Protection of User Data by Differential Privacy Algorithms

Jian Liu[1] and Feilong Qin[2]
(Corresponding author: Feilong Qin)
School of Automobile and Transportation, Chengdu Technological University[1]
School of Big Data and Artificial Intelligence, Chengdu Technological University[2]
No. 1, The second section of Zhongxin Avenue, Pidu District, Chengdu, Sichuan 611730, China
(Email: liujian@cdtu.edu.cn)
(Received Apr. 13, 2019; Revised and Accepted Jan. 5, 2020; First Online July 13, 2020)

Abstract

With the emergence of more and more social software users, increasingly larger social networks have appeared. These social networks contain a large number of sensitive information of users, so privacy protection processing is needed before releasing social network information. This paper introduced the hierarchical random graph (HRG) based differential privacy algorithm and the single-source shortest path based differential privacy algorithm. Then, the performance of the two algorithms was tested by two artificial networks without weight, which was generated by LFR tool and two real networks with weight, which were crawled by crawler software. The results show that after processing the social network through the differential privacy algorithm, the average clustering coefficient decreases, and the expected distortion increases. The smaller the privacy budget, the higher the reduction and the more significant the increase. Under the same privacy budget, the average clustering coefficient and expected distortion of the single-source shortest path differential privacy algorithm are small. In terms of execution efficiency, the larger the size of the social network, the more time it takes, and the differential privacy algorithm based on the single-source shortest path spends less time in the same network.

Keywords: Differential Privacy; Hierarchical Random Graph; Single Source Shortest Path Model; Social Network

1 Introduction

The popularity of wireless communication technology and intelligent mobile terminals makes people's communication more and more convenient, and a variety of community communication application software makes more and more registered users on the Internet, to build a vast and sophisticated social network [1, 12]. Social network contains different kinds of relevant information. Service providers of application software mine information using big data mining technology, analyze users' preferences, and provide more accurate personalized services [9]. However, the social network also contains sensitive private information, which is usually collected and archived by service providers, so the protection measures of privacy and confidential data become critical issues of service providers.

The traditional privacy protection is mainly to encrypt sensitive data, but this method is gradually challenging to play an active role in big data mining technology [13]. Differential privacy algorithm is a method to deal with the above problem. Its basic principle is to disturb the original data and network structure, including adding, deleting, exchanging, etc., to make the disturbing data different from the original data, *i.e.*, protecting original data through publishing the disturbed data. To reduce the large amount of noise caused by separate privacy in related data sets, Zhu *et al.* [15] proposed an effective correlated differential privacy solution. They found that the scheme was superior to the traditional differential privacy scheme in terms of mean square error on a large group of queries. Li *et al.* [7] proposed segmentation mechanisms based on privacy perception and utility to deal with the personalized privacy parameters of every individual in the data set and maximize the efficiency of the differential privacy calculation. Experiments a large amount of original data sets verified the effectiveness of the method.

Chen *et al.* [2] proposed two optimization techniques, PrivTHR and PrivTHREM, to optimize the differential privacy in wave clusters, and the simulation results showed that the optimization technique had high practicability when the privacy budget allocation was appropriate. This paper briefly introduced the differential privacy algorithm based on a hierarchical random graph (HRG) and the differential privacy algorithm based on the single-source shortest path. Then the performance of the two algorithms was tested by two artificial networks without weight, which was generated by LFR tool and two real

networks with weight, which were crawled by crawler software.

