2013 10th Web Information System and Application Conference

Protecting Sensitive Labels in Weighted Social Networks

Ke Chen[1], Hongyi Zhang[2], Bin Wang[1], Xiaochun Yang[1]

[1]College of Information Science and Engineering
Northeastern University
Shenyang 110819, China
kechen11neu@gmail.com
{binwang, yangxc}@mail.neu.edu.cn,

[2]International School of Software
Wuhan University
Wuhan 430079, China
Lurenjia0417@qq.com

***Abstract*—With the popularity of social networks, data privacy preserving in social networks has become a hot issue among scholars. An attacker can use a variety of background knowledge to attack against privacy. Most of the present technology on anonymity weighted social network graphs can only deal with edge weight, but cannot be applied to sensitive labels. We consider a new generalization approach for sensitive labels, which can afford utility without compromising privacy. In this paper, we investigate the sensitive label privacy disclosure problem in weighted graph, propose *k*-histogram-inverse-*l*-diversity (KH-inv-LD for short) anonymity to protect sensitive label information, and develop a label anonymous approach to achieve this model. Extensive experiments on real data sets show that the algorithm performs well in terms of sensitive label privacy protection in weighted graph.**

Keywords—weighted social networks; privacy preserving; anonymous; data publishing

I. Introduction

In the study of social networks, the label information of the user can be used to describe the characteristics and interests of the user. Through the analysis of user label information, the personalized recommendation system (e.g. Netflix, YouTube and Amazon recommend movies, videos and products respectively) can predict the user's preference and recommend interesting resources to help users find the effective information, which can reduce the search time. However, the implicit information in the label resource can lead to the leakage of user’s privacy. Therefore, it has been a challenging work to protect user privacy at the same time, as much as possible to release the label information in social network.

Fig.1 (a) shows a weighted undirected graph containing no nodes’ labels. Recently, much work on weighted social networks privacy preservation has been developed, which consider edge weight as attackers’ background knowledge. However, in many applications (e.g. recommendation system), each node’s label should be published to describe the real world social networks. Prior techniques in edge weight preservation cannot protect sensitive label effectively.

A. Motivation

[1] considered the neighborhood weight distribution of a node, and proposed *k*-histogram anonymous model: a graph is *k*-histogram if there are at least *k*-1 other nodes in the graph with the same histogram. Subsequently, we will explain *k*-histogram in more detail. Table.1 (a) and (b) show the corresponding weight histogram distribution of Fig.1 (a) and Fig. (b). From Table.1 (b), we know that the weight histogram <4, 3, 3, 2, 1> correspond directly with node A and B. Likely, each of <5, 3>, <5, 2>, <4>, <3> corresponds to two nodes respectively. The graph G_1 in Fig.1 (b) is 2-histogram anonymous, which doesn’t take node labels into account. An attacker cannot identify any nodes in G_1 with probability larger than 50%. However, the node label information can reveal the node privacy as same as the weight histogram. Fig.2 (a) shows the weighted and labeled graph of G_1. If an attacker knows Alice’s sub-graph structure, as example Fig.3 (a) shows, he is able to determine Alice in the highest monthly salary 100K. Similarly, it’s easy to determine David in the lowest monthly salary 10K, which causes the sensitive label disclosure problem.

B. Challenges and Contributions

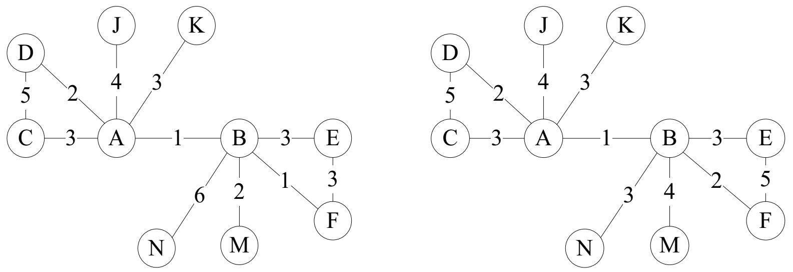
Privacy preservation in weighted social network has been studied. However, most of the prior studies only focus on edge weight privacy leakage and cannot protect sensitive label effectively. Compare with anonymizing edge weight in social networks, anonymizing node label is much more challenging.

