A Generalization Based Approach for Anonymizing Weighted Social Network Graphs*

Xiangyu Liu and Xiaochun Yang

College of Information Science and Engineering,
Northeastern University, Shenyang 110819, China
yangxc@mail.neu.edu.cn

Abstract. The increasing popularity of social networks, such as online communities and telecommunication systems, has generated interesting knowledge discovery and data mining problems. Since social networks usually contain personal information of individuals, preserving privacy in the release of social network data becomes an important concern. An adversary can use many types of background knowledge to conduct an attack, such as topological structure and/or basic graph properties. Unfortunately, most of the previous studies on privacy preservation can deal with simple graphs only, and cannot be applied to weighted graphs. Since there exists numerous unique weight-based information in weighted graphs that can be used to attack a victim's privacy, to resist such weight-based re-identification attacks becomes a great challenge. In this paper, we investigate the identity disclosure problem in weighted graphs. We propose k-possible anonymity to protect against weight-based attacks and develop a generalization based anonymization approach(named GA) to achieve k-possible anonymity for a weighted graph. Extensive experiments on real datasets show that the algorithm performs well in terms of protection it provides, and properties of the original weighed network can be recovered with relatively little bias through aggregation on a small number of sampled graphs.

1 Introduction

Social network applications, such as online communities and telecommunication systems, have become popular for information sharing. Exploring the properties of social networks has generated interesting knowledge discovery and data mining problems. However, social networks usually contain individuals' sensitive information. Preserving privacy in the release of social network data becomes an important concern.

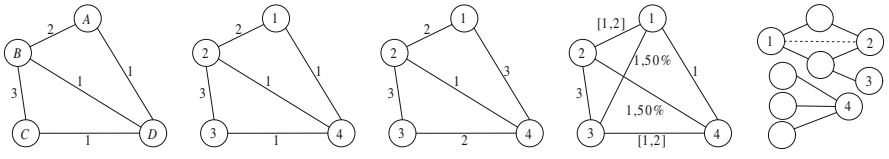
One fundamental privacy issue in publishing social network data is identity disclosure problem. Most of existing studies on privacy preservation deal

* The work is partially supported by the National Natural Science Foundation of China (Nos. 60973018, 60973020) and the Fundamental Research Funds for the Central Universities(Nos. N090504004, N100604013, N100704001, N090104001).

H. Wang et al. (Eds.): WAIM 2011, LNCS 6897, pp. 118–130, 2011.

with simple graphs(i.e. undirected, unweighted and loopless graphs), which consider graph structure as adversaries' background knowledge. However, weighted graphs are more descriptive and common in real world than simple graphs. Applying previous privacy preservation techniques to weighted graphs cannot protect privacy effectively.

In weighted graphs, there exists numerous weight related information that can be used to attack a victim's privacy. [1] studies how the weights on the connections of a target vertex with other vertices in weighted graph results in identity disclosure problem. For a vertex v, the sequence of weights on all edges connecting v with other vertices sorted in descending order is defined as v's weight bag, denoted $\boldsymbol{w}_v$. Fig.1(a) shows a weighted undirect graph which contains 4 vertices. The weight bag for vertex A is $\boldsymbol{w}_A = [w_{AB}, w_{AD}]$=[2,1] and the weight bag set for G is $W = \{[2,1],[3,2,1],[3,1],[1,1,1]\}$. Without considering edge weight, anonymized graph G' in Fig.1(b) can resist with previous proposed privacy attacks using graph topological structure as background knowledge, including degree[3], neighborhood[4], subgraph[5], etc. Adversary cannot identify any vertex in G' with confidence larger than 50%. However, when an adversary knows A's weight bag $\boldsymbol{w}_A = [2,1]$, the adversary can easily identify vertex 1 in G' as A, since only vertex 1 has the same weight bag as A, which causes an identity disclosure problem. In real social network **NetSci**[15], 6.48% of vertices could be uniquely identified based on weight bag information[1].



