PROCEEDINGS OF SCIENCE

(k.l)-Anonymity for Social Networks based on k-Neighborhood Anonymity

Qi Hu[1]
Department of Computer Science and Technology Guizhou University
Guiyang. China
E-mail:huqi19910808@163.com

Chaohui Jiang
Department of Computer Science and Technology Guizhou University
Guiyang. China
E-mail:jiangchaohui@126.com

Yuanyuan Wang
Department of Computer Science and Technology Guizhou University
Guiyang. China
E-mail:circle_1990@sina.com

Preserving the privacy of the data as released in social network has now become a hot topic. Many corresponding algorithms are proposed. An important factor to measure the privacy preserving algorithm is the utility of the released data. The k-anonymous algorithm is the most widely used in many privacy protection algorithms. K-anonymity algorithm ensures the security of data to some extent; but in the process of anonymity. due to the lack of data diversity. there is also a higher risk of privacy disclosure. In the current study. a simple undirected graph of a class of nodes that have attribute values is plotted. For the attack by the knowledge at the background of the neighborhood information of the nodes. we've developed. on the basis of k-anonymous and l-diversity. a (k.l)-anonymous model to preserve privacy. In this model. the nodes are neighborly anonymized and their attribute values are at meanwhile generalized. As shown in the experimental results. this model is safer than the simple k-anonymous model and more accurate in response to the queries of the anonymous data released in aggregate networks.

ISCC2015
18-19. December. 2015
Guangzhou. China

[1]Speaker
[2]This study is supported by Large data aggregation mechanism and analysis and Mechanism Research (JZ [2014]2001)

PoS(ISCC2015)012

1.Introduction

The rapid development of the Internet and information technology has resulted in the popularity of social networks. such as Facebook and Twitter that are widely used in foreign countries. and MicroBlog. Renren and Tencent. etc.. which are popular in China. On these social platforms everyday. a great many of users are interacting with their friends or sharing their feelings. stories and experience. The consequence is the generation of a great deal of unstructured data. which need to be released for the purpose of scientific research and data sharing; however. the users' privacy information may be contained in such data. An arbitrary release may lead to privacy disclosure; therefore. preserving the privacy of the data as released in social networks has now become a hot topic.

For the case where the background knowledge of the attacker is the neighborhood information of the target node. B. Zhou and J. Pei had employed the k-anonymity and divided the neighborhoods of nodes into components and encoding. which makes it easy to compare the isomorphism between neighborhoods and keeps the accuracy of the aggregate network queries[1]. The k-anonymity model is safe to the simple undirected graph. in which the nodes have no attribute values. but is risky when the nodes in the undirected graph have attribute values. which may lead to privacy disclosure. Machanavajjhala et al. also pointed out that the *k*-anonymity may lead to privacy disclosure due to lack of diverse sensitive attributes[2]. To overcome this problem. they builted a *l*-diversity model. which was proved to be more effective to preserve privacy than the *k*-anonymity model[2]. A *(k. l)*-anonymous model based on *k*-isomorphism is proposed [3-6]. The model anonymity the social network data into an anonymous social network graph that contain *k* isomorphic subgraph. and divides all the nodes which have the same location in the subgraph into an equivalent class.

This paper focuses on studying the simple undirected graph. in which the nodes have attributes values but the edges have no attributes values. We presents a *(k.l)*-model based on the k-neighborhood anonymity. First of all. we use k- neighborhood anonymous algorithm for social network data. then. the sensitive attributes of nodes are generalized with respect to *l*-diversity. The empirical study-indicates that *(k.l)*-model not only keeps the accuracy of aggregation network queries of the released data. but also combines the advantages of k-anonymity algorithm with *l*-diversity algorithm and makes the k-anonymity more effective to preserve privacy.

2.Social Network Graph And Related Concepts

In practices. the social networks are very complex and usually demonstrated by graphs. where. the nodes represent entities. such as individuals. firms or organizations. and the lines connecting nodes represent the relation between the entities. such as friendship. lover relation or commercial relation. Generally. nodes and edges in a practical social network have attribute values. This paper focuses on studying the simple undirected graph. in which the nodes have attributes values but the edges have no attribute value.

Definition 1: in the social network graph *G=(V.E.L)*. *V* is a set of nodes. representing entities in the graph; *E* is a set of edges. representing relations between the entities; *L* is a set of attributes; and function *L*: $V \to L$ represents the attribute value of each node.

A social network contains lots of information. The information of the target node known by the attacker is defined as background knowledge. In practical. it is difficult to predict the attacker's background knowledge. and there is no privacy preservation model that can defend all attacks on the background knowledge. In our study. the attacker takes the 1-neighborhood information of the target node as the background knowledge. Unless otherwise stated. all neighborhoods in this paper are the 1-neighborhood.

Definition 2: 1-neighborhood attack. For a node V in the social network graph $G=(V.E.L)$. the attack on the target node V by the attacker. who knows the number of the nodes adjacent to the node V and the relations between them and takes this as the background knowledge. is defined as 1-neighborhood attack.

3.(k.l)-Anonymity Model

3.1 K-neighborhood Anonymity

The k-neighborhood anonymity model in our study is improved from the classical k-anonymity to defend privacy against the attack that takes the neighborhood information as the background knowledge. We hope the probability that the position of the target node being identified by the attacker is no lager than a threshold. Given a positive integer k. for every node . there are at least $k-1$ nodes' neighborhoods that are isomorphic with u neighborhood after the graph G being anonymized. The anonymization of the graph is exactly the isomorphism of the node neighborhoods.

Definition 3: let $G=(V.E.L)$ be a social network graph and G' an anonymization of G. If G' is k-anonymous. then any node in G cannot by re-identified in G' with confidence lager than $1/k$.

3.1.1Neighborhood Anonymity and Coding

It is a NP-hard problem to determine whether two graphs are isomorphic. In this paper here. we used a coding technique. as stated to divide the neighborhood of the node into components and encoding them so as to use their codes to determine whether two node neighborhoods are isomorphic or not.

Definition 4: in a social network G . a subgraph C of G is a neighborhood component of $u \in V(G)$. then C is a maximal connected subgraph in $Neighbor_G(u)$.

We used the neighborhood component coding[1]. and made DFS (depth-first search) to the neighborhood component C in $Neighbor_G(u)$. The minimal DFS-tree is taken as the code of the neighborhood component. denoted by $DFS(C)$. In a social network $G=(V.E.L)$. for every node $u \in V(G)$. the neighborhood component code of $Neighbor_G(u)$ is a vector $NCC(u)=(DFS(C_1).....DFS(C_m))$. where $C_1.....C_m$ are the neighborhood components of $Neighbor_G(u)$. i.e.. $NeighborG(u)=\cup_{i=1}^{m} C_i$ By using the neighborhood component coding technique. we can easily identify two isomorphic neighborhoods.

Definition 5: for two nodes $u.v \in V(G)$. where G is a social network. $Neighbor_G(u)$ and $Neighbor_G(v)$ are isomorphic if and only if $NCC(u)=NCC(v)$.

Anonymization Cost

Information loss during neighborhood anonymization can be measured by the number of edges added and the number of nodes that are not in the neighborhood of the target node. Consider two nodes $u_1.u_2 \in V(G)$.in a social network *G*. Suppose $Neighbor_G(u_1)$ *and* $Neighbor_G(u_2)$ be anonymized into $Neighbor_G(u'_1)$ and $Neighbor_G(u'_2)$ respectively. Let $H=Neighbor_G(u_1) \cup Neighbor_G(u_2)$ and $H'=Neighbor_G(u'_1) \cup Neighbor_G(u'_2)$.

The anonymization cost of the two nodes is:

$$Cost(u.v)=\alpha\cdot|\{(v_1.v_2)|(v_1.v_2) \notin E(H).(v'_1.v'_2) \in E(H')\}| + \beta\cdot(|V(H')|-|V(H)|) \quad (3.1)$$

where α.β. are specified according to the practical requirements.

3.1.2k-neighborhood Anonymity Algorithm

We used a greedy algorithm to anonymize the social network graph.

Input: A social network graph *G=(V.E.L)* and the parameter *k*;

Output: the anonymized graph *G'* of *G*;

Initialization:*G'=G*. mark all nodes as "unanonymized"

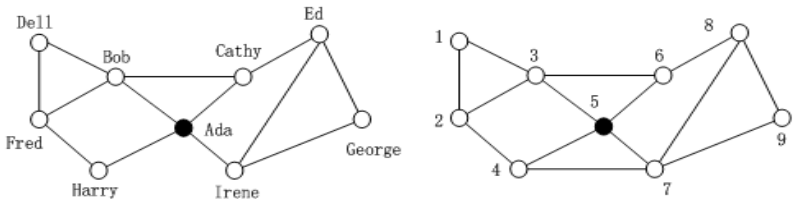
Steps:

- Put $V_i \in V(G)$ into *VertexList* in a descending order. and code the neighborhood components of all nodes;
- Put the first node in *VertexList* into *SeedVetex*. and remove it from *VertexList*.
- For each node $v_i \in$ *VertexList* . use the neighborhood component coding to determine whether v_i and *SeedVetex* are isomorphic; if isomorphic. then the anonymization cost is 0; otherwise. using equation (3.1) to calculate the anonymization cost of v_i and *SeedVetex*.
- If *VertexList.size( )≥2k-1*. then let *CandidateSet* contain the top *k-1* nodes with the least cost; otherwise. let *CandidateSet* contain all the unanonymized nodes.
- Anonymize all the nodes in *CandidateSet* and mark them as "anonymized".
- Repeat steps 2-5 till all nodes are marked as "anonymized".

The size of a node is determined by the number of nodes and edges contained in the neighborhood. For $u.v \in V(G)$. if $|V(Neighbor_G(u))|<|V(Neighbor_G(v))|$. *or* $|V(Neighbor_G(u))|=|V(Neighbor_G(v))|$. and $|E(Neighbor_G(u))|<|E(Neighbor_G(v))|$. then node *v* is larger than node *u*. If their neighborhoods have the same numbers of nodes and edges. they can be ordered arbitrarily.

3.2 L-diversity

k-neighborhood anonymity is safe against the privacy attack on the simple undirected graph in which both the nodes and edges have no attribute values; however. it is risky when the nodes and edges in the simple undirected graph have attribute values. In a social network graph shown in Fig. 1(a). for example. Ada is the target node of the attacker. who knows the neighborhood information of Ada; so the position of Ada can be identified uniquely by the attacker. After 2-neighborhood anonymity. the anonymized graph of the social network graph can be obtained. as shown in Fig. 1 (b). in which. all neighborhoods are isomorphic with the neighborhood of one node at least. which makes it impossible for the attacker to identify the target node Ada with a confidence higher than 1/2.



(a)Social Network Graph G (b)Anonymized Graph

Figure 1: Anonymous Release of A Social Network Graph

In the above case. however. *k*-neighborhood anonymity is safe when the social network nodes have no attribute values. For nodes in graph G. their attribute values are listed in Table 1. in which "Salary" is a sensitive attribute. In Fig. 1(b). the neighborhoods of node 3. 5 and 7 are isomorphic. but their salaries are the same: 3400. Because of the lack of diversity. an attacker is easy to identify that the salary of Ada is also 3400.

For this reason. we introduce the *l*-diversity model based on the *k*-neighborhood anonymity to make the k-neighborhood anonymity model safer when defending the attack on the target node. of which the neighborhood information is taken as the background knowledge.

ID	Age	Salary
1	21	3300
2	20	3500
3	24	3400
4	20	3300
5	22	3000
6	20	3400
7	21	3400
8	25	3500
9	24	4000

Table 1: Label Information

Consider a social network graph *G=(V.E.L)* and its anonymous graph *G=(V'.E'.L')*. Let the nodes in *G'* that have isomorphic neighborhoods be contained in an equivalence group. denoted as *VCS*; then we can obtain the sensitive attribute group of the nodes in *VCS*. denoted as *SA(VCS)*. As shown in Fig. 1(b). VCS_1*={3.5.7}*. VCS_2*={2.4.6.8}* and VCS_3*={1.9}* are the equivalence groups of the 2-neighborhood anonymity graph. and $SA(VCS_1)$*={3400. 3400. 3400}*. $SA(VCS_2)$*={3500. 3300.3400.3500}* and $SA(VCS_3)$ *={3500.4000}* are the corresponding sensitive attribute groups.

Definition 6: for a social network *G=(V.E.L)* and its anonymous graph *G=(V'.E'.L')*. if all the sensitive attribute groups in *G'* meet $|SA(VCS)| \geq l$. where l is a positive integer ($l>1$). then *G'* meets the *l*-diversity.

We used the dividing equivalent class method[3] to generalize the sensitive attributes and classify SA into two types: categorical sensitive attributes and numerical sensitive attributes.

For categorical sensitive attributes. *C* is the categorical inheritance tree of *SA*. The generalization of node V_i is to use the parent node of V_i^C in the categorical inheritance tree to replace V_i^C. $|SA(VCS)|$ is the number of different categorical values in the sensitive attribute group.

For the numerical sensitive attributes and for all equivalence groups *SA(VCS)*. divide the interval $[min\{s_1.s_2.....s_n\}.\ max\{s_1.s_2.....s_n\}]$ into *m* subintervals ($l \le m \le n$). $|SA(VCS)|$ is the number of sensitive attribute values that are in different subintervals. The generalization is to replace the value of s_i by the subinterval of s_i. During the generalization. move the subinterval of s_i forward or backward by 1 subinterval.

3.2.1Anonymization Cost

For a social network *G=(V.E.L)* and its anonymous graph *G'=(V'.E'.L')*. there are *t VCS* equivalence groups in *G'*. and *gen(SA(VCS))* represents the generalized sensitive label group of *SA(VCS)*. then the generalization information loss *(GIL)* of the social network graph *G* is

$$GIL(G) = \sum_{i=1}^{t} \frac{|gen(SA(VCS_i))| - |SA(VCS_i)|}{|SA(V)|} \tag{3.2}$$

where. |SA(V)| is the number of different sensitive attribute values in graph G.

3.3 (k.l)-anonymity Algorithm

According to the analysis on the k-neighborhood anonymity as well as on the l-diversity model. as described in Section 2.1 and 2.2. we can obtain the *(k.l)*-anonymity algorithm. Based on the *(k.l)*-neighborhood anonymity. the algorithm is to generalize nodes' sensitive attributes by using *l*-diversity. as demonstrated as follows:

Input: A social network graph *G=(V.E.L)* and the parameter *k* and *l(k≥l≥2)*;

Output: the anonymized graph *G'* of *G*;

Initialization: *G'=G*. mark all nodes as "unanonymized".

Steps:

- Put $v_i \in V(G)$ into *VertexList* in a descending order. and code the neighborhood components of all nodes;
- Select the first node from *VertexList* into *SeedVertex*. and remove it from *VertexList*.
- For each node $v_i \in VertexList$. using the neighborhood component coding to determine whether v_i and *SeedVetex* are isomorphic; if isomorphic. then the anonymization cost is 0. otherwise. using equation (3.1) to calculate the anonymization cost of v_i and *SeedVetex*.
- If *VertexList.size( )≥2k-1*. then let *CandidateSet* contain the top *k-1* nodes with the least cost; otherwise. let *CandidateSet* contain all the unanonymized nodes.
- Extract all nodes and their sensitive attribute values from *SeedVertex* and *CandidateSet* into *VCS* and *SA(VCS)*. respectively.
- Anonymize all the nodes in *CandidateSet* and mark them as "anonymized".
- If *|SA(VCS)|≥l*. then do next step; otherwise. the determine the type of *SA (VCS)*. Using the method described in section 3.2 to make generalization from the first node in *VCS*. After each generalization. judge whether *|SA(VCS)|≥l*; if so. stop the generalization.
- Repeat Steps 2-5 till all nodes are marked as "anonymized".

4.Experimental Results And Analysis

4.1Experimental Consonditi

In this paper. we used the Pajek software to generate a medical network G that had 300 nodes and the average node degree was 6. in which. the disease information was the sensitive information. The experiment is conducted in the Windows 7 Ultimate operating system. CPU FX-6100 3.30G-Hz. 4.00GB memory. programming language C++. and the operating platform of Microsoft visual studio 2010.

4.2 Information loss and Execution Efficiency

The information loss of this algorithm is composed of two parts. one is the loss of *k*-neighborhood anonymous and the other one is loss of *l*-diversity. that is. the total amount of information loss is Equation (3.1) plus Equation (3.2).

The information loss and execution efficiency of the algorithm are calculated at different *k* values. when compared to the *k*-isomorphism method[3]. As shown in Fig. 2. the demand for relative privacy preservation is raised with increase of the *k* value. The information loss and execution efficiency are also increasing. It can be seen from Fig. 2 that. when the *k* value is larger than a threshold. the information loss by our algorithm is higher than the *k*-isomorphism method[3] because our algorithm is safer than the *k*-isomorphic algorithm and has more operations on graph edges. As shown in Fig. 3. the execution time of our algorithm is also longer than the *k*-isomorphism method[3] because our algorithm requires to isomorphically compare each node when making the *k*-neighborhood anonymity.

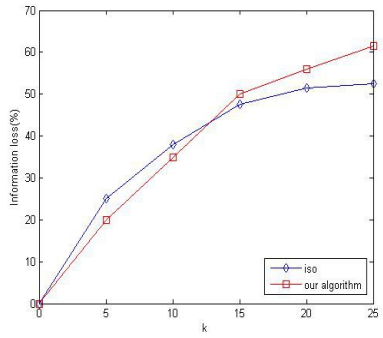


Figure 2: Information Loss

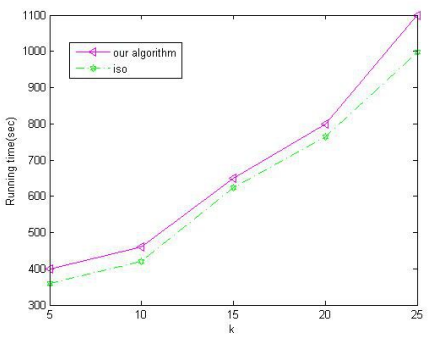


Figure 3: Execution Efficiency

4.3 Aggregate Network Query

Error rate of aggregate network query is tested by changing the value of *k*. Select 50 node pairs and calculate their average shortest distance. The error rate of the average shortest distance is calculated by the followed method:error rate: $r=(d-d')/d$. where. *d* and *d'* are the average distances in the original network and in the anonymized network. respectively. The experimental results are shown in Fig. 3. The error rate increases with the k value. but the error rate is small even when k is up to 25.

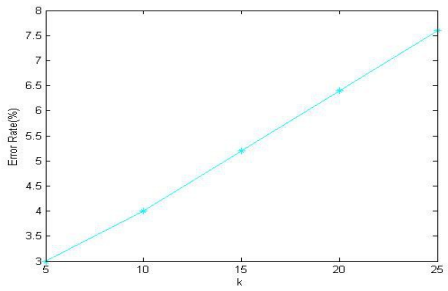


Figure 3: Error Rate of Aggregate Network Query

5. Conclusion

This paper conducted a study on (k.l)-anonymity for the simple undirected graph. in which the nodes have sensitive attribute values and use the k-neighborhood anonymity to defend the attack from the background knowledge of neighborhood information. On this basis. the sensitive attributes of nodes are generalized with respect to l-diversity. In an actual social network graph. the nodes often have sensitive or insensitive attribute values. Our algorithm not only improves the safety of the k-anonymity but also is valuable in practical applications; but in the actual social network graph. the background knowledge of the attacker is varied. In addition to attacks based on node neighborhood information. there also is redefinition attack from the background knowledge of the node degree. In the next research. we can establish the (k.l)-anonymity model for attacking from the background knowledge of the node degree. In the practical aspects of anonymous data. in addition to considering the aggregation network query of the graph. the general properties of the graph should be considered. such as node degree distribution. average shortest path and clustering coefficient. etc. In the future. we can improve (k. l) - anonymous algorithm to further improve the utility of anonymous data.

References

[1] B. Zhou and J. Pei.. "*Preserving privacy in social networks against neighborhood attacks*" In Proc. of the 24th IEEE International Conference on Data Engineering. USA:IEEE. pp. 506-515. 2008

[2] Machanavajjhala A. Kifer D. Gehrke J. et al.. "*l-diversity: Privacy beyond k-anonymity*". In Proc. of the 22th IEEE International Conference on Data Engineering. Washington. USA:IEEE. vol. 24. 2006.

[3] HW Wu. RW Zhang. HT Wang. ZB Sun. "*(k.l)-Anonymity for Social Networks Publication Against Composite Attacks*". Journal of Harbin University of Science and Technology. vol. 18. n. 3. pp. 47-53. 2013.(In Chinese)

[4] B. Zhou and J. Pei. "*The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks*". Knowledge and Information System. vol. 28. n. 1. pp.47-77. 2011.

[5] J Cheng. WC Fu. J Liu. "*K-isomorphism:Privacy Preserving Network Publication Against Structural Attacks*". In Proc. of the 2010 International Conference on Management of Data. ACM. pp. 459-470. 2010.

[6] XY Liu. B Wang. XC Yang. "*Survey on Privacy Preserving Techniques for Publishing Social Network Data*" Journal of Software. vol. 25. n. 3. pp. 576-590. 2014. (In Chinese)