We study the main challenges and make the following contributions:

- We first discuss the sensitive label information leakage problem in weighted social networks.
- We propose *k*-histogram-inverse-*l*-diversity (KH-inv-LD) model to protect sensitive label.
- We design a generalization approach for the KH-inv-LD model, which can achieve enough privacy protection and accurate data analysis.
- We firstly consider the multi-sensitive-attribute privacy preserve in weighted social networks and present an efficient algorithm to protect the sensitive labels.
- We also demonstrate that our approach provides efficient result and performs well in protecting sensitive labels by extensive experiments.

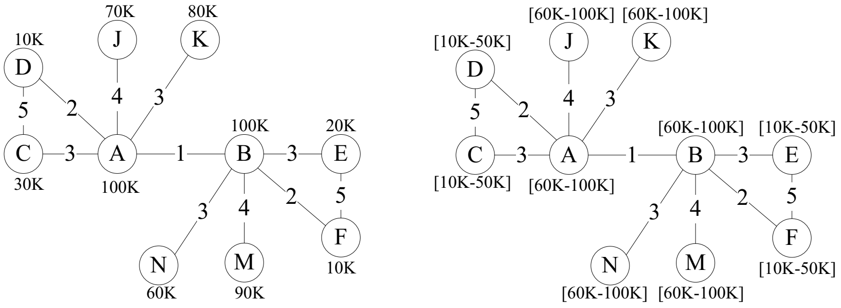
This work is partially supported by the National Basic Research Program of China (973 Program) (No. 2012CB316201), the National Natural Science Foundation of China (Nos. 61173031, 61272178), the Joint Research Fund for Overseas Natural Science of China (No. 61129002), the Doctoral Fund of Ministry of Education of China (No. 20110042110028), and the Fundamental Research Funds for the Central Universities (Nos. N120504001, N110404015).

978-1-4799-3219-1/13 $31.00 © 2013 IEEE
DOI 10.1109/WISA.2013.50



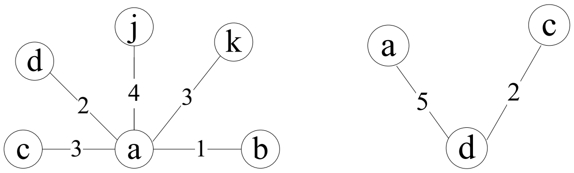
(a) Original graph G_0 (b) 2-histogram graph G_1

Fig. 1. Example of edge weight protection.



(a) Original graph G_0 (b) 2-histogram graph G_1

Fig. 2. Example of sensitive label protection.



(a) Alice's subgraph (b) David's subgraph

Fig. 3. Possible background knowledge of an attacker.

TABLE I. WEIGHT DISTRIBUTION IN WEIGHTED GRAPH.

Node	Weight Histogram
A	<4,3,3,2,1>
B	<6,3,2,1,1>
C	<5,3>
D	<5,2>
E	<3,3>
F	<3,1>
J	<4>
K	<3>
M	<2>
N	<6>

(a) G_0's Distribution

k=2

Histogram Anonymization

Node	Weight Histogram
A	<4,3,3,2,1>
B	<4,3,3,2,1>
C	<5,3>
E	<5,3>
D	<5,2>
F	<5,2>
J	<4>
M	<4>
K	<3>
N	<3>

(b) G_1' Distribution

II. RELATED WORK

Since [2] put forward the privacy problem in social network, many techniques about social networks privacy protection have been proposed. In recent years, Edge weight protection has caused concern in most of existing work. Das et al. [3] proposed a linear programming model to anonymous edge weight while preserving properties of graph data. The work in [4] discussed the perturbation of some edges' weights while keeping the shortest path between some pairs of nodes. [5] proposed a generalization-based approach to preserve identity and edge weight disclosure. However, most of the existing researches we mention above only focus on un-labeled weighted graphs.