(a) G (b) G' (c) G_{HA} (d) G_{GA} (e) MPN

Fig. 1. Anonymization of weighted graph

1.1 Motivation

[1] proposes histogram anonymization(HA) to prevent from weight-based attacking. G_{HA} in Fig.1(c) shows an anonymized graph of G using histogram anonymization. After histogram anonymizing graph G into G_{HA}, for $\forall \boldsymbol{w} \in W_{HA}$, there are at least $k-1$ other weight bags that are the same with $\boldsymbol{w}$ and graph G_{HA} is defined as k-histogram anonymous. The weight bag set of G_{HA} is W_{HA} = $\{[3,2,1],[3,2,1],[3,2],[3,2]\}$, thus G_{HA} is 2-histogram anonymous and the vertices cannot be identified with confidence larger than 50% when weight bags are considered as the adversary's background knowledge. However, modifying edges and weights in histogram anonymization reduces the statistical utility of the anonymized graph. Considering graph query Q:

Q: SELECT COUNT($\forall edge \in G$) WHERE $edge.weight \geq 2$

The result of Q is 4 on graph G_{HA}, which is a little biased to the real answer 2 on G. Since there are a large amount of vertices and edges in weighted graphs, modifying edges and weights in histogram anonymization would cause estimates of a graph property to be systematically biased or highly variable, which will be introduced in the experimental section. Privacy preserving can also be achieved through increasing uncertainty of anonymized graph. Considering G_{GA} in Fig.1(d), $w_{12} = [1, 2]$ means w_{12} would be any value in the interval of [1,2], and "50%" on edge e_{13} denotes that e_{13} exists with the probability of 50%, thus $\boldsymbol{w}_1$ in G_{GA} can be formalized as [([1,2],100%),(1,50%),(1,100%)]. The uncertainty in G_{GA} is achieved through generalization, which is commonly adopted in privacy preserving techniques. If G_{GA} is an anonymized version of G in Fig.1(a), vertex 1 may correspond to A, since $\boldsymbol{w}_A$ is a possible instance of $\boldsymbol{w}_1$. We use *candidate set* in [10] to represent the vertices of G_{GA} that could feasibly correspond to *target* x, denoted cand(x). In graph G_{GA}, we have cand(A)=cand(D)={1,4},cand(B)=cand(C)={2,3}, thus an adversary cannot identify vertex in G_{GA} with confidence larger than 50%, which is the same as G_{HA}. Let's consider query Q on G_{GA}. In G_{GA}, edge e_{23} definitely satisfies Q, edges e_{12} and e_{34} possibly satisfy Q, and the other edges cannot satisfy Q, thus the result of Q on G_{GA} is an interval [1,3], which encloses the real answer 2 of Q. Generalization based graph anonymization can protect privacy in weighted graphs while preserving statistical properties and utilities.

1.2 Challenges

Although privacy preservation in social network has been studied extensively and several important models such as K-Degree[3], K-Isomorphism[5], K-Symmetry[7] and K-Automorphism[6] as well as many efficient algorithms have been proposed, most of the previous studies can deal with privacy preservation on simple graphs only. As example in Fig.1 shows, applying those methods to weighted graphs straightforwardly still causes privacy leakage.

In practice, we need to anonymize weighted social network before it is released. However, anonymizing weighted social network graphs is much more challenging than anonymizing simple graphs.

First, it is much more challenging to design privacy preserving model for weighted graphs. As shown in Fig.1, although anonymizing different weight bags into the same ones through modifying edges and weights can prevent from weight based attacks, statistical properties and utilities of weighted graphs are reduced.

Second, it is much more challenging to measure the information loss in anonymizing weighted graphs. Besides the information loss in each edge weight, the information loss of weight related statistical properties and utilities should also be measured. Anonymizing weighted graphs would affect the topological structures of the graphs which should be considered. Thus, there exists different ways to define the information loss and data utility for weighted graphs. Instead of measuring those information loss separately, all types of information loss should be measured in combination.

Last but not least, it is much more challenging to devise anonymization methods for weighted social networks than simple ones. One simple observation is that when the weight of an edge is modified or generalized, this operation would affect the two vertices that are connected with this edge. When devising anonymiztion approach for weighted social networks, both information loss on weights and graph properties should be as less as possible.

Our contributions can be summarized as follows. (1)We propose k-possible anonymity model to protect against weight-based attacks. (2)We prove that to achieve optimal k-possible anonymity is NP-hard. (3)We develop a generalization based anonymization approach to achieve k-possible anonymity, and design a number of techniques to make the anonymization process efficient while maintaining the utility. (4)Our extensive empirical studies show that our method performs well in anonymizing weighted graphs of real world.