2 Differential Privacy Algorithm

2.1 The Concept of Differential Privacy

Social network is a network of points and lines in the visual image. Every node represents a user, while the line represents the connection between users. Points and lines in the social network diagram contain various vital data. At present, the commonly used social network privacy is divided into two categories, both of which substantially change the overall structure of the social network graph. One is to cluster the network nodes into "clusters" by using the clustering algorithm [8] and then encrypt them; the other is to add disturbance to the network graph structure, including deleting, exchanging, and adding nodes and connections. Although the former can hide the privacy data well, it seriously destroys the local structure of the network and affects the typical mining of the network structure data. Although the latter disturbs the network structure, the overall scale is the same, and the impact on the regular use of the data is not significant. Differential privacy is one of the protection methods of the latter. The definition of differential privacy is as follows. If the following equation holds:

$$Pr(F(D_1) \in S) \leq e^{\varepsilon} Pr(F(D_2) \in S).$$

Then the algorithm F can complete ε-differential privacy. D_1 and D_2 are two data sets which only had difference in one data; S is the output result of algorithm F to D_1 and D_2, and it is in the domain of definition of algorithm F; ε is called privacy budget [5], and its value determines the protection degree of differential privacy to data, in details, the smaller the value is, the higher the protection degree is and the larger the disturbance of data addition is.

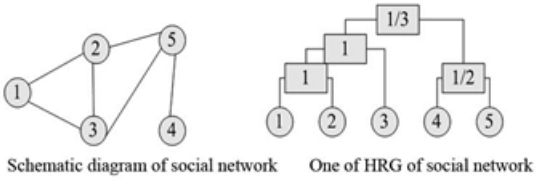

Figure 1: The schematic diagram of social network and one of its HRG

2.2 Differential Privacy Algorithm Based on HRG

HRG [6] divides the hierarchical structure of $G = (V, E)$ using binary tree, in which V is a set of nodes in a network and E is a set of relationships among network nodes. Figure 1 is one kind of HRG in the social network. The division of G by binary tree is similar to the random dichotomy of a node set. As shown in Figure 1, HRG dichotomizes five nodes into (1, 2, 3) and (4, 5) groups. The binary tree root (*i.e.* the box in Figure 1) of the two groups shows the connection probability of the two groups, and the formula is:

$$Pr = e_y/(n_{L,r} \cdot n_{R,r}),$$

where r is the internal node (root node) in the sample tree, *i.e.* the box node of HRG in Figure 1, $n_{L,r}$ and $n_{R,r}$ are the number of network nodes on the left and right sides under root node r, and e_r is the number of connection edges between node sets on both sides of the root node. After that, the dichotomy of groups continues, and the connection probability is calculated until the segmentation completes. The HRG based differential privacy algorithm is as follows.

1) Firstly, the sample tree of HRG is sampled by Markov Chain Monte Carlo (MCMC) method [3], and the details are as follows. A sample tree (HRG) T_0 is randomly selected. Then a neighbor tree is generated according to the previous sample tree, and whether to update the sample tree is determined according to the acceptance probability. The formulas are:

$$\begin{cases} T_i & = \begin{cases} T' & \alpha \\ T_{i-1} & 1-\alpha \end{cases} \\ \alpha & = \min(1, \exp(\frac{\varepsilon_1(\log L(T') - \log L(T_{i-1}))}{})) \\ \log L(T) & = -\sum_{r \in T} n_{L,r} n_{R,r} h(p_r) \\ h(p_r) & = -p_r \log p_r - (1-p_r)\log(1-p_r) \end{cases} \quad (1)$$

where T_i, T_{i-1}, and T' are sample trees after and before the update and the neighbor tree of T_{i-1} respectively, α is the probability of acceptance, $\log L(T)$ is the logarithm of similarity between the sample tree and G, $h(p_r)$ is Gibbs Shannon entropy function. Through Equation (1), the sample tree is updated and iterated until $\log L(T)$ before and after update and iteration is smaller than the set threshold. The number of samples is selected after a certain number of iterations, and finally the stable sample tree set $S_{ST} = (T_{S1}, T_{S2}, \cdots, T_{SN})$ is obtained through sampling, where T_{SN} stands for the sample tree which is obtained by the N-th sampling after the sample tree becomes stable through iterations.

2) Sample tree set S_{ST} is added with Laplace noise [5]. After noise addition, the calculation formula of the connection probability of node r inside the sample tree is:

$$Pr' = \min(1, \frac{e_r + laplace(\varepsilon_2^{-1})}{n_{L,r} \cdot n_{R,r}}),$$

where p'_r is the connection probability of internal node r after noise addition.