The research in [6] discussed the privacy preserving problem in social networks. They allowed users to set personalized privacy. Song et al. [7] presented node labels as sensitive information and designed privacy protection algorithms in graph data publishing. Yang et al. [8] firstly discusses the problem of secure publishing data when sensitive data contains multiple attributes. The work in [9] put "Semi-Edge Anonymity" as a new protection to protect sensitive attributes and links information. Bhagat et al. [10] presented a clustering-based model which allowed rich data compression. The work in [11] defined a *k*-degree-*l*-diversity anonymity model to protect label-node relationship. Although most of the prior studies focus on label anonymity, they have ignored the sensitive label disclose in weighted social networks.

III. PROBLEM DEFINITION

In this paper, we define a social network as a labeled weighted graph $G = (V, E, W, L, \lambda)$, 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, L is a set of sensitive labels on vertices, and $\lambda : V \rightarrow L$ is a mapping function that maps vertices to their labels. For each vertex $v \in V$, we sort the sequence of weights in descending order on all edges connecting v to define weight bag. "Vertex" and "node" are often used interchangeably in this paper, the same as "graph" and "network".

Definition 1. (*Vertex sensitive label*) *For each vertex v in graph G, we use SL_v = (S_1, S_2, …, S_r) as sensitive label belonging to v.* S_j ($1 \le j \le r$) are *sensitive attributes such as Disease and Salary that are assumed to be unknown to an attacker.*

In Fig 2. (a), for vertex $A \in G_2$, SL_A = (100K) is the highest salary among other labels, which is private information of the vertex. 100K is the sensitive attribute in the graph which is also sensitive attribute in relational data. In fact, the node label includes not only sensitive attributes, but also identifier attributes and quasi-identifier attributes. For simplicity, we assume that nodes are labeled with sensitive attributes only in this paper.

Definition 2. (*Graph k-histogram anonymous*) *If graph G is k-histogram anonymous, for each node v in graph G, there exist k*-1 *or more nodes that have the same weight bags with v.*

For instance, Fig 1. (b) shows the 2-histogram anonymous graph. Node *A*'s weight bag is <4, 3, 3, 2, 1>, and node *B*'s weight bag is also <4, 3, 3, 2, 1>, the same as *C* and *E* etc.

Definition 3. (*KH-inv-LD anonymity*) *Give an anonymous graph G^* of the original graph G, for every node v in G^*, there are existing at least k*-1 *other nodes that have the same weight bags. Furthermore, for each generalization label that associates with a node, there must be at least l*-1 *other nodes with the same label in the k-histogram anonymous group, where $l \le k$ is the constraint condition.*

Fig 2. (b) shows the 2-histogram-inverse-2-diversity anonymous graph. We use inverse-*l*-diversity to ensure that every generalization label has at least *l* distinct histogram anonymous nodes. This notion is a bit like *l*-diversity [12] in relational data, but not exactly the same. The *l*-diversity model guarantees that there are at least *l* distinct labels in each anonymous group which modify the label values without considering the data utility. In this paper, we try to retain the utility of the data as much as possible while protecting the sensitive information. So we design corresponding algorithms to achieve this goal.

IV. Generalization Based Anonymization

In this section, we describe how to protect sensitive labels for the individuals by generalization methods. We adopt two steps to solve this problem. First, the *k*-histogram anonymous groups will be generated to prevent weight bag information leakage. Then, we use sensitive attributes generalization strategy to achieve inverse *l*-diversity. We also consider the multiple sensitive attributes in this strategy. Next, we will explain how to implement these two steps.

A. k-histogram Anonymous Groups

In order to prevent disclosure of weight bag information, we produce the *k*-histogram anonymous groups. For every vertex v_i in V ($\bigcup_{i=1}^{m} v_i = V$), its weight bag w_i= [w_{d1},w_{d2},...,w_{di}] can be mapped into *d*(*max*)-dimension space to *d*(*max*) is the maximum degree of v_i($i \in$[1,*m*]). If vertex v_i's degree $d(v_i) < d(max)$, then we set its weight bag w_i by filling ($d(max)$- $d(v_i)$) zeros.