The remainder of the paper is organized as follows. Related work are summarized in Section 2. In Section 3, we give the problem definition and formalize the k-possible anonymity model. The generalization based anonymization approach that ensures k-possible anonymity is investigated in Section 4. We evaluate our method in Section 5. Section 6 concludes the paper.

2 Related Work

There has been much research on privacy preserving in social network graphs. Backstrom et al. [2] firstly proposed the privacy problems in social network.

Most of existing research has focused on *graph anonymization*, which considers the topological structures of graph as adversary's background knowledge. One popular graph anonymization is to add and/or delete edges for modifying the topological structure of the graph in order to prevent re-identification in the anonymized graph. Liu et al. [3] studies graph anonymizing to prevent from privacy attacking using degree as background knowledge. Zhou et al. [4] provides identity privacy through anonymizing each vertex's neighborhood subgraph. In order to prevent privacy attacks using subgraph as background knowledge, Cheng et al. [5], Zou et al. [6] and Wu et al. [7] propose privacy models K-Isomorphism, K-Automorphism and K-Symmetry respectively for graph anonymization. Ying et al. [8] studies graph randomization while preserving the spectrum of the graph. Yuan et al. [9] gives a solution to satisfy different needs on privacy protecting level. Another main design of existing graph anonymization is based on clustering vertices into super vertices, which makes the vertices in a super vertex indistinguishable from each other. Based on vertices clustering, Hay et al. [10] and Campan et al. [11] study vertex anonymization to prevent re-identification. Zheleva et al. [12] formulate the problem of preventing sensitive edge disclosure in unweighted graph.

Although most of existing research considers anonymization problems for simple unweighted graphs, recent work [1,13,14] has focused on privacy preservation in weighted graphs. Das et al. [13] aims at preserving the linear properties of the graph while modifying edge weights. [14] perturbs edge weights within a

threshold that follows a probability distribution while retaining the cost of the shortest path. However, the threshold used in this approach is always small and the anonymized weights are close to the original ones, there still exists the possibility of privacy leaking. [1] considers re-identification in weighted graphs and proposes histogram anonymization to protect vertex privacy through modifying edge weights and vertex connections, which causes a great information loss on the utilities and statistical properties of weighted graphs.

3 Problem Definition

In this paper, we model a *social network* as an uncertain weighted graph $G = (V, E, W, P)$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, W is a set of weights between vertices in V, and $w_{uv} = w_{e_{uv}}$ denotes the weight of edge e_{uv}. P_e denotes the probability that edge e exists. For $\forall e \in E$, P_e is initialized with 100%.

Definition 1. *(edge set) For each vertex $v \in V$, the edge set for v is defined as $\boldsymbol{e}_v = \bigcup_{e_{vu} \in E} \{e_{vu}\}$.*

For instance, in Fig.1(d), the edge set for vertex 1 is $\boldsymbol{e}_1 = \{e_{12}, e_{13}, e_{14}\}$.

Definition 2. *(possible candidate) Graph G is anonymized into G_a through generalization, if weight bag $\boldsymbol{w}_v (v \in G)$ is a possible instance of $\boldsymbol{w}_u (u \in G_a)$, u is defined as a possible candidate of v, denoted $u \succeq v$.*

In Fig.1, for vertices $1 \in G_a$ and $A \in G$, $\boldsymbol{w}_1$=$[([1,2], 100\%), (1, 50\%), (1, 100\%)]$ and $\boldsymbol{w}_A = [2, 1]$, $\boldsymbol{w}_A$ is a possible instance of $\boldsymbol{w}_1$, thus vertex $1 \succeq A$.

Definition 3. *(k-possible anonymous) Graph G is anonymized into G_a through generalization, G_a is k-possible anonymous if there exist vertex groups $g_1, \ldots, g_m$ of G and $g'_1, \ldots, g'_m$ of G_a that satisfy $\bigcup_{i=1}^{m} g_i = V, \bigcup_{i=1}^{m} g'_i = V_a, g_i \bigcap g_j = g'_i \bigcap g'_j = \emptyset (i \neq j), |g_i| = |g'_i| \geq k (i \in [1, m])$ and $u \succeq v (\forall u \in g'_i, \forall v \in g_i)$, where $g'_i (i \in [1, m])$ is defined as anonymization group.*

In Fig.1, considering vertex groups $\{A, D\}$ and $\{B, C\}$ of G, $\{1, 4\}$ and $\{2, 3\}$ of G_a, since $1 \succeq A$, $1 \succeq D$, $4 \succeq A$, $4 \succeq D$, $2 \succeq B$, $2 \succeq C$, $3 \succeq B$ and $3 \succeq C$, G_a in Fig.1(d) is 2-possible anonymous.

