nodes less than *k*.

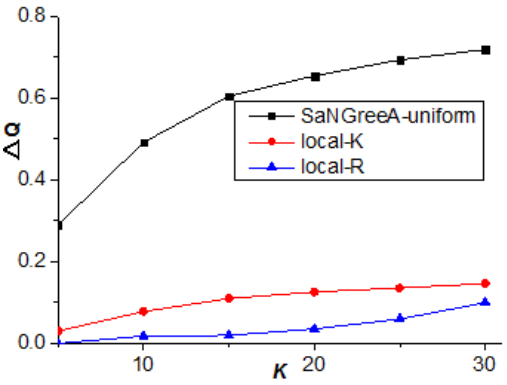
The social network data is complex and has many topological properties. The CC is an important metric used to identify social network data. We use this metric



(a)WebKB



(b)Citation

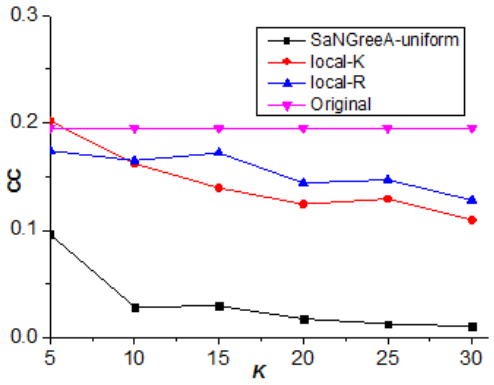


(c)Cora

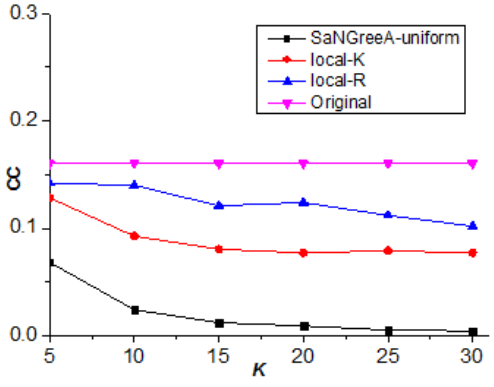
Figure 6. ΔQ for different *k*

to evaluate the impact of all three privacy-preserving methods. The changes in *CC* are presented in Figure 7(a), (b), and (c). As the value of *k* increase, the *CC* term becomes smaller. The *CC* values of *SaNGreeA-uniform* algorithm are even close to 0. Intuitively, our approaches exhibit less different to the original social network.

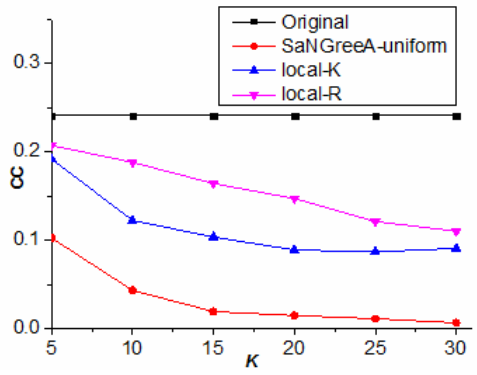
The comparison results show that the local-R method performs better than the other two algorithms in all three metrics. The results also show that the randomization approach is more advantageous for protecting privacy. Especially when the stochastic process is restricting, the randomization method can keep the user specified information better.



(a) WebKB



(b)Citation



(c)Cora

Figure 7. *CC* for different *k*

5 Related Work

There are many research efforts to develop privacy-preserving methods for social networks. The privacy of social network data can be categorized into two types. One type is node-privacy, which focuses on node re-identification [3, 8, 10, 18] and node-attribute disclosure [3]. In node re-identification, the attacker's goal is to identify the target victim to obtain valuable information. In nodal-attribute disclosure, the attack goal infers sensitive information from a targeted victim, such as disease or salary. The other type is edge-privacy, which consists of link re-identification [13, 17] and edge-based attribute disclosure [3]. For link re-identification, the attack goal is to identify sensitive relationships between nodes; while for edge-based attribute disclosure, the attack goal is to infer sensitive relationships between nodes. This paper focuses on preventing node re-identification in unlabeled graphs. In this way, we could protect the sensitive information about individuals, such as the importance of the target victim in the community.

To protect sensitive information, some data anonymization techniques have been proposed in recent years. These techniques can be classified into four categories: adding nodes [4], adding and deleting edges [3, 10, 17], generalization [3, 8], and randomization [1, 21].

Recently, research on community-based node re-identification have been studied in [14-16]. Tai [14] presented the model of structural diversity. In this method, for each node *v*, there must exist at least *k*-1 other nodes located in at least *k*-1 other communities with the identical degree of *v*. This implies that nodes can be protected. Due to the existence of community structure in social networks, this will cause more structural information losses in previous research [15-16].

On the whole, studies related to this paper we using are social network clustering model and graph reconstructing model. Besides, we also use the community detection approach to detecting community structure of social network graph.

6 Conclusion

In this paper, we formally define the problem of anonymizing social network in order to share data with third-parties. We propose two novel local-perturbing approaches that localize two types of privacy models, *k*-anonymity and randomization, to solve the privacy problem. Considering the community structure in the clustering and perturbing procedure, our proposed methods can achieve the same privacy requirement of the *k*-anonymity model while minimizing the impact on community structure. Our methods can be made into a software. The data owner uses the software to sanitize the data before it is released. Our methods can protect the privacy of users in the data.We performed experiments on three datasets: the *WebKB dataset*, *Citation dataset*, and *Cora dataset*. Each dataset was measured against the three metrics: the *jaccard similarity*, the change of modularity, and the average clustering coefficient. The experimental results show that our methods can provide the same privacy protection level of *k*-anonymity and have less loss of community structure information compared with existing techniques.

Acknowledgments

The research is partially supported by the National Science Foundation of China (Nos. 61672176, 61662008 and 61502111), the Guangxi Natural Science Foundation (Nos., 2014GXNSFAA118018, 2015GXNSFBA139246 and 2016GXNSFAA380192), the Guangxi Science and Technology Project (No. AD16380008), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, and the Guangxi Collaborative Center of Multisource Information Integration and Intelligent Processing.

References

[1] P. Boldi, F. Bonchi, A. Gionis, T. Tassa, Injecting Uncertainty in Graphs for Identity Obfuscation, *Proceeding of the VLDB Endowment*, Vol. 5, No. 11, pp. 1376-1387, August, 2012.

[2] A. Campan, Y. Alufaisan, T. M. Truta, Community Detection in Anonymized Social Networks, *Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference*, Springer, Athens, 2014, pp. 396-405.

[3] A. Campan, T. M. Truta, Data and Structural k-anonymity in Social Networks, *Lecture Notes in Computer Science, Vol. 5456*, F. Bonchi, E. Ferrari, W. Jiang, B. Malin, eds., pp. 33-54, Springer, 2009.

[4] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, S. Venkatesh, k-Anonymization of Social Networks by Vertex Addition, *Proc. of 15th ADBIS (2). CEUR Workshop Proceedings*, Vol. 789, 2011, pp. 107-116, CEUS-WS.org.

[5] M. Faloutsos, On power-law Relationships of the Internet Topology, *Computer Communications Review*, Vol. 29, No. 5, pp. 251-262, October, 1999.

[6] S. Fortunato, Community Detection in Graphs, *Physics Reports*, Vol. 486, No. 3, pp. 75-174, January, 2010.

[7] M. Girvan, M. E. Newman, Community Structure in Social and Biological Networks, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 99, No. 12, pp. 7821-7826, June, 2002.

[8] T. Tassa, D. J. Cohen, Anonymization of Centralized and Distributed Social Networks by Sequential Clustering, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 2, pp. 311-324, February, 2013.