In Algorithm 1, we describe how to produce the *k*-histogram anonymous groups. After mapping weight bag to *d*(*max*)-dimension space, we sort the degree sequence of vertices in descending order. Considering about data utility, we set vertices with same degree into the anonymous group without adding noise nodes. After we have formed these groups, we need to guarantee that the members of each group are indistinguishable in terms of the weight bag information. Therefore, vertices' weight bags are modified after finding the *k* vertices with the same degree. The objective of modification is simply to make sure that all nodes have the same weight bag in the *k*-histogram anonymous group.

Algorithm 1：Generate *k*-histogram anonymous groups

```
Input：   weight bag set $W$, parameter $k$;
Output:   $k$-histogram anonymous group set $KHG$;
mapping $W$ to $d(max)$-dimension space;
sort degree sequence;
for $i \leftarrow 1$ to $d(max)$ do
    select $k$ vertices with $d_i$;
    compute $k$ weight bag and modify $w_1$=$w_2$,…,=$w_k$;
    $W_i \leftarrow \{w_1,w_2,…,w_k\}$;
    update $KHG_i$ with $W_i$ ;
update $KHG$ with $KHG_i$, $i \in [1,d(max)]$;
return $KHG$;
```

B. Generalization Tree

After generating the *k*-histogram anonymous groups, we should protect the sensitive labels of nodes. In the rest of this section, we use generalization tree to achieve this goal. Let *D* be a finite attribute domain. *GT* is a tree structure, where the leaf nodes contain the attribute value of the *D* domain and non-leaf nodes contain the generalization value of their sub-tree nodes. For numerical sensitive attribute $N_a \in SL_v$, $Gen(N_a)$=the interval [min{$S_1(N_a)$, $S_2(N_a)$, …, $S_r(N_a)$}, max{$S_1(N_a)$, $S_2(N_a)$, …, $S_r(N_a)$}]. For categorical sensitive attribute $C_a \in SL_v$, the generalization information $Gen(C_a)$ = the lowest common ancestor in generalization tree of {$S_1(C_a)$, $S_2(C_a)$, …, $S_r(C_a)$}. $S_j(N_a)$ and $S_j(C_a)$ ($1 \le j \le r$) are the leaf node in *GT*. The *salary* sensitive attribute is numerical and *disease* is categorical.

C. Single-sensitive-attributes Generalization of Node Labels

First, we use full-domain generalization [13] to map the values in the domains of sensitive attributes to other values. Full-domain generalization maps the whole domain of every sensitive attributes to its generalization tree, which guarantees all values of sensitive attributes in the same domain. However, we need to consider the data utility when publish the graph data. The generalization method must disturb the original data as little as possible. Therefore, we introduce an efficient solution named split-domain generalization to anonymize attribute data while achieve the data utility. Our algorithm makes the nodes with similar sensitive attributes into one group, which can maintain the utility after generalization. For numerical sensitive attributes N_{a1} and N_{a2}, the similarity of the two attributes is defined as follow:

$$Sim\ (N_{a1}, N_{a2}) = \sqrt{\frac{(|MaxValue - MinValue| - |N_{a1} - N_{a2}|)^2}{(MaxValue - MinValue)^2}} \quad (1)$$

where |*MaxValue*-*MinValue*| denotes the domain value interval of numerical attributes. For categorical sensitive attributes C_{a1} and C_{a2}, we calculate similarity with:

$$Sim\,(C_{a1}, C_{a2}) = \frac{|C_{a1} \cap C_{a2}|}{|C_{a1} \cup C_{a2}|} \qquad (2)$$