Vertices in the same anonymization group g are indistinguishable from each other based on weight bags. Thus, if an anonymized graph G_a is k-possible anonymous, the adversary cannot re-identify each vertex $v \in G_a$ with confidence larger than $\frac{1}{k}$.

Problem 1. (*graph anonymization*) Given a weighted graph $G(V, E, W, P)$, and an integer k, find generalizations that anonymize G into a k-possible anonymous graph G_a, such that information loss $L(G, G_a)$ is minimized.

The information loss incurred by generalizations is formalized as Equation 1,

$$L(G, G_a) = \sum\nolimits_{e \in E_a} |maxW_e - minW_e| \quad (1)$$

where $[minW_e, maxW_e]$ denotes the weight interval assigned to edge e through generalization. Equation 1 is simple and clear to capture the information loss and uncertainties of the edge weights resulted from generalization.

Theorem 1. *(Complexity:) Achieve k-possible anonymity with minimal information loss is NP-hard.*

Proof. The proof is constructed by reducing the NP-hard optimal k-anonymity problem to k-possible anonymity with minimal information loss through mapping weight bag $\boldsymbol{w}_v (v \in V)$ into d_{max}-dimensional space, where d_{max} is the maximum degree of vertices in V. Due to the space limitation, we omit the details here.

4 Generalization Based Anonymization

In this section, we introduce an efficient heuristic algorithm as a solution to Problem 1. We achieve k-possible anonymity through generalization based anonymization. We adopt a two-step strategy in our graph anonymization method. It first try to generate anonymization groups with minimal information loss L, and then k-possible anonymity is achieved through edge generalization.

4.1 Generating Anonymization Groups

The first challenge is to generate anonymization groups for privacy preserving while retaining high data utility. We first introduce an efficient clustering-based grouping technique in Algorithm 1.

For each vertex v_i, we map its weight bag $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \ldots, w_{id_i}](w_{i1} \geq w_{i2} \geq \ldots \geq w_{id_i})$ into a point in m-dimension space, where d_i is the degree of v_i and $m = max(d_i)$,$i \in [1, n]$. For vertex v_i with degree $d_i < m$, corresponding weight bag $\boldsymbol{w}_i$ is expanded by filling $(m - d_i)$ zeros.

In Algorithm 1, Line 1 initializes the number of anonymization groups based on input k value, and then the centers of the groups are randomly picked in Line 3. Line 6 adds the closest k points to each group based on the Euclidean distances between points and group centers. After all points are added to anonymization groups, the center of each group is updated in Line 10. Here we adopt the mean of the points in each group as new center. Line 11 shows that Lines 5-10 are iteratively performed until the distance between old and new center of each group is smaller than predefined threshold ϵ. In our experiments, we adopt $\epsilon = 0$, which means clustering and center updating are performed iteratively until points in each group do not change. Although we set $\epsilon = 0$, Algorithm 1 still has an efficient convergence performance which is shown in our experiments.

Algorithm 1. Generate Anonymization Groups

Input: A weight bag set W and an integer k
Output: Anonymization group set AG

```
1  N = ⌊|W|/k⌋;
2  I=1;
3  Randomly pick N points as group centers AG_j(I), j ∈ [1, N];
4  repeat
5      for j = 1 to N do
6          Find the closest k points to AG_j(I) in W and add to AG_j;
7          W = W − AG_j;
8      Assign each remaining point in W to the closest group;
9      I = I + 1;
10     Update AG_j(I), j ∈ [1, N];
11 until ‖AG_j(I) − AG_j(I − 1)‖ ≤ ε, j ∈ [1, N] ;
12 return AG = {AG_1, ..., AG_N};
```

The computational complexity of Algorithm 1 is $O(t\frac{n^2}{2k})$, where n is the number of points, k is the size of anonymization group, and t is the number of iterations. Each iteration introduces $O(\frac{n^2}{2k})$ operations to calculate distances between points and group centers.

