DEIM Forum 2013 E4-3

An Improved Label-bag based Graph Anonymization based on Utility

李　重頡† 天笠　俊之† 北川　博之† GautamSrivastava††

† 筑波大学大学院システム情報工学研究科　〒 305–8573　つくば市天王台 1-1-1
†† Dept of Computer Science, University of Victoria
Victoria, BC Canada
E-mail: †lichongjie@kde.cs.tsukuba.ac.jp, ††amagasa@cs.tsukuba.ac.jp, †††kitagawa@cs.tskuba.com,
††††gsrivast@uvic.ca

An Improved Label-bag based Graph Anonymization based on Utility

Chongjie LI†, Toshiyuki AMAGASA†, Hiroyuki KITAGAWA†, and Gautam SRIVASTAVA††

† Graduate School of Information Engineering, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305–8573 Japan
†† Dept of Computer Science, University of Victoria
Victoria, BC Canada
E-mail: †lichongjie@kde.cs.tsukuba.ac.jp, ††amagasa@cs.tsukuba.ac.jp, †††kitagawa@cs.tskuba.com,
††††gsrivast@uvic.ca

Abstract Privacy concerns in publishing graph data, such as social-network graphs, have been gaining much public attentions in recent years due to the growing demands for publishing graph data containing privacy. So far, there have been lots of researches focusing on a labeled graph anonymization problem that is more applicable and since more difficult than that in unlabeled situations. In this paper we address the K-anonymity problem in edge-labeled graphs based on the label-bag model. We provide efficient greedy algorithms and evaluate them by experiments on both synthetic and real data. Further we extend the algorithm considering other utility measurements and show that our algorithm can be applied to varied utility metrics.

Key words Graph Privacy, K-anonymity, Label-bag, Utility

1. Introduction

Social networks (SNs) have shown remarkable development in recent years. Due to the rapid proliferation of SNs, there is a growing concern in privacy. SN data publishing is one of the main channels for privacy breaches. SN data owners, such as Facebook and LinkedIN, sometimes have responsibility to publish their data for various purposes. In this case, transforming data in such a way that privacy information is not released is important. Such data transformation is called "anonymization".

In many cases, SN data is represented as a graph $G = (V, E)$, where vertices (V) represent entities and edges (E) represent relationship between them. To model more complex information, a graph may have labels on vertices and/or edges that describe the attributes of entities or properties as shown in Figure 1.

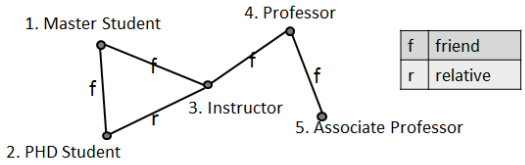

Figure 1 Social Network Graph

By having access to this graph, adversaries, with their existing knowledge about some involved entities (i.e. Bob has 3 friends), can gain more information such like that node 3 is representing Bob and he is an instructor (since node 3 is the only node with degree 3 in this graph). To prevent

the privacy of the entities from being violated, an appropriate graph anonymization method is needed to sanitize the original graph before publishing.

Varied anonymization methods have been proposed to retrieve the privacy requirement and to realize it with minimum extra information added becomes a challenging task. The concept utility is then proposed to measure the information loss during data publishing. Therefore most researches aim to solve the anonymity problem with the least information loss, or the largest utility, within extent privacy level.

In this paper we address the K-anonymity problem of edge-labeled graphs based on the label-bag model. Since it has been shown that the problem is NP-hard [4], we provide heuristic methods based on edge-addition in which we try to include the utility measurement as well. Additionally, we evaluate the effectiveness of our proposed scheme in some experiments.

The rest of this paper is organized as follows. Section 2 introduces some related works and we formalize the problem in Section 3. Section 4 gives the proposed methods and some discussion of utility measurement is involved in Section 5. We show the results and analysis of experiments in Section 6 and make a conclusion in Section 7.

2. Related Work

Privacy-preserving graph publishing is to transform a graph into another in such a way that adversaries with certain background knowledge cannot get the identity and cannot re-identify the entities and/or relations. So far, there have been lots of methods proposed to address the graph anonymization problem.