Algorithm 2: Single-sensitive-attribute generalization(SSAG)

```
Input:  KHG with single-sensitive labels ,parameter $l$;
Output: KH-inv-LD anonymous $G^*$;
1  for each $KHG_i$ in $KHG$ do
2    if the number of $KHG_i \geq l$ then
3      $N \leftarrow \lfloor |KHG_i|/l \rfloor$;
4      sort vertices in $KHG_i$ with $SL$;
5      split domain into $N$ groups with max similarity;
6      select $N$ vertices as the starting element $P_N$;
7      if $SL \in N_a$ then
8        for $i \leftarrow 1$ to $N$ do
9          compute top $l$-1 with max similarity $P_i$;
10         assign [$maxValue$, $minValue$] to labels;
11     else if $SL \in C_a$ then
12       for $i \leftarrow 1$ to $N$ do
13         compute top $l$-1 with max similarity $P_i$;
14         assign $Gen(C_a)$ to labels;
15   modify nodes labels in $KHG_i$;
16   update $G^*$ with $KHG_i$;
17 Return $G^*$;
```

In the SSAG algorithm, we compute the number of inverse-*l*-diversity groups based on input *l* value in Line 3. Then we sort the vertices in KHG_i anonymous group based on the sensitive attribute similarities. After all vertices are sorted, the split-domain generalization method splits the full-domain into *N* groups, and then randomly selects *N* vertices as the starting elements (i.e. the centers of each group) in Lines 5 and 6. After we form these groups, we must guarantee that every group's vertices are indistinguishable in terms of the label information. Thus, we use generalization method to make sure that each group has at least *l* vertices which have the same labels. For numerical sensitive attributes of labels, we find the *l*-1 vertices with the maximum similarities by using equation (1) and generalize the attribute values with interval in Lines 7-10. For categorical sensitive attributes, Lines 11-14 show the generalization process. The KH-inv-LD anonymous G^* is updated in Line 16.

D. *Multi-sensitive-attribute Generalization of Node Labels*

While some nodes have multiple sensitive attributes, generalizing single sensitive attribute only is not enough. Most of existing privacy preserving researches in social networks have focused on anonymity with only one sensitive attribute. As far as we are concerned, there are a lot of multiple sensitive attributes in real-world applications. We firstly consider the multi-sensitive-attribute privacy preserve in weighted social networks and present an efficient algorithm to protect the sensitive labels.

Algorithm 3: Multi-sensitive-attribute generalization(MSAG)

```
Input:  $KHG$ with multi-sensitive labels , parameter $l$;
Output: $KH$-$inv$-$LD$ anonymous $G^*$;
1  for each $KHG_i$ in $KHG$ do
2    if the number of $KHG_i \geq l$ then
3      $N \leftarrow \lfloor |KHG_i|/l \rfloor$;
4      for $S_j \in SL$ $(1 \leq j \leq r)$ do
5        compute $C(S_j)$ as the number of distinct values
6        set $CS \leftarrow \{|C(S_j)|,\ldots,|C(S_1)|\}$ in descending
         order ;
7      while $|CS|>0$ do;
8        sort vertices in $KHG_i$ with $|C(S_j)|$ in descending;
9        split domain in $S_j$ into $N$ groups with
         max similarity;
10       select $N$ vertices as the starting element $P_N$;
11       $|CS| \leftarrow |CS|$-1;
12       $j \leftarrow j$-1;
13     generalize $SL$ in $KHG_i$;
14   update $G^*$ with $KHG_i$;
15 Return $G^*$;
```

In MSAG algorithm, we also compute the number of inverse-*l*-diversity groups based on input *l* value in Line 3. The MSAG algorithm differs from the SSAG algorithm which needs to compute the number of distinct values in each sensitive attribute. Grouping sequence is determined by the number of distinct values. When the sensitive attribute is modified or generalized, this operation would affect the data utility, so we try to make the information loss on generalization labels as less as possible. One simple observation is that the more distinct values the group has, when generalization performs, the more information it will lose. Thus, in Line 8, we firstly sort the attribute which has the maximum number of distinct values among all attributes belong to the nodes in *KHG* anonymous group. Then we sequentially sort for other attributes according to the number of distinct values in descending order. We try to make sure that the nodes with the maximum similarities are in the same group as much as possible. Only after all the preliminary operations are performed can we make less information loss. The main techniques adopted for attributes generalization in Line 13 are the same as algorithm SSAG. The KH-inv-LD anonymous graph G^* is updated in Line 14.