4.2 Edge Generalization

Edge generalization is the second stage of the two-step strategy, which aims to achieve k-possible anonymity through generalizing edges based on generated anonymization groups. We provide *Edge Generalization*(EG) method in Algorithm 2.

We take G' in Fig.1(b) as an example to demonstrate EG algorithm. For vertices in Fig.1(b), $AG = \{AG_1 = \{1, 4\}, AG_2 = \{2, 3\}\}$ is generated when $k = 2$ using Algorithm 1. In order to let each vertex in AG_i be possible candidates for all other vertices in AG_i after anonymization, edges of each vertex should be generalized. Firstly, Lines 4-5 generalize existing edges. $e_v(j)$ denotes the edge in $\boldsymbol{e}_v$ that has the j-th weight $w_v(j)$ of $\boldsymbol{w}_v$, where d_v is the degree of v and obviously $w_v(j) = w_{e_v(j)}$. $Min_{i,j}$ denotes $min_{u \in AG_i}(w_u(j))$, which is the minimum value of all the j-th weights of the weight bags in AG_i, and $Max_{i,j}$ denotes $max_{u \in AG_i}(w_u(j))$ analogously. For example, weight bags of vertices in AG_2 are $\boldsymbol{w}_2 = [w_{e23}, w_{e21}, w_{e24}] = [3, 2, 1]$ and $\boldsymbol{w}_3 = [w_{e32}, w_{e34}] = [3, 1]$, thus $Max_{2,2} = 2$ and $Min_{2,2} = 1$, respectively. Line 4 generalizes $w_{e_{21}}$ and $w_{e_{34}}$ into [1,2] as shown in Fig.1(d). Since edge weight w_{uv} occurs in both $\boldsymbol{w}_u$ and $\boldsymbol{w}_v$, w_{uv} would be generalized twice with different $[Min, Max]$ and weight interval of w_{uv} is continuously extended. For instance, generalize $w_{uv} = [1, 3]$ with $[2, 4]$ would extend w_{uv} into $[1, 4]$. Existence probabilities are updated in Line 5, which aims to preserve privacy and statistical properties, where $P_{i,j} = \frac{count_{u \in AG_i}(d_u \geq j)}{|AG_i|}$. Take $e_2(3) = e_{24}$ for example, Line 5 updates $P_{e_{24}}$ with $min\{P_{2,3}, P_{e_{24}}\} = min\{\frac{count_{u \in AG_2}(d_u \geq 3)}{|AG_2|}, P_{e_{24}}\} = min\{\frac{1}{2}, 100\%\} = 50\%$.

Algorithm 2. Edge Generalization

Input: Weighted graph $G = (V, E, W, P)$,
anonymization group set $AG = \{AG_1, \ldots, AG_N\}$
Output: k-possible anonymous G_a

```
1  for i = 1 to N do
2      for each v ∈ AG_i do
3          for j = 1 to d_v do                    /* Generalize existing edges */
4              Generalize w_{e_v(j)} with [Min_{i,j}, Max_{i,j}];
5              P_{e_v(j)} = min{P_{i,j}, P_{e_v(j)}};
6          for t = (d_v + 1) to maxDegree(v) do        /* Add new neighbors */
7              cand = null; pos = null; Cost_{cand,pos} = +∞;
8              for each u ∈ V with e_uv ∉ E and u needs new neighbor do
9                  for m = (d_u + 1) to maxDegree(u) do
10                     if Cost_{u,m} < Cost_{cand,pos} then
11                         cand = u; pos = m;
12             if cand == null then
13                 Find the most potential neighbor u ∈ V of v;
14                 cand = u; pos = d_u + 1;
               /* Create an edge between v and cand, where cand ∈ AG_c  */
15             Add new edge e_{v,cand} to E;
16             e_v(t) = e_cand(pos) = e_{v,cand};
17             Generalize w_{e_{v,cand}} with [Min_{i,t}, Max_{i,t}] and [Min_{c,pos}, Max_{c,pos}];
18             P_{e_{v,cand}} = min{P_{i,t}, P_{c,pos}};
19 return G;
```

After existing edge generalization, $\boldsymbol{w}_2$ is generalized into $[(3, 100\%), ([1, 2], 50\%), (1, 50\%)]$, which makes vertex 2 be a possible candidate of vertex 3.