[9] M. Hechter, *Principles of Group Solidarity*, University of California Press, 1988.

[10] K. Liu, E. Terzi, Towards Identity Anonymization on Graphs, *Proceedings of ACM SIGMOD*, Vancouver, Canada, 2008, pp. 93-106.

[11] A. Narayanan and V. Shmatikov, De-anonymizing Social Networks, *30th IEEE Symposium on Security and Privacy*, Washington, DC, 2009, pp. 173-187.

[12] M. E. Newman, M. Girvan, Finding and Evaluating Community Structure in Networks, *Physical Review E*, Vol. 69, No. 2, pp. 026113, June, 2004.

[13] C.-H. Tai, P. S. Yu, D. N. Yang, M. S. Chen, Privacy-preserving Social Network Publication Against Friendship Attacks, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 2011, pp. 1262-1270.

[14] C.-H. Tai, P. S. Yu, D.-N. Yang, M.-S. Chen, Structural Diversity for Resisting Community Identification in Published Social Networks, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 26, No. 1, pp. 235-252, January, 2014.

[15] Y. Wang, L. Xie, B. Zheng, K. C. Lee, Utility-oriented k-anonymization on Social Networks, *DASFAA 2011*, J. X. Yu, M. H. Kim, R. Unland, eds., pp. 78-92, Springer, 2011.

[16] Y. Wang, L. Xie, B. Zheng, K. C. K. Lee, *High Utility K-Anonymization for Social Network Publishing, Knowledge and Information Systems*, Vol. 41, No. 3, pp. 697-725, December, 2014.

[17] M. Yuan, L. Chen, P. S. Yu, Personalized Privacy Protection in Social Networks, *Proceedings of the VLDB Endowment*, Vol. 4, No. 2, pp. 141-150, September, 2010.

[18] B. Zhou, J. Pei, Preserving Privacy in Social Networks Against Neighborhood Aattacks, *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, Washington, DC, 2008, pp. 506-515.

[19] H. Wang, P. Liu, S. Lin, X. Li, A Local-perturbation Anonymizing Approach to Preserving Community Structure in Released Social Networks, *12th EAI International Conference on Quality, Reliability, Security and Robustness in Heterogeneous Networks*, Seoul, South Korea, 2011, pp. 36-45.

[20] R. Y. Chang, S. L. Peng, G. Lee, C. J. Chang, Comparing Group Characteristics to Explain Community Structures in Social Media Networks, *Journal of Internet Technology*, Vol. 16, No. 5, pp. 957-962, November, 2015.

[21] X. Ying, K. Pan, X. Wu, L. Guo, Comparisons of Randomization and k-degree Anonymization Schemes for Privacy Preserving Social Network Publishing, *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, Paris, France, 2009, p. 10.

[22] M. E. J. Newman, Fast Algorithm for Detecting Community Structure in Networks, *Physical. Review E*, Vol. 69, No. 6, pp. 066133-066133, June, 2004.

Biographies

Peng Liu is currently a Ph.D. candidate in School of Computing Science and Engineering at Beihang University, China. He joinded the College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China, as an assistant professor in 2007. Since 2015, he has been an associate professor. His current research interests include network security, data privacy, and graph mining.

Huanjie Wang received her B.S. degree in department of Computer and Information Technology from Anyang Normal University in 2012 and MS degree in department of Computer Science and Information Technology from Guangxi Normal University, Guilin, China, in 2016. Her current research topic is network security in Huawei Technologies Co. Ltd, Shenzhen, China.

Shan Lin received his B.S. degree in the College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China, in 2014 and is currently a master degree candidate in the same college since 2014 .His current research interests include network security and data privacy.

Xianxian Li received his Ph.D. degree in computer software and theory from Beihang University in 2002. He worked as a professor at Beihang University during 2003-2010. He joined the faculty of the College of Computer Science and Information Technology, Guangxi Normal University Guilin, China, as a professor in 2010. his current research interests mainly include network and multi-source information security.