In Table 2, we show the example of multiple sensitive attributes generalization process in one *k*-histogram anonymous group, where the number of sensitive attributes is 2(i.e. *Salary* and *Disease*). First, we calculate the number of distinct values in *Salary* and *Disease*. We get *C*(*Salary*) = 8 and *C*(*Disease*) = 4, so we sort the nodes of Table.3 (a) in descending order of *Salary*. Then we split the domain into 5 groups to make sure that each group has at least 2 nodes in terms of $l = 2$. After such splitting operations, we try to generalize the first sensitive attribute *Salary*, which is the same as SSAG algorithm. Next, we assign the generalization value to the second attribute *Disease*. Since the number of distinct values in *Disease* is less than *Salary*, each group has the same *Disease* values as much as possible. When we generalize the attribute, the greater the similarity among the values is, the less the information loss is. Meanwhile, an attacker can not accurately identify the sensitive attributes *Salary* and *Disease*.

The computational complexity of Algorithm 3 is $O(k \cdot r \cdot n^2 / l)$, where k is the number of k-histogram anonymous group, r is the number of sensitive attributes, n is the number of vertices in k-histogram anonymous groups, and l is the KH-inv-LD model's constraint condition. At the experimental evaluation section, we will show that the Algorithm 3 performs well in the real data sets.

TABLE II. WEIGHT DISTRIBUTION IN WEIGHTED GRAPH.

Node	Label
A	100K flu
B	100K flu
C	30K cancer
E	20K ulcer
D	10K ulcer
F	10K cancer
J	70K cancer
M	90K flu
K	80K pneumonia
N	60K cancer

l=2 →

Group	Generalization Label	Node
1	[90K-100K] respiratory	A
1	[90K-100K] respiratory	B
2	[80K-90K] respiratory	M
2	[80K-90K] respiratory	K
3	[60K-70K] digestive	J
3	[60K-70K] digestive	N
4	[20K-30K] digestive	C
4	[20K-30K] digestive	E
5	[10K-20K] digestive	D
5	[10K-20K] digestive	F

(a) Original labels Distribution (b) Generalized labels Distribuiom

E. Information Loss

The k-histogram anonymous and generalization of sensitive labels' attributes reduce the data utility. To measure the amount of information loss, we introduce (3),

$$IL(G,G^*) = \sum_{v \in G, v \in G^*} \frac{\left| W_{bag}(v^*) - W_{bag}(v) \right|}{Size(W_{bag}(v))} + \sum_{sl \in G, gl \in G^*} \frac{Height(GT(sl,gl))}{Height(GT)} \quad (3)$$

Where $\left| W_{bag}(v^*) - W_{bag}(v) \right|$ denotes changes of each weight in the weight bag, $Size(W_{bag}(v))$ is the number of weights in v's weight bag. $Height(GT(sl,gl))$ denotes the absolute value of the level difference between original *sl* and generalized *gl* in the generation tree and $Height(GT)$ is the height of generalization tree. For instance, if we set v_1's weight bag $W(v_1)$ = <6, 3, 2, 1, 1> and sensitive label *sl* = (*flu*), where $v_1 \in G$, $sl \in G$. After generating KH-inv-LD anonymous graph G^*, we get v_2 ($v_2 \in G^*$) corresponding vertex v_1, where $W(v_2)$ = <4, 3, 3, 2, 1> and generalization label *gl*=(*respiratory*). We compute weight variation in every dimension of weight bag, so we can get,

$$| W_{bag}(v_1) - W_{bag}(v_2) | = \sum_{i=1}^{5} | w(v_1) - w(v_2) | = (2+0+1+1+0) = 4 \cdot$$

The weight bag size of v_1 is 5. From generalization tree, we know that *flu* is in layer 3 and *respiratory* is in layer 2 of the generalization tree, so we can get $Height(GT(sl,gl))$ = |3-2| = 1. In addition, the height of the generalization tree is 3. Therefore, the information loss $IL(v_1,v_2)$ = 4/5+1/3 = 1.13.

V. EXPERIMENTAL EVALUATION

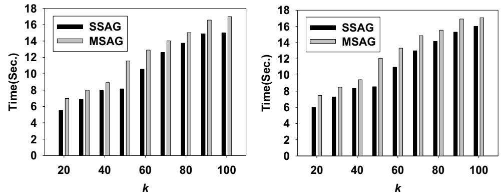
In this section, we use two real datasets [14] Hep-Th and Cond-Mat-2005 to evaluate our methods. All the algorithms are implemented in Java. The experiments run with a 3.00GHz Intel Core 2 Duo CPU and 2GB of main memory on Windows XP operating system.

A. Data Sets

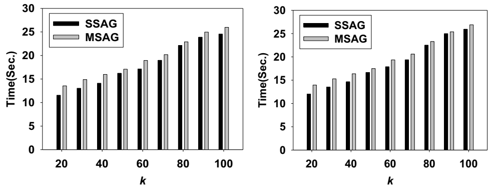
The Hep-Th dataset contains 8361 nodes and 15751 edges, which collects weighted collaboration network of scientists posting preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999. The Cond-Mat-2005 also describes a weighted network of coauthor-ships between scientists posting preprints on the Condensed Matter E-Print Archive. This version includes all preprints posted between Jan 1, 1995 and March 31, 2005, which contains 40421 nodes and 175692 edges. Every node represents an author and an each edge denotes co-author relationship between two authors. The edge weight means the number of papers co-authored by the two nodes. Each node has one label and each label have three attributes. The three attributes contain the name of the author, the number of the published papers and the research field of the author. The author's name is the identifier attribute that can be used to identify an individual, so we will remove this attribute when we publish the graph data. In this paper, we set the number of papers and research field as the sensitive attributes.

B. Runtime

Fig.4 shows the runtime for different k values on Hep-Th and Cond-Mat-2005. The runtime increases with the k values. This is for the reason that the size of the anonymous group increases with the k values, and the size of anonymous group will affect the attributes generalization. Fig.5 shows the runtime for different parameter l values on two datasets. We can see that the runtime of SSAG and MSAG increase with the increment of l value. However, as time increases SSAG and MSAG change not obviously. This is due to the same amount of dataset, and l does not affect the number of generalization data. As a result, the value of l has little effect on the execution time of the algorithm. In addition, MSAG's processing time is more than SSAG, because MSAG deals with multi-attribute.

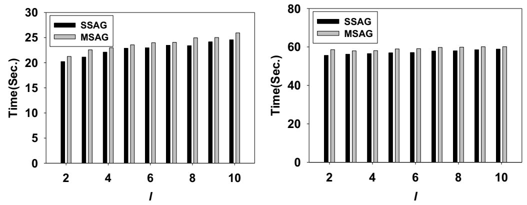


(a) Hep-Th, l = 2 (b) Hep-Th, l = 10



(c) Cond-Mat-2005, *l* = 2 (d) Cond-Mat-2005, *l* = 10

Fig. 4. Runtime for different *k* values.



(a) Hep-Th, *k* = 300 (b) Cond-Mat-2005, *k* = 300

Fig. 5. Runtime for different *l* values.

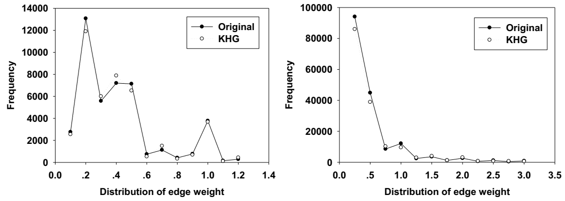
C. Data Utility

Fig.6 shows the distributions of edge weights for two datasets. Our results demonstrate that the distribution of edge weight in *KHG* is similar to the original graph. We consider the data utility, so we don't add noise node into anonymous group when we anonymize the weight bag.