This problem can be categorized into several classes according to the graph model and quantity of knowledge that adversaries are assumed to have. The main part of researches are focusing on anonymizing structural information. These methods can be roughly divided to two categories: unlabeled graph anonymization [2] [3] [6] [10] [11] [12] [13] and labeled graph anonymization [4] [7]. Labeled cases can be further divided according whether vertices or edges (or both) are labeled. Most early works focus on unlabeled or vertex-labeled cases. Sweeney [2] gave the early idea of K-anonymity by replacing the identifiers of published data. Liu [3] defined the K-degree anonymity. This work was assuming that only the degrees of certain vertices are known by the adversary. Zhou [6] considered neighborhood attacks of a certain vertex and is extended by Tripathy [13], who assumed the adversaries having more information beyond 1-neighborhood knowledge. Later researches such like K-automorphism [10] and K-symmetry [12] proposed models with stronger privacy assurance on the whole structure property of graphs. Cheng [11] further discussed the K-isomorphism situation with consideration on link information. Recent years, attentions were paid on edge-labeled problems like Yuan [7] and Kapron [4].

In terms of the anonymization methods used, there are proposals focusing on edge addition/deletion [3] [4] [6] [11] [18], vertex/edge addition [12] [16], vertex generalization [14], edge label generalization [7] and class/cluster-based method [1] [13]. In terms of the attack model, there are models focusing on entity re-identification like [2] [3] [12], link re-identification like [8] [20], or many others considering both sides such as [7] [11]. Other approaches may be related to active attacks like Backstrom [5], in which attackers may change the data.

Our work is inspired mainly by [4] [7]. However, we focus on edge addition operation instead of label generation methods in [7] that is essentially a direct extension from unlabeled models. The core idea of [7] is to realize K-degree anonymity and generalize edge labels, which introducing a lot of noises (noticing that in the early grouping and addition steps, edge labels are not taken into consideration).

In some cases we may have restrictions not to modify the edge labels. Kapron [4] gives defined label sequences instead of degree sequences and proved that finding a result for the label sequence based anonymization problem with minimum cost is NP-hard when $K > 2$. What is more, this result can be used to prove the NP-hardness of some previously defined anonymization problems in unlabeled situations: K-neighborhood [6], i-hop anonymity [13] and k-symmetry [12].

It can be easily understood that an applicable algorithm for the label-graph anonymization problem can be applied to unlabeled graph situations with very slight modification. However in [4] only the algorithm for bipartite graph with $K = 2$ is given, which belongs to P. For this reason, in this paper, we provide greedy algorithms based on label-graphs. Formal definition of the problem is given in the next section.

3. Problem Definition

Now we formally definition the label-bag based graph anonymization problem. The definition of label-bag (LB) is as follows:

Definition 1. *label-bag (LB): A label-bag is a multi-set of labels. For a vertex v_i, LB_i is the set of all labels on edges which have one end point of v_i.*

With the definition of a label-bag, we can define the concept of label-bag based K-anonymity.

Definition 2. *label-bag based K-anonymity (LB K-anonymity): Given an simple edge-labeled graph G and integer K, for any vertex v in graph G, there exists at least K-1 vertices with the same label-bag.*

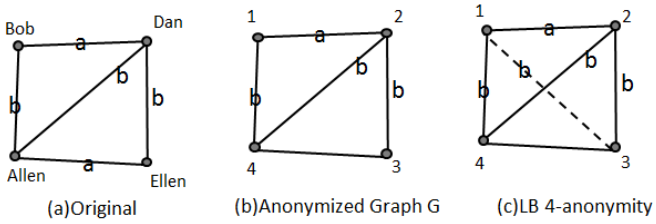

Figure 2 Example of LB K-anonymity

In the definition, a simple graph means an undirected graph without self-loop or multi-edges. For example, Figure 2(b) is an anonymized version of the original graph in Figure 2(a) by replacing identifies (names) with meaningless arbitrary unique numbers. The label-bag of vertex 2 and 3 in Figure 2(b) is {a,b} and {a,b,b} respectively (In the following of the paper, we may use strings like 'ab' and 'abb' to represent the label-bag just for simplicity).