3) The lower triangular matrix of every HRG in S_{ST} after noise addition is calculated, and then the lower triangular mean value matrix of S_{ST} is calculated. The element in the lower triangular matrix is the connection probability of each pair of network nodes, which can be obtained through the multiplication of p'_r in HRG.

4) According to the connection probability of network nodes in the lower triangular mean value matrix, the connection edges between nodes is set.

2.3 Single Source Shortest Path Constraint Model Based Differential Privacy Algorithm

The HRG based differential privacy algorithm described above can effectively protect the privacy of social networks, but it is more aimed at the weightless social networks, *i.e.* although the degree of connection between nodes in the social networks in this algorithm is different, the importance of each connection is similar, or it does not matter to the algorithm. However, with the expansion of the scale of the Internet and the increase of social software users, social networks not only increase in scale, but also have different sensitivities between nodes. In order to describe social networks more accurately, in addition to the connection between nodes, different weights are also given to the connection, which is used to indicate the importance of the connection. The original expression of the social network transforms to: $G = (V, E, W)$, where W represents the weight set of the corresponding connection edges. For the social network with weight, the weight that it has is also part of the sensitive information, and moreover it t is also necessary to deal with the weight when dealing with the differential privacy of the social network as the importance degree of connection edges is represented by weight.

The HRG based differential privacy algorithm will affect the edge weight in the processing of differential privacy of social network with weight. Once the edge weight in the social network with weight changes, the structure of the whole graph will change; although the encryption of the information is achieved, the data availability seriously reduces. Therefore, the constraint model of social network was constructed by the single source shortest path algorithm [10] in this study, and linear constraints were applied to the disturbance of differential privacy on the basis of the constraint model. The single source shortest path constraint model based differential privacy algorithm is divided into two steps: 1) Building a single source shortest path constraint model; 2) Adding noise to differential privacy.

1) The first step is to build a single source shortest path constraint model. For $G = (V, E, W)$, nodes in the network are induced into the corresponding spanning tree using Dijkstra algorithm, and the constraint matrix representing the constraints is obtained. The relevant steps are as follows.

   a. Firstly, node is selected from the network as a source point and induced into to set V_0, and then nodes which can be reached in one step from v_0 are selected from the remaining nodes to form set Q.

   b. Node μ which has the smallest edge weight with V_0 is selected from Q. Then a row is added in constraint matrix A according to constraint condition $\{f(v_0, pre\_\mu) \leq f(v_0, \mu)\}$. Values of the corresponding positions in the matrix are constraint coefficients obtained by the set of constraint conditions. pre_μ is μ which is selected from Q previously. Then μ is induced into V_0, and the path between V_0 and Q is updated. Set Q is updated, *i.e.*, deleting μ.

   c. Step 2 repeats until Q becomes an empty set. Then the spanning tree which is composed of nodes in V_0 that has complete induction and corresponding connection edges is added to spanning tree sequence T.

   d. A new source point is selected from the remaining nodes which are not induced, and then Steps 1, 2, and 3 repeat until all the nodes are induced. Finally constraint matrix A and spanning tree sequence T are output.

2) After getting the single source shortest path constraint model of social network, noise is added to differential privacy. The noise addition of social network includes two aspects: one is to add constraint noise to the weight of the network connection edge, and the other is to disturb network nodes. The algorithm steps of the former are as follows.

   a. Firstly, according to the spanning tree in spanning tree sequence T, edge set E in G is divided into E_T and E_N, where E_T is the edge set of spanning tree and E_N is the remaining edge set.

   b. Laplace noise is added to the edge weight in E_T, and the formula of noise addition is:

   $$w'_i = w_i + laplace(\varepsilon_1),$$

   where $S(f)$ stands for the sensitivity of f, i stands for the edge of node pair which accepts search by f, and w_i and w'_i are edge weights of the corresponding node pair before and after noise addition respectively.

   c. The edge weight in E_N is solved based on the weight in E_T after noise addition, constraint matrix A and constraint inequation.