We also calculate the label query error rate to evaluate the utility of the anonymous data. The query error rate is formalized as equation (4),

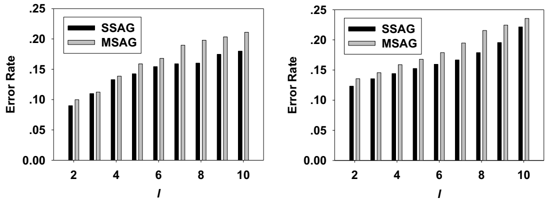
$$R_Q = \sqrt{(q_o - q_g)^2} \,/\, q_o \tag{4}$$

Where q_o is the number of query value in the original data and q_g denotes the number of query value in the generalization data. We set the error range of numerical attributes ± 10 in the query of generalization data. In addition, the error range of categorical attributes is that the generalization value in the generalization tree must be the parent node of query values. For example, we want to know how many scientists who study in the field of *functional nanomaterial* and publish 78-122 papers. We get 159 in the original graph and 201 in the generalization graph within the error range. So the query error rate is 0.26. We randomly pick 10 sets of query values for testing and compute the average query error rate. Fig.7 shows the average query error rate in our anonymous graph. The error rate of our algorithm is acceptable. Since we consider the information loss when we generalize the sensitive attributes, the average query error rate is small even when *l* is up to 10.



(a) Hep-Th, *k* = 50 (b) Cond-Mat-2005, *k* = 300

Fig. 6. Distribution of edge weight.



(a) Hep-Th, *k* = 300 (b) Cond-Mat-2005, *k* = 300

Fig. 7. Error rate for different *l* values.

VI. Conclusion

In this paper, we model a social network as a labeled weighted graph and develop techniques to protect sensitive labels. We first propose *k*-histogram-inverse-*l*-diversity model to protect sensitive labels while maintaining the data utility. Extensive experiments on real datasets show that the algorithm performs well in terms of sensitive label privacy protection in labeled weighted graph.

References

[1] Li Y, Shen H. Anonymizing graphs against weight-based attacks[C]//Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE, 2010: 491-498.

[2] Backstrom L, Dwork C, Kleinberg J. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography[C]//Proceedings of the 16th international conference on World Wide Web. ACM, 2007: 181-190.

[3] Das S, Egecioglu O, El Abbadi A. Anonymizing weighted social network graphs[C]//Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, 2010: 904-907.

[4] Liu L, Wang J, Liu J, et al. Privacy preservation in social networks with sensitive edge weights[C]//2009 SIAM International Conference on Data Mining (SDM 2009), Sparks, Nevada. 2009: 954-965.

[5] Liu X, Yang X. A generalization based approach for anonymizing weighted social network graphs[M]//Web-Age Information Management. Springer Berlin Heidelberg, 2011: 118-130.

[6] Yuan M, Chen L, Yu P S. Personalized privacy protection in social networks[J]. Proceedings of the VLDB Endowment, 2010, 4(2): 141-150.

[7] Song Y, Karras P, Xiao Q, et al. Sensitive label privacy protection on social network data[C]//Scientific and Statistical Database Management. Springer Berlin Heidelberg, 2012: 562-571.

[8] Yang X, Wang Y, Wang B, et al Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing. Chinese Journal of Computers, 2008, 31(4): 574-587.

[9] Yuan M, Chen L. Semi-Edge anonymity: graph publication when the protection algorithm is available[C]//Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2012: 367-381.

[10] Bhagat S, Cormode G, Krishnamurthy B, et al. Class-based graph anonymization for social network data[J]. Proceedings of the VLDB Endowment, 2009, 2(1): 766-777.

[11] Yuan M, Chen L, Philip S Y, et al. Protecting Sensitive Labels in Social Network Data Anonymization[J]. IEEE Transactions on Knowledge and Data Engineering, 2013: 633-647.

[12] Machanavajjhala A, Kifer D, Gehrke J, et al. l-diversity: Privacy beyond k-anonymity[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2007, 1(1): 3.

[13] LeFevre K, DeWitt D J, Ramakrishnan R. Incognito: Efficient full-domain k-anonymity[C]//Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, 2005: 49-60.

[14] Newman M E J. The structure of scientific collaboration networks[J]. Proceedings of the National Academy of Sciences, 2001, 98(2): 404-409.