Then we are able define the label-bag based anonymization problem. As have been mentioned, besides privacy requirements, another necessary task of privacy-preserving data publishing is to assure the usefulness of resulted anonymized data. There is a need to include the measure of information loss to the process of solving such kind of problems.

Definition 3. *LB K-anonymity problem: Given an simple edge-labeled graph* $G = (V, E)$ *and an integer* $K(K \geqq 2)$*, the LB K-anonymity problem is to construct a graph* $G' = (V, E \cup \Delta E)$ *such that* G' *satisfies LB K-anonymity condition with the minimum information loss.*

Definition 4. *LB K-anonymity problem 2: Given an simple edge-labeled graph* $G = (V, E)$ *and an integer* $K(K \geqq 2)$*, the LB K-anonymity problem 2 is to construct a graph* $G' = (V, E \cup \Delta E)$ *such that* G' *satisfies LB K-anonymity condition and makes* $|\Delta E|$ *minimal.*

The above definition is modified from that of [4]. Here we use the term "label-bag", since we do not consider the order among labels, whereas "label sequence" is used in [4]. It is easily to find that Definition 4 is just an extension of Definition 3 by specify the measurement as "make $|\Delta E|$ minimal". Making the number of added edges minimum is not a restricted condition in all LB K-anonymity problems. For example, the sum of degree difference of all vertices is also a possible criteria even with the same edge addition/deletion setting, although it is essentially equivalent to the number of added/deleted edges (see [3] for more details). In the rest paper, we assume Definition 4 as our main topic.

Figure 2(c) shows how to make a graph LB K-anonymized by adding edges to graph G in Figure 2(b). G' satisfies LB K-anonymity for any $K \leqq 4$. Note that only edge addition is allowed in this problem setting. In other words, we do not consider other operations, such as edge deletion or perturbation.

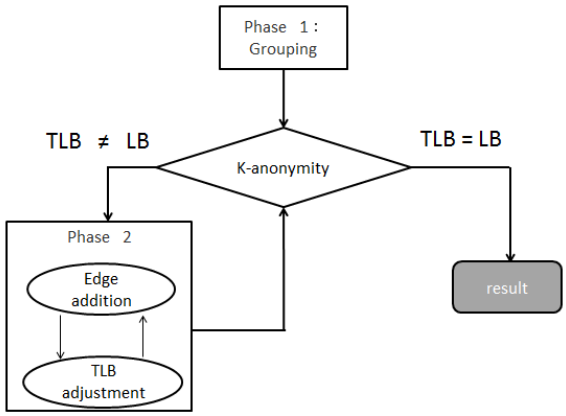

Figure 3 Algorithm Overview

4. LB K-anonymity

As already mentioned above, the LB K-anonymity problem is NP-hard. So we propose a two-phase heuristic algorithm.

Figure 3 shows the overview of our method. First we divide the input vertex sets into several groups and check if the LB K-anonymity condition is satisfied. If so, the result would be output. Otherwise, we move to the second edge-addition phase. The second phase is iterated until the LB K-anonymity condition is satisfied. Here a new term called TLB is used to record the graph state, which will be explaned later. Details of each phase will be introduced in sections 4.1 and 4.2 respectively. Now we give some definitions used in the algorithm introduction.

Definition 5. *target label-bag (TLB): For each group* g_m*,* TLB_m *is the ideal LB that all members in this group are supposed to reach.*

Definition 6. *residual label-bag (RLB): For every vertex* v_i*,* RLB_i *is the difference between* TLB_m *and* LB_i*, where* v_i *belongs to group* g_m*.*

Definitions 4 and 5 define two different kinds of label-bags, one for an anonymous group while the other for a vertex. The following equation shows how to compute the TLB_m for group m.

$$TLB_m = \sum_{i \in g_m} LB_i \qquad (1)$$

Here an operation " + " combines two label-bags and gives the union of two multi-sets. For example, 'aa'+'ab'='aab' and 'aab'+'bb'='aabb'. In other words,if a 3-size group contains vertices with label-bag 'aa','ab' and 'bb', we can tell

the TLB for this group is 'aabb'.

The equation below shows the way to calculate RLB. Also this is the relationship between TLB and RLB.

$$RLB_i = TLB_m - LB_i \tag{2}$$

Here the operation " $-$ " means to compute the label-bag that belongs to LB_i but not LB_j.

4.1 Grouping

The first phase is to divide the vertices into several non-overlapping subsets called anonymous groups. We utilize the TLB as a measurement to result in good grouping. Two different algorithms are used.

First is the feature based grouping algorihtm. This algorithm repreats a sequential scan to all vertices. Within each step, the vertex with least TLB increse is chosen to be added to group g until it reaches the pre-defined group size. Algorithm 1 shows the pseudo code of feature based grouping.

Algorithm 1 Feature based Grouping

Input: Graph $G = (V, E)$, strategy s
Output: Grouping {g}

```
{g} ← empty;
for m = 1 to s.numOfgroup(m) do
    TLB_m ← ∅
    for i = 1 to |g_m| do
        find v in V with least |TLB_m|
        insert v to group g_m
        TLB_m = ComputeTLB(TLB_m, LB_i)
    end for
end for
```

Since the edge addition procedure highly depends on the previous stage and the property of a grouping strategy would have great influence to the final cost, a more effective grouping algorithm is need. Inspired by clustering approaches like Campan [9], we give a clustering based method shown in Algorithm 2.

Algorithm 2 Clustering based Grouping

Input: Graph $G = (V, E)$, integer K
Output: Grouping {g}

```
{g} ← V;
while !(∀|g_m| ≧ K or no more merging) do
    Merging two closet clusters with condition C;
end while
{g} ← GroupAdjust({g})
```

Here condition C refers to: Any clusters with size larger than K are not further merged. Considering that traditional clustering methods have no restrictions on cluster sizes, this condition can effectively help to reduce the average cluster (group) size and avoid bias clustering result.

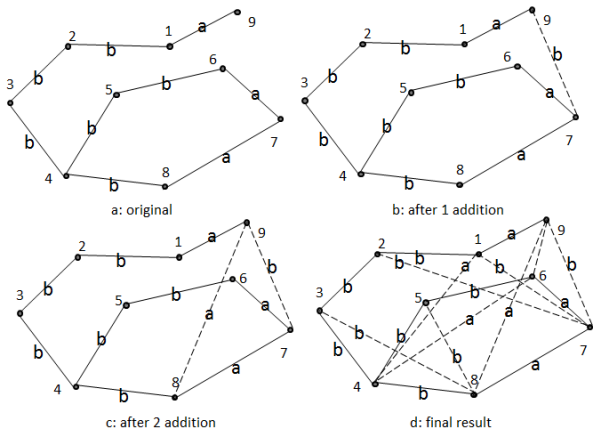

Figure 4 Algorithm Overview

A prototype-base hierarchical clustering method is used. The group set is initialized with vertex set V and we keep merging the closet clusters. Distance between two cluster is defined as:

$$Dist_1: \quad Dist(g_{m_i}, g_{g_j}) = |TLB_{m_i} - TLB_{m_j}| + |TLB_{m_j} - TLB_{m_i}| \tag{3}$$

A major drawback of this clustering grouping method is that group size is not taken into consideration when choosing closet clusters, while in fact it could be much easier to satisfy the same TLB for a group with smaller size than a larger one. So, we give two alternative distance metrics as follows:

$$Dist_2: \quad Dist(g_{m_i}, g_{m_j}) = Dist_1 * (|g_{m_i}| + |g_{m_j}|) \tag{4}$$

$$Dist_3: \quad Dist(g_{m_i}, g_{m_j}) = (TLB_{m_i} - TLB_{m_j}) * |g_{m_j}| + (TLB_{m_j} - TLB_{m_i}) * |g_{m_i}| \tag{5}$$

The function in last line would be responsible for the situation when only one cluster is under size K while it cannot be merged to any other one according to condition C. In that case, we can merge this cluster with a previous cluster which would results in minimum increase to the original value of $|TLB|$.

4.2 Edge Addition

After we have assigned vertices to groups, before we start to add edges to make all groups satisfying the anonymity condition, value of TLB for each group and RLB for each vertex need to be calculated for later use. Figure 5 is an example showing grouping result of graph in Figure 4(a), through feature based grouping algorithm.