   d. Every spanning tree in spanning tree sequence T is processed as follows. Node pair which has no edge originally in the spanning tree is randomly

Table 1: Data sets of artificial network and Weibo social network

	Number of nodes	Number of edges	Average node degree	The maximum number of nodes in the community	The minimum number of nodes in the community
LFR1	1200	4125	30	110	30
LFR2	5000	9230	35	250	50
Weibo 1	12530	151132	55	1123	122
Weibo 2	21650	213578	64	1624	231

selected. A new edge is added between the node pair. The weight of the new edge is the smaller one among the maximum weight of the spanning tree and the shortest path of node pair.

The algorithm steps of network node disturbance are as follows.

a. Firstly, the number of nodes to be disturbed is calculated according to the set privacy budget,

$$N_n = |laplace(1/\varepsilon_2)|.$$

b. In order to reduce the influence of disturbance such as addition and deletion of nodes on the sensitivity of query function, nodes whose node degree is smaller than the set threshold are selected firstly. Node v is randomly selected, and then node set V_1 which is connected with v is processed as follows.

If $(\mu_1, v) \in E$, $(\mu_2, v) \in E$ and $f(\mu_1, \mu_2) = w(\mu_1, v) + w(v, \mu_2)$, then $w(\mu_1, \mu_2) = w(\mu_1, v) + w(v, \mu_2)$; if there is no edge between μ_1 and μ_2, then an edge is constructed. μ_1 and μ_2 are any two nodes in set V_1. f is a query function in the single source shortest path model, and it returns the shortest path between two nodes. After nodes in V_1 are processed, v and its edge are deleted.

The increase of virtual nodes is as follows. Node v is randomly selected. Then virtual node v_1 is added. A connection line is added between cv and v_1, and the weight of the connection edge is the average value of edge weights of other nodes which connected with v. Moreover, node μ which connects with v is also connected with v_1, and the estimation formula of its weight value is:

$$w(v_1, \mu) = w(\mu, v) + laplace(S(f)/\varepsilon_2).$$

c. Step 2 repeats to disturb network nodes until the number of disturbed nodes reaches N_n. After the noise addition of differential privacy for original social network G, social network G' is output.

3 Simulation Experiment

3.1 Experimental Environment

In this study, the coding of the above algorithm was realized by Python software [4]. The experiment was carried out with a laboratory server which was configured with Core i7 processor (2.6 GHz), Windows 7 operating system and 16 GB memory.

3.2 Experimental Setup

The performance of the two differential privacy algorithms was tested by the artificial network data set generated by LFR tool and the Weibo social network data set crawled by crawler software. The relevant parameters of the artificial network data set generated by LFR and the Weibo social network data set crawled from the Weibo interface by crawler software are shown in Table 1. LFR generated two artificial network data sets, and the artificial network also forms communities of different sizes for simulating the real network. In LFR1, there were 1200 nodes and 4125 edges, with an average node degree of 30; the maximum and minimum number of nodes in the community composed of nodes was 110 and 30 respectively. In LFR2, there were 5000 nodes and 9230 edges, with an average node degree of 35; The maximum and minimum number of nodes in the community composed of nodes was 250 and 50 respectively. The artificial network generated by LFR tool only contained node identification and connection relationship, which belongs to undirected network graph without weight. According to the preliminary statistics of two Weibo data networks which were composed of Weibo data crawled by crawler software, there were 12530 nodes and 151132 edges in Weibo 1, and an average node degree of 55, and the maximum and minimum number of nodes in the community was 1123 and 122 respectively; there were 21650 nodes and 213578 edges in Weibo 2, and an average node degree of 64, and the maximum and minimum number of nodes in the community was 1623 and 231 respectively. Besides the basic node identification and connection relationship, the real network which is composed of Weibo data also included weight information such as attribute labels, and the real network is a social network with weight.

For the above four social network data sets, the soil

networks are processed by the above two differential privacy algorithms. Privacy parameter ε of two algorithms in differential privacy processing was set as 10, 1 and 0.1 respectively.

1) