Technique On K-Dynamic Graph Anonimyzation Algorithm

Jingyu Dai[†],

[†]Junior, [§]ZJU, [‡]Information Security, College of Computer science and technology

Abstract

In the information age, the transmission of network information has brought great convenience to people, and it has also brought new security risks.The rapid development and popularization of social networks have attracted researchers in various research fields to engage in the analysis and research of social networks, and put forward and solved many problems with important applications and research values in the process of social network analysis. Among them, the most troublesome thing is how to protect personal information transmitted on the Internet. In August 2019, some experts published papers on how hackers learned about personal privacy information by anonymizing attacks on social relationships, which caused a sensation in society. In such a large environment, the role of graph anonymization becomes more and more important. Compared with text encryption, how to encrypt the relationship diagram while ensuring that its content can still be interpreted correctly is technically more challenging. On this basis, this paper proposes a link complexity guarantee anonymization algorithm (referred to as CCA algorithm). By adding a new child node after generating the connected subgraph, it is possible to combat anonymization while ensuring that the picture information can still be correctly interpreted. The reliability of the algorithm was verified based on experimental data and analysis.

Index Terms

Graph Anonymization, Dynamic algorithm, greedy algorithm, sub-graph

I. Introduction

Since the beginning of the 21st century, the rapid development of the Internet has driven the development of many industries. With the development of online social software, more personal information, such as photos, diaries, etc., are displayed on the personal social circle through the Internet. The Internet generates a large amount of data every day, providing a reliable data

sample and research platform for scientific research. In the study of social networks, we abstract the social relationships between users into the form of diagrams, using points to represent users, and using links to represent relationships between users (usually undirected graphs). Research on social relations has been transformed into a study of the relationship between graph connectivity and other features. By studying these diagrams, we can push users to their favorite content, reduce the uselessness of users in finding information, and let people experience the convenience brought by the information age.[1] Due to the need to maintain security, scientific research, etc., a large amount of social data will be published on special websites for scholars to study. When a publisher publishes data, the data is likely to be stolen by illegal elements that are used to mine the user's private information and then harass users or other privacy attacks. Hackers use the customer's private information to conduct illegal information transactions, so that these users are subject to junk advertising. What's more, extorting them directly by contacting customers. These behaviors not only violate basic legal and ethical norms, but also interfere with normal diagram research. Their research results may always be converted into time bombs by lawless elements. In this context, the study of graph anonymous algorithms is very necessary. The primary way to handle this method is to anonymize the data that needs to be published. Simply encrypting the user's personal information (such as username, birthday, etc.) in the diagram is not very effective. Because intruders can infer the personal information of other users through the unencrypted information of other nodes. With the continuous upgrading of image anonymization methods, the way image anonymization is also limited to a certain extent.

A. Research Status

The current research on graph anonymity has the following methods: Datafly algorithm:This algorithm needs to sequentially select the number of different attributes in each attribute, select the most attribute as the generalization attribute, and generalize according to the attribute. If the generalization result satisfies

K-anonymity, the algorithm ends, otherwise the previous steps are repeated until the result meets the requirements. The generalized anonymous method is based on the principle of giving a unified identification information to multiple points or edges in a social network. This information indicates that it should be a union of multiple points or sides of information.[2]

The Backstrom anonymity method is based on the identification of an anonymized sub-graph

to identify an anonymous social graph. Narayanan's anonymous method, the principle is: to re-identify the anonymous point through the sequence of the degree of multi-layer neighbors.

KACA Algorithm: The clustering is carried out through the attribute structure hierarchy, and the concept of weighted generalized hierarchical distance including the weighting method of unified weight and highest weight is proposed, and the distance between the tuples is defined to describe the degree of distortion of generalization.

B. Problems

1. The current algorithm does not achieve a good balance between usability and execution efficiency. Higher execution efficiency requires sacrificing data availability; algorithms that guarantee data integrity are not able to handle large-scale data.

2. Anonymous algorithm can not guarantee the security of data while ensuring accurate data division, which increases the risk of privacy leakage.[3]

C. Research Aim

Using the knowledge learned, design an algorithm that can guarantee the anonymity of the relationship graph. The relationship graph anonymized by this algorithm can effectively protect user information stored in and between nodes without excessively destroying the original information. At the same time, the anonymous graph processed by this algorithm should also have a certain ability to prevent de-anonymization program attacks.

II. My algorithm theory

This paper will implement the KDA algorithm through Java or Python. The principle is to find a specific x points ($x \leq K$) on the basis of the smallest sub-graph by first reducing the relation graph containing K points into the smallest sub-graph. Add new edges and x special points from the x points, and perform second "encryption" on the anonymized graphs based on the same connection relationship and without affecting the original relationship graph, so as to achieve a more secure image. Anonymized.

III. PRELIMINARY MODEL

The model of the KDA algorithm should consist of the following parts: input, division, generating the smallest anonymous graph, combining the sub-graphs, and generating the final anonymized relationship graph

A. *Input*

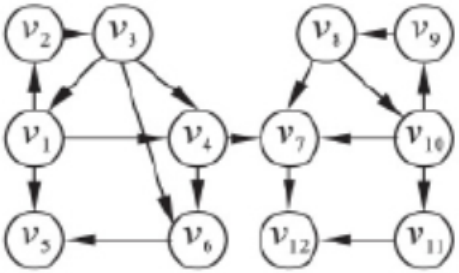
At the beginning, the algorithm needs to correctly read the graph node information from the .cvs file, restore the association between the nodes, and perform preprocessing such as sorting to facilitate the subsequent anonymization process.

B. *division*

We need to formulate a reasonable standard to segment all nodes of the relationship graph. The complete relationship graph is divided into multiple sub-graphs, and these sub-graphs are anonymized separately.In order to facilitate the final anonymization without affecting the data restoration, we need to mark those nodes with fewer feature values but cannot be deleted directly, so that we can reduce the useless work in the de-anonymization process. These attributes should be few in the original relationship graph, and these features must exist on multiple nodes at the same time to avoid targeting these points in the case of improper cracking.

C. *generating*

Initializing image anonymization with a minimal sub-graph is an NP-hard problem. This can be proved by NP-complete reduction analysis. Suppose we have a relation graph G(V, E) and a positive integer d, then let G'(V', E') be a sub-graph of G and satisfy the condition N(V) = N(V') and N(E) $\leq$ N(E'). For human factors u and v, if there is a path from point u to point v in G, and there is also a path in G', then G' is a connectable sub-graph of G. Let d denote the number of paths that can be connected. When N(E') reaches the minimum value and there are still d connectable paths in G', then G' is the minimum connectable sub-graph of G.



d^{in}	d^{out}	vertices
1	3	v_1, v_2, v_3
1	1	v_2, v_9, v_{11}
2	0	v_5, v_{12}
2	2	v_4
2	1	v_6
3	1	v_7
1	2	v_8

connective graph

D. generating

In the final stage, we need to combine the previously generated multiple anonymous graphs and call the algorithm again to check whether it meets the standard anonymization requirements. In this way, we can get the anonymized relationship graph generated by the KDA algorithm.

IV. Difficulties

A. Time complexity

In this algorithm, if we need to record the characteristics of each individual, then we must traverse all the features and build a statistical table. Such an approach is uneconomical in terms of time complexity, and when such an algorithm processes large amounts of data, it leads to reduced efficiency. We need a way to ensure that most of the features are recorded.

B. Graph restore

When anonymizing the graph, because some nodes are deleted, these nodes may carry more important information. Therefore, in the process of anonymizing the relationship graph, it is also necessary to consider the influence of these feature points on the accuracy of the graphic restoration. In this case, we cannot blindly pursue the effect of anonymization and ignore the ultimate purpose of graph anonymization, while protecting privacy and ensuring that the restored data is still truly usable. The original data and the relationship between nodes cannot be changed too much for anonymization. If the authenticity of the restored data is too low, the anonymization algorithm will lose its meaning.

V. Solutions

A. Choice of algorithm

The degree greedy anonymization algorithm can deal with the problems of time complexity. In this algorithm, We can traverse the data to calculate the degree of each node, and then sort these nodes according to the degree, and then decide whether to anonymously process these points during the anonymization process according to the degree of the node and the degree of connection repetition.

Compare with other algorithms, greedy algorithm has the fastest average time about O(nlogn), If it can be optimized with the stack and queue. However, when we solve the problem with the time complexity, we are shocked that greedy algorithm makes trouble in data restore. Because greedy algorithm always making the best choice when solving a problem. That is to say, without considering the global optimality, what he makes is a local optimal solution in a sense. This might be a good choice when we deal with normal sub-graph problem, but when it comes to anonymization, it means that it will abandon some nodes even if they contain unique information. When I start my test, I try some groups of data and find that greedy algorithm is not reliable enough for graph anonymization. With the grow of number of the data, the information distortion rate of the anonymous graph generated by the greedy algorithm is also gradually increasing, so that we ultimately cannot accurately restore the original data from the anonymous graph.

```
Begin
    1 original: s={1},v={2...n},dist[i]=c[i][j];
    2. data process
    2.1 u=min{deg[i][j]|i∈v};
    2.2 s=sU{v},v=v-{u};
    2.3 judge node i in collection v;
        if(dist[u]+c[u][i]<deg[i][j]) deg[i]=deg[u]+c[u][j];
    3. record deg;
    4. sort();
End
```

Fig. 1. greedy algorithm brief code

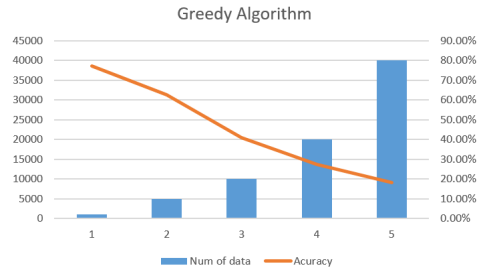


Fig. 2. greedy algorithm accuracy

In order to solve the influence of greedy algorithm on information accuracy, using dynamic programming algorithm is a good solution. First build an array to store node connection information. In the dynamic planning stage, by increasing the number of degrees and the changes in connected nodes, it is determined whether the anonymization process has excessively destroyed

the information saved in the original image. If the information is damaged, connection reconstruction is performed to guide the construction of node connections that meet the requirements.

B. Describe and prove

To reach the goal, The brief implementation steps of the dynamic programming algorithm are as follows:

(1) Read node data from .cvs file, including name, contact object and other information. First, we should prepare a class and define its elements as follow:

```
String name;
int degree;
ArrayList<String> edge;

public Node() {
    this.name= "";
    this.degree= 0;
    this.edge= new ArrayList<String>();
}
```

Some attribution of node

Then, the input class needs to pre-process the relational graph data that is read in. Including deleting irrelevant symbols, counting the number of nodes, and the degree of each node. Finally, the node numbers are sorted according to the degree and saved in the newly created array.

```
for each message
{
        delete irrelevant symbols;
        find valid node;
        create new number for nodes;
}
for each node:
        counting degrees and record;
sort by degree
```

structure of pre-process

(2) Suppose one natural number K = x and another natural number s, where s will change continuously as the dynamic programming progresses. Traverse the read node information, and divide the group during dynamic programming according to the degree of the node. The specific method is: according to whether the degree of the node v satisfies the following relationship: degree (v)–s $\leq$2 * k. Group based on whether the node meets this condition.

```
initialize;

for each node
{
        if (deg-s<2*k)
                record in group_1;
        else
        {
                record in group_2;
                record relationship;
                note association;
        }
        renew s;
}
```

structure of division part

(3) Based on (2), the segmented sub-graphs are anonymized according to the degree and attributes of the nodes in each group. At the same time, a judgment mechanism is introduced: when a node deletes too many edges due to anonymization, which affects the authenticity of the information, it is necessary to re-plan the edges connected to the node until the average standard is reached.

```
for each node
{
        check deg(node);
        if get_deg>0 renew deg(node);

        for each association
        {
                check info(node);
                if deg(node)<N refactoring;
                renew info(node);
                resort();
        }
}

combind sub-graph;
recheck relationship;
output;
```

structure of anonymize part

(4) Finally, reconnect the anonymized sub-graphs to a complete anonymization relationship graph.

```
Min-Degree(G, s, t)
{
    n ← number of nodes in G
    for all v except t, initialize M[v] ← +∞
    M[t] ← 0
    for i←1 to n-1
        for all e=(v,w)∈E(G)
            M [v] ← min(M [v], c(v, w) + M [w])
    return M[s]
}
```

Fig. 3. Dynamic algorithm brief code

This algorithm may cost more time on graph anonymization, but its result is more reliable and keep more unique information.

VI. Experimental result

A. Software and hardware

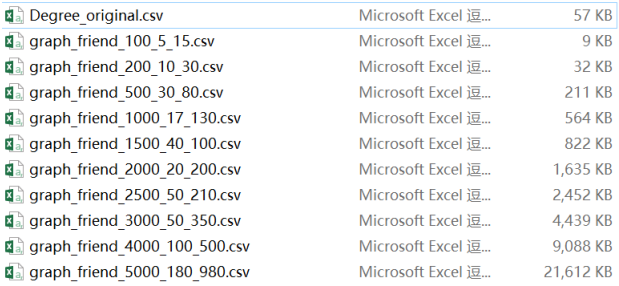
All experiment has been finished on Window 10 operation system. The IDE of the Java program is Eclipse2019-09.This experiment validates the feasibility of dynamic programming algorithms in graph anonymization

Fig. 3. Operation system

Fig. 4. Java IDE

B. Database

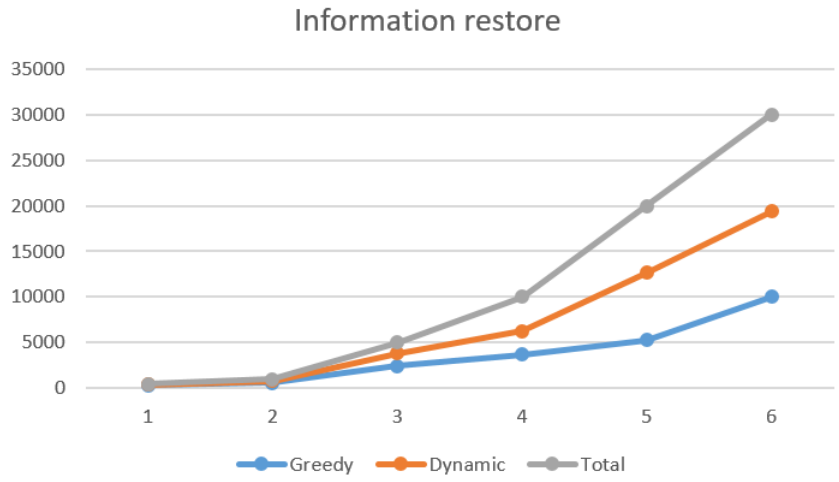
I use .cvs file to keep my test data. I choose several forms of data to test my program and compare different algorithms with each other. All the experimental databases are put in a file, when they are used, we'd better change the route in eclipse to confirm they can work well.



C. Experimental data and analysis

1. First, I tested the working accuracy of two graph anonymization algorithms, greedy algorithm and dynamic programming, with the same database. From the graph, we can conclude

that in most situations, KDA method does a better job than greedy method. Especially when the database becomes larger, KDA can keep more information than greedy method.



algorithm comparison

2. In addition, by changing the value of K, I found that the change in K value will also affect the algorithm's preservation of the relationship graph information. In a certain range, the larger the K value, the greater the loss of information after de-anonymization; conversely, the smaller the K value, the less information is lost, but the effect of anonymization becomes very unsatisfactory, and the relationship is almost impossible The nodes in the graph effectively hide each other's relationship.

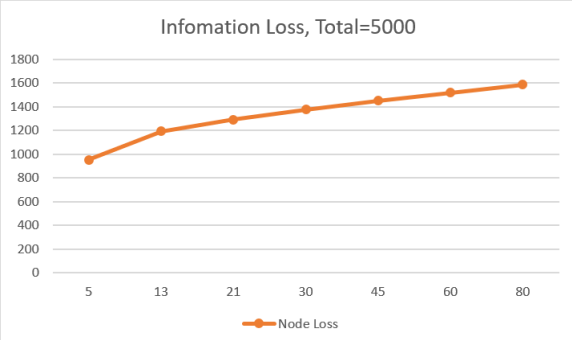


Fig. 5. Relation of K and node loss

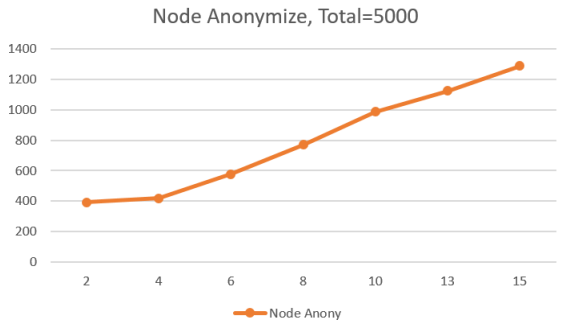
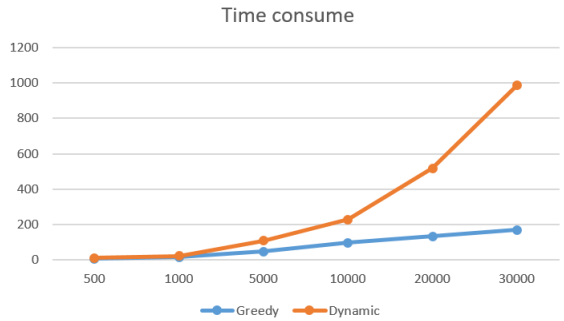


Fig. 6. Relation of K and node anonymization

According to the analysis of these statistical data, it can be seen that when K is 13, it is a most suitable algorithm. At this time, the algorithm found a balance between privacy protection

and information authenticity.

3. In terms of time complexity, the KDA algorithm performs poorly. When the amount of data is small, the calculation time of the KDA algorithm is not much different from that of the greedy algorithm, but when the amount of data increases significantly, the KDA algorithm takes much longer than the greedy algorithm to calculate the final result, which is limited to a certain extent The scope of the KDA algorithm is used.



comparison on time between two algorithms

Therefore, before using the KDA algorithm to process the data, it is best to pre-process the data to reduce the useless work of the KDA algorithm in the data processing stage.

D. summary

The KDA algorithm has a higher ability to ensure the authenticity of the information than the greedy algorithm of the same type.

At the same time, due to the variability of some parameters in the process of anonymizing pictures, to a certain extent, it also guarantees that the anonymized relationship graph is not easily invalidated by partial deanonymization program attacks, and private information is guaranteed.

However, the KDA algorithm is still relatively weak in processing large-scale data. Therefore, if it is to be applied to big data processing, it is necessary to perform a more standardized preprocessing process on the received data, reducing the useless work of the KDA algorithm in the data preprocessing and anonymization process.

VII. FUTURE OUTLOOK

Although I tried my best to complete this experiment, if we look at it from the perspective of practical applications, there are still many things to be improved in this experiment: 1. In this experiment, I paid more attention to the anonymization of the relationships between individuals

(ie nodes) in the relationship graph, so all the algorithms are designed for this purpose. However, in reality, we still need to consider the anonymization of node information, and the KDA algorithm currently does not have an ideal effect on the anonymization of node information, which needs to be improved.

2. The databases used in this experiment are all single-graph structures. In practical applications, we are likely to encounter multi-graph structures. Can the KDA algorithm handle multi-graph structures as well as single-graph structures? It is unknown whether the accuracy can still guarantee the effective anonymization of the relationship graph.

3. Compared with the greedy algorithm, the time complexity of the KDA algorithm is relatively high, and the efficiency will be greatly reduced when the size of the processed data is too large. Although this is a problem of the dynamic programming algorithm itself, I still hope to design an anonymous algorithm that takes into account both execution efficiency and accuracy in the future.

In future research, I also need to pay more attention to the research in the above two directions, so that the KDA algorithm is more complete and better applied to real life.

VIII. THANKS

This is probably the most difficult professional course I have ever taken. After studying for nearly one semester, I finally completed this project. Even though my project dwarfed everyone else. Thanks to the teacher for imparting the cutting-edge knowledge to us and for the patient guidance and support for the problems I encountered in the project. I hope that in the future, I can still keep my teacher's teachings in mind and go further on the road of information security

REFERENCES

[1] Jiang Huowen, *Clustering-anonymity method for data-publishing privacy preservation* , 1School of Mathematics and Computer Science, Jiangxi Science and Technology Normal University, Nanchang, 330038, China

[2] A. Narayanan and V. Shmatikov, *De-anonymizing Social Networks*, S&P 2009.

[3] Debasis Mohapatra, Manas Ranjan Patra, *Anonymization of attributed social graph using anatomy based clustering*, Multimedia Tools and Applications, 2019

[4] Tong WuChao Peng, *Time-Efficient Algorithm on Degree Anonymization by Combination of Vertex and Edge Addition*, S&P 2017-10-27

[5] Mohd Izuan Hafez Ninggal; Jemal H. Abawajy, *Utility-aware social network graph anonymization*, KDD 2011.