As described in Figure 3, two steps are performed to achieve final LB K-anonymity state. Firstly we check all unconnected pairs of vertices with a common label in their RLB. We can find that both vertices 7 and 9 in Figure 5 have a label $'b'$. So, an edge can be added in between (Figure 4(b)). However, it is impossible to add an edge of label $'a'$ to vertices 4 and 8 since there already exists one and

Group	Vertex	neighbor	LB	TLB	RLB
1	9	1	a	a,b	b
	1	2,9	a,b		
	6	5,7	a,b		
2	2	1,3	b,b	b,b	
	3	2,4	b,b		
	5	4,6	b,b		
3	4	3,5,8	b,b,b	a,a,b,b,b	a, a
	7	6,8	a,a		b, b, b
	8	4,7	a,b		a, b,b

Figure 5 Algorithm Overview

we cannot find any other pairs. In this situation, procedure $TLBAdjustment()$ is conducted as shown in Figure 4(c), we increase the TLB of group 1 by $'a'$ and succeed to find vertex pair 8 and 9. We follow this route until all nodes satisfy the condition that $TLB = LB$.

Although it seems impossible to expect number of pairs being found in the next iteration of $GreedyEdgeAddition()$, same characteristics can help to find a better candidate group for $TLBAdjustment()$, like group 1 in the previous case. The following few factors are thought to be meaningful when selecting a candidate group:(1) average degree of the group;(2) average $|RLB|$ value;(3) connectivity to those vertices with large $|RLB|$ and (4) connectivity with other group members. The first factor is straightforward since vertex with larger degree can be more difficult to be wired with others. However most real-data of social network is so sparse that make the influence of this factor relatively slight. The last two factors mean the possibility vertices of the candidate group can be wired to other vertices. They are quite effective but is expensive to calculate. In all, a strategy combining factors (2)(3)(4) is used in our algorithm: sort all groups in descending order of their values of total $|RLB|$ values while excluding those with low connectivity.

By carefully choosing the candidate group, the algorithm can terminate in finite time (unless the situation of no answer and need to be solved by other means like adding noise nodes, which is beyond the range of this paper).

5. Improved Algorithm on Utility

Method introduced above focus on edge-labeled K-anonymity problem under the goal of less number of edges added. Nevertheless, in order to retain varied characteristics of the original graph, other utility measurement should be included. So in this chapter we introduce some other utility metrics an how their can be applied to our LB K-anonymity model.

In terms of the social network data, some of the graph properties are of interest. Most of them are basic graph properties as introduced in [14]: degree, path length, transitivity and so on. The other kind is related to the answers to extent queries like [7] and [20].

The utility metrics used in our paper are belonging to the first category as follows:

(1) Degree Distribution and Label Distribution. They are the measurement of degree and label property of graph data. EMD value (Earth Mover's Distance) [19] is used to measure the difference between two distributions. We adapt this concept to our research to calculate EMDD (EMD for degree) and EMDL (EMD for label). According to [19], having two signature (distributions) $P = \{(p_1, w_{p_1}), ..., (p_m, w_{p_m})\}$, $Q = \{(q_1, w_{q_1}), ..., (q_n, w_{q_n})\}$, Earth Mover's Distance is defined as

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{6}$$

where d_{ij} and f_{ij} represent the distance an flow between cluster p_i and p_j. This model is applied to network graph scenario by [17] in the following way. Let the attribute domain of numerical values be $\{v_1, v_2, ..., v_m\}$ and $r_i = p_i - q_i, (i = 1, 2, ..., m)$, the EMD value can be calculated as

$$EMD(P,Q) = \frac{1}{m-1}(|r_1|+|r_1+r_2|+...+|r_1+r_2+...+r_{m-1}|) \tag{7}$$

In our case, to calculate EMDD, let Distribution $P = \{(d_1, p_1), (d_2, p_2), .., (d_t, p_t)\}$, $Q = \{(d_1, q_1), (d_2, q_2), ..., (d_n, q_n)\}$, having their degree sequences sorting in ascending order. Here $p_i (i = 1, , 2...t)$ and $q_j (j = 1, 2, ..., n)$ means the weight of each degree d_i and d_j respectively. Let $m = maxt, n$, the common degree domain can be expressed as $\{d_1, d_2, ..., d_m\}$ while formula 5.2 can be used directly.

Similarly, in case of EMDL, let $\{l_1, l_2, ...l_m\}$ be the label domain makes formula 5.2 applicable.

(2) Shortest Path Distance (SPD). This is a widely used measurement in graph theory which means sum of weights of shortest path between two vertices. We utilize the average shortest path distance (ASPD) and calculate it through the famous Floyd-Warshall algorithm [22].

The Floyd-Warshall algorithms compares all pair of vertices' paths in a graph with complexity of $O(N^3)$. The idea of this algorithm can be expressed by the following recursive equation.

$$SPD(i,j,k) = min(SPD(i,j,k-1), SPD(i,k,k-1) + SPD(j,k,k-1)) \tag{8}$$

By computing this for all (v_i, v_j) pairs and for $k = 1, 2, ..., |V|$, we are able to get average shortest path distance by

$$ASPD = \frac{1}{|(v_i, v_j)|v_i \in V, v_j \in V|} \sum_{v_i \in V, v_j \in V} SPD(v_i, v_j) \tag{9}$$

(3) Clustering Coefficient (CC). This is a measure of how vertices in a graph tend to cluster together. Again we use the average local clustering coefficient (ACC) a s criterion.

Clustering coefficient can be roughly dived into two categories, global clustering coefficient and local clustering coefficient. Though they differ in definition, core idea is to measure the ratio of $k|edges|/|vertices|$. Suppose for a vertex v_i, its neighborhood N_i is defined as $\{v_j|e(v_i,v_j) \in E\}$, then $E_{N_i} = \{e(v_j,v_k)|v_j \in N_i, v_k \in N_i\}$ represents all the edges between vertex i's neighbors. The local clustering for vertex v_i is defined as follows.

$$CC_i = \frac{|E_{N_i}|}{|N_i|(|N_i|-1)} \tag{10}$$

We can directly derive the formula for average clustering coefficient as

$$ACC = \frac{1}{|V|}\sum_{i \in V} CC_i \tag{11}$$

Applying these utility measurements to anonymization algorithms is very simple, the only difference is in the edge addition step. Note that in the original method (without utility consideration) introduced in Section 4, we decide a match whenever two vertices share a common label in RLB and has no conflict of edge addition. While in new algorithm we tend to choose the candidate match with highest utility. To achieve this we compute the utility score for every pair of match before decision.

Algorithm 3 Edge Addition Algorithm with Utility

Input: Graph $G = (V, E)$, integer K, Grouping {g}, Utility Metric C

Output: LB K-anonymity Graph $G' = (V', E')$

```
1: {TLB_m} ← GetTLB({g_m})
2: Q ← {v|cRLB_i(l) > 0}
3: sort Q in descending order of cRLB_i(l)
4: for each u ∈ Q do
5:    minCost = INFINITY
6:    vv ← NULL
7:    for each v ∈ Q, v ≠ u do
8:       if !e(u,v) && (RLB_u ∪ RLB_v ≠ ∅) then
9:          cost = Compute(C,u,v)
10:         if cost < minCost then
11:            cost = minCost
12:            vv ← v
13:         end if
14:      end if
15:   end for
16:   if minCost ≠ INFINITY then
17:      add edge between u and vv
18:   end if
19: end for
```

Algorithm 6 is the description of *GreedyEdgeAddition*() function considering utility metrics. The utility metric C could be either metric be introduced above.

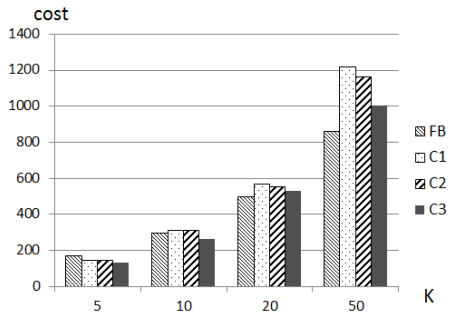

Figure 6 Result on variation of K

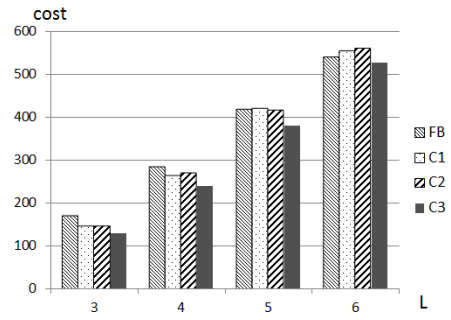

Figure 7 Result on variation of L

The new method considering utility metrics will bring extra execution expense, especially for case of ACC ($O(N^2)$ complexity) and ASPD ($O(N^3)$ complexity). We evaluate resulted utility scores of this method in the next section.

6. Experiments

We conduct a series of experiments to evaluate the efficiency and effectiveness of our algorithm. Experiments are conducted in environment as Table 1.

表 1 Experimental Environment

CPU	2.26 GHz Intel Core 2 Duo
Memory	4GB 1067MHz DDR3
Language/Compiler	c/gcc-4.2

Since social network is well-modeled by the small world graph, we use the Small World Graph generated by [23] as the synthetic data. Figure 6 and Figure 7 are the results on variant of anonymity level parameter K and number of labels L in 500 size synthetic dataset. Both of these parameters are positive correlated to the total cost. In case of parameter K (fixing L as 3), clustering based method, especially the one with distance metric 3 has good result in relative small K values. While feature based algorithm performs best in larger K situations. In terms of labels ($K = 5$), similarly the feature based algorithm and clustering based algorithm 3 outperform the others.

Table 2 shows the result of dataset-1, extracting from Speed Dating Data [24]. The graph is constructed by 551

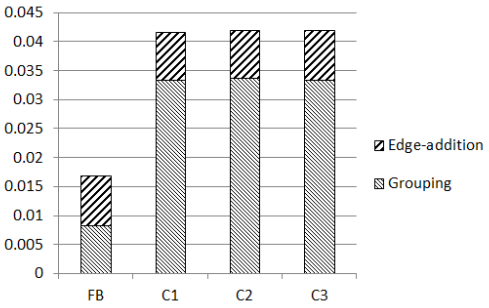

Figure 8 Execution Time

vertices, 8368 edges and 2 kinds of labels. The max degree of this graph is 22 and the average degree is around 15. The result is close to that of synthetic data and again reflects that clustering based algorithm with distance metric 3 and feature based grouping algorithm fit for different K values.

表 2 Real Dataset-1

K	Feature based	Clustering 1	Clustering 2	Clustering 3
5	169	112	117	131
10	398	309	312	291
20	664	567	526	540
50	1259	1325	2239	1820

In result of Table 3 we use a dataset extracted from the co-author network on condensed matter section of arXiv E-print Archive [25]. The graph consists of 16726 vertices and 47594 edges with the number of labels set as 3. Average and max degree are 5.09 and 107 respectively. It can be seen that feature based algorithms outperforms the others since K exceeds 10.

表 3 Real Dataset-2

K	Feature based	Clustering 1	Clustering 2	Clustering 3
5	1730	1871	1740	1628
10	3212	3779	3317	3063
20	5207	6864	6053	5706
50	10337	16502	12450	11629

Figure 8 shows the exectuion time of 2 phases for each algorithms applied to real dataset-1. Since the edge addition time is relatively small to that of grouping, to choose the feature based algorithm is more efficient.

Table 4 and 5 compare the results of methods considering utility with previous one on the 3 real datasets. Row with label "original" is the baseline that contains all utility scores of resulted LB K-anonymity graph by using method without considering utility metrics. Following are several lines each representing the results by applying different utility measurements.

表 4 Real Dataset-1 using utility metric

	Statics of Output Anonymized Graph				
Applied Utility	COST	EMDD	EMDL	ACC	ASPD
original	169	0.0198	0.0074	0.0455	3.2840
EMDD	169	0.0198	0.0074	0.0431	3.2954
EMDL	169	0.0198	0.0074	0.0431	3.2954
ACC	169	0.0198	0.0074	0.0051	2.5269
ASPD	169	0.0198	0.0074	0.0192	2.7787

In the result of real dataset-1, although EMD based methods do not have any improvement, which probably due to small label size of data, methods using ASPD and ACC result in good utility compared to the baseline.

For the sake of great computation cost, only EMDD and EMDL based metrics are evaluated for the second datasets. In this case both of these two utility metrics have no manifest influence to the results. This is because the low average degree of the whole graph makes the result of a few rewiring not obvious.

表 5 Real Dataset-2 using utility metric

	Statics of Output Anonymized Graph		
Applied Utility Metrics	COST	EMDD	EMDL
original	1730	0.0018	0.0060
EMDD	1733	0.0018	0.0060
EMDL	1733	0.0018	0.0060

7. Conclusion

In this paper, we discussed the K-anonymity problem in privacy-preserving data publishing. This is an extension from the unlabeled model and can be applied to many real-world situations. We provide a heuristic algorithm based on label-bag model and realize it in two different ways.

We evaluate them by some experiments on both synthetic and real data. Through the results, it is proved to be efficient and of good utility. Also, we investigate how choices in parameters would influent the cost. In consideration of the problem when there does not exist an answer, we give an algorithm based on noise vertex. An improved method is proposed considering four utility metrics and is proved to be of good utility through experiment results.

There is still a need to improve the edge addition algorithm to guarantee the realization of LB k-anonymity graph in arbitrary condition. To extend our model to utilize edge deletion and label generalization operations are other interesting topics.

Acknowledgement

This work has been supported in part by Grant-in-Aid for

文　献

[1] S. Bhagat, G. Cormode, B. Krishnamurthy, D. Srivastava. Class-based graph anonymization for social network data. VLDB Endowment 2009.

[2] L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty anonymization on graphs, SIGMOD 2008.

[3] K. Liu, E. Terzi. Towards identity anonymization on graphs. SIGMOD 2008.

[4] B. Kapron, G. Srivastava, S. Venkatesh. Social network anonymization via edge addition. ASONAM 2011.

[5] L. Backstrom, C. Dwork, J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. IW3C2 2007.

[6] B. Zhou, J. Pei. Preserving privacy in social networks against neighborhood attacks. ICDE 2008.

[7] M. Yuan, L. Chen, Philip S. Yu. Personalized privacy protection in social networks. VLDB Endowment 2010.

[8] C. Tai, Philip S. Yu, D. Yang, M. Chen. Privacy-preserving social network publication against friendship attacks. SIGKDD 2011.

[9] A. Campan, T. Marius Truta. A clustering approach for data and structural anonymity in social networks. PinKDD 2008.

[10] L. Zou, L. Chen, M. Tamer Özsu. K-automorphism: a general framework for privacy preserving network publication. VLDB Endowment 2009.

[11] J. Cheng, AW Fu, J. Liu. K-isomorphism: privacy preserving network publication against structural attacks. SIGMOD 2010.

[12] W. Wu, Y. Xiao, W. Wang, Z. He, Z. Wang. K-symmetry model for identity anonymization in social networks. EDBT 2010.

[13] B. Thompson, D. Yao. The union-split algorithm and cluster-based anonymization of social networks. ASIACCS 2011.

[14] M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis. Resisting structural re-identification in anonymized social networks. VLDB Endowment 2008.

[15] X. Liu, Y. Song, S. Bressan. Fast identity anonymization on graphs. DEXA 2012.

[16] S. Chester, B. Kapron, G. Ramesh, G. Srivastava, A. Thome. K-anonymization of social networks by vertex addition. ADBIS 2011.

[17] N. Li, T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. ICDE 2007.

[18] BK Tripathy, GK Panda. A new approach to manage security against neighborhood attacks in social networks. ASONAM 2011.

[19] Y. Rubner, C. Tomasi, L.J. Guibas. The earth mover's distance as a metric for image retrieval. Int. J Comput 2000.

[20] A. Korolova, R. Motwani, U.N. Shubha, Y. Xu. Link privacy in social networks. ICDE 2008.

[21] http://math.standford.edu/muellner.

[22] Cormen, H. Thomas, Leiseron, E. Charles, Rivest, L. Ronald Introduction to Algorithms 1990.

[23] Networkx 1.6. http://networkx.lanl.gov/index.html.

[24] Speed Dating Data. http://flowingdata.com

[25] M. Newman. Scientific collaboration networks 2. Shortest paths, weighted networks, and centrality 2001.