Generalizing existing edges only cannot achieve k-possible anonymity, some vertices need to be connected with new neighbors. For instance, $\boldsymbol{w}_3$ is generalized into $[(3, 100\%), ([1, 2], 100\%)]$ after existing edges generalization, of which $\boldsymbol{w}_2$ is not a possible instance. In order to be possible candidate of vertex 2, vertex 3 needs to add a new neighbor. Function $maxDegree(v)$ in Line 6 returns the max degree of vertices in AG_i that v belongs to, and v needs $(maxDegree(v) - d_v)$ new neighbors in order to be possible candidate for vertices in AG_i. For vertex v that needs a neighbor, Lines 7-14 find a neighbor candidate *cand* and Lines 15-18 add an edge connection between v and *cand*. When we choose a neighbor candidate *cand* for v, vertices that also need new neighbors are considered with priority in Lines 8-11. Vertex u that causes the lowest information loss on e_{vu}, which is to be added to E, is picked as neighbor candidate *cand*. For instance, when we choose neighbor candidate for vertex 3, vertex 1 in AG_1 which also needs a new neighbor is picked as neighbor candidate of vertex 3. If Lines 8-11 did not find neighbor candidate, Lines 12-14 choose the most potential neighbor(MPN) of v to be neighbor candidate *cand*. We define the vertex u that has the most common neighbors with v and $e_{uv} \notin E$ as v's most potential neighbor. In Fig.1(e), vertex

2 is the most potential neighbor of vertex 1. When new edge $e_{v,cand}$ is created, Line 17 generalizes $w_{e_{v,cand}}$ with both weight intervals of v and *cand* in order to meet k-possible anonymity requirement. If *cand* is v's most potential neighbor, $[Min_{c,pos}, Max_{c,pos}]$ in Line 17 and $P_{c,pos}$ in Line 18 should be neglected.

The computational complexity of Algorithm 2 is $O(mn)$, where n is the number of vertices in V and m is the number of vertices that need to add new neighbors. For each vertex v that needs to add new neighbor, there are maximum n vertices to be v's neighbor candidates. Therefore, the total complexity is $O(mn)$. Although the worst case for Algorithm 2 is $O(n^2)$ when $m = n$, m is much less than n in real datasets as shown in our experiments.

5 Experiments

In this section, we evaluate our methods using two real data sets, **NetSci** and **Hep-Th**. All these graphs are weighted and undirected. **NetSci** describes a coauthorship network of scientists working on network theory and experiment. There are 1,589 scientists(vertices) and 2742 edges in this data graph, and each edge is assigned with a real weight as described in [15]. The **Hep-Th** database[16] extracts weighted network of coauthorships between scientists posting preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999, which contains 8361 vertices and 15751 edges.

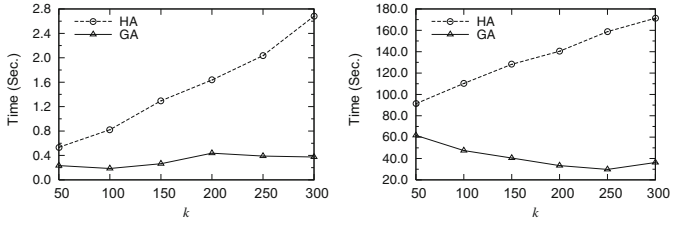
All the programs are implemented in Java. The experiments are performed on a 2.33GHz Intel Core 2 Duo CPU with 4GB DRAM running the Windows XP operating system.

We focus on the utility and statistical properties of the anonymized graphs. Here we use some weight related measurements to evaluate the utilities of the anonymized weighted graphs. (1)*Edge Weight* describes the distribution of all edge weights in the graph. (2)*Degree* is the distribution of the degrees of all vertices in the graph. (3)*Volume* is a distribution of the volumes of vertices in the network, where volume is the sum of a weight bag. (4)*Path length* is a distribution of the lengths of the shortest paths between 1000 randomly sampled pairs of vertices in the network.

We measured these characteristics for the original graph G, the graph G_{HA} anonymized through HA[1] and a set of 20 output graphs sampled from the graph G_{GA} which is anonymized through our proposed GA approach. For each edge e in G_{GA}, we sampled w_e uniformly from the generalized weight interval of e, and whether e exists in the sampled graph depends on the existence probability of e.

5.1 Runtime

Fig.2 shows the runtime of HA and GA on the two data sets with different k values. Our results show that GA requires less runtime than HA. The runtime of HA increases with the increment of k value, while the runtime of GA keeps stable and even decreases with the increment of k value. After anonymizing weight set, HA constructs the graph based on the anonymized weight set, while



(a) NetSci (b) Hep-Th

Fig. 2. Runtime for different k values

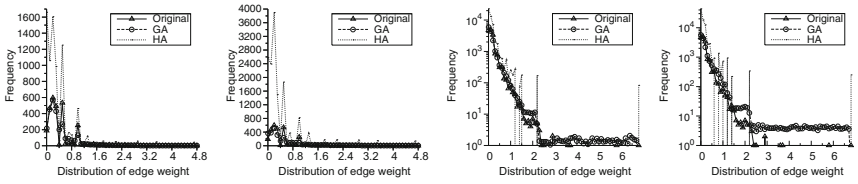
GA anonymizes on the original graph, thus HA costs more runtime than GA. In Fig.2(b), the runtime of GA decreases when k increases from 50 to 250, and increases when k increases from 250 to 300. This is for the reason that the size of anonymization group depends on k value, and different size of anonymization group would affect the performance of the clustering in generating anonymization groups. Anonymization group size which is suitable to the properties of the data set would result in efficient clustering.

5.2 Data Utilities

In this work, data utilities refer to statistical properties of the graphs. Figs.3 to 6 show the results of the experiments with respect to the four statistical characteristics for both data sets, where Original, GA, and HA denote the distributions for original graph, generalization based anonymization and histogram anonymization[1], respectively.

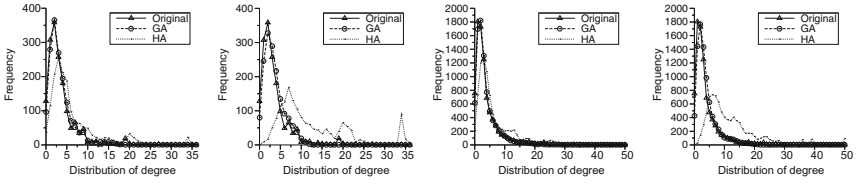
Fig.4 shows the distributions of degrees when $k = 20, 100$ for both data sets. When $k = 20$, Fig.4(a) and 4(c) show that the distributions of degree for GA are very similar to the original graph, and the distributions for HA are a little biased to the original graph. In Fig.4(b) and 4(d), when k increases to 100, the curves for GA and Original are still aligned, while the curves for HA are quite different. Similar results could be observed in Figs.3, 5 and 6.

Our results in Figs.3 to 6 demonstrate that GA protects privacy in weighted graphs while guaranteeing the accuracy of statistical analysis. When k value increases, GA still preserves the essential graph information, while the bias of the distributions for HA increases. This is for the reason that HA preserves privacy through modifying weights and edges, which results in more information loss when k increases. GA preserves privacy through anonymizing graphs using generalization, which preserves the properties and statistical utilities of original graph. Thus, the network measures of original graph could be recovered with little bias through aggregation on a small number of sampled graphs.



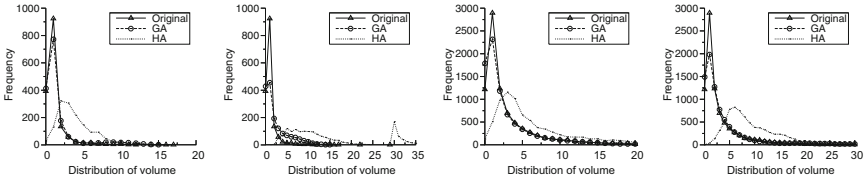
(a) NetSci, k=50 (b) NetSci, k=150 (c) Hep-Th, k=100 (d) Hep-Th, k=300

Fig. 3. Distributions of edge weights



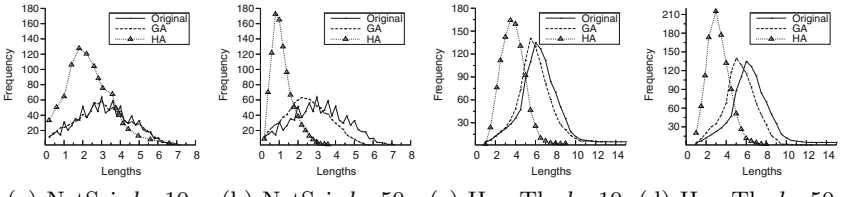
(a) NetSci, k=20 (b) NetSci, k=100 (c) Hep-Th, k=20 (d) Hep-Th, k=100

Fig. 4. Distributions of degrees



(a) NetSci, k=100 (b) NetSci, k=300 (c) Hep-Th, k=100 (d) Hep-Th, k=300

Fig. 5. Distributions of volumes



(a) NetSci, k=10 (b) NetSci, k=50 (c) Hep-Th, k=10 (d) Hep-Th, k=50

Fig. 6. Distributions of path lengths

6 Conclusion

Most of existing research work studies privacy preserving for simple graphs only, and cannot be applied to weighted graphs. In this paper, we focus on protecting privacy in weighted graphs. We propose a privacy preserving model, named

k-possible anonymity, to prevent from identity disclosure against weight-based attacks. We develop a generalization based anonymization approach(named GA) to achieve k-possible anonymity for a weighted graph, and design a number of techniques to make the anonymization process efficient while maintaining the utility. Extensive experiments on real datasets show that our method performs well in terms of protection it provides, and a wide range of weight related graph analysis can be performed accurately.

References

1. Li, Y., Shen, H.: Anonymizing graphs against weight-based attacks. In: International Conference on Data Mining Workshops(ICDMW 2010), pp. 491–498 (2010)
2. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore are thouR3579X?: Anonymized social networks, hiddern patterns and structural steganography. In: International World Wide Web Conference(WWW 2007), pp. 181–190 (2007)
3. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 93–106. ACM Press, New York (2008)
4. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE 2008), pp. 506–515 (2008)
5. Cheng, J., Fu, A.W.-C., Liu, J.: K-Isomorphism: Privacy preserving network publication against structural attacks. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD 2010), pp. 459–470 (2010)
6. Zou, L., Chen, L., Ozsu, M.T.: K-automorphism: A general framework for privacy preserving network publication. In: Proceedings of the 35th International Conference on Very Large Databases (VLDB 2009), vol. 2(1), pp. 946–957 (2009)
7. Wu, W., Xiao, Y., Wang, W., He, Z., Wang, Z.: K-Symmetry model for identity anonymization in social networks. In: Proceedings of the 13th International Conference on Extending Database Technology (EDBT 2010), pp. 111–122 (2010)
8. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: Proceedings of the 2008 SIAM International Conference on Data Mining (SDM 2008), pp. 739–750 (2008)
9. Yuan, M., Chen, L., Yu, P.S.: Personalized privacy protection in social networks. Proceedings of the VLDB Endowment 4(2), 141–150 (2010)
10. Hay, M., Miklau, G., Jensen, D., Towsley, D.: Resisting structural re-identification in anonymized social networks. In: Proceedings of the 34th International Conference on Very Large Databases (VLDB 2008), pp. 102–114. ACM, New York (2008)
11. Campan, A., Truta, T.M.: A clustering approach for data and structural anonymity in social networks. In: Bonchi, F., Ferrari, E., Jiang, W., Malin, B. (eds.) PinKDD 2008. LNCS, vol. 5456, pp. 33–54. Springer, Heidelberg (2009)
12. Zheleva, E., Getoor, L.: Preserving the privacy of sensitive relationships in graph data. In: Bonchi, F., Malin, B., Saygın, Y. (eds.) PinKDD 2007. LNCS, vol. 4890, pp. 153–171. Springer, Heidelberg (2008)

13. Das, S., Egecioglu, O., Abbadi, A.E.: Anonymizing weighted social network graphs. In: The 26th International Conference on Data Engineering, ICDE 2010 (2010)
14. Liu, L., Wang, J., Liu, J., Zhang, J.: Privacy preservation in social networks with sensitive edge weights. In: 2009 SIAM International Conference on Data Mining(SDM 2009), Sparks, Nevada, pp. 954–965 (April 2009)
15. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E, 0605087 (2006)
16. Newman, M.E.J.: The structure of scientific collaboration networks. Proc. Natl. Acad. Sci. USA 98, 404–409 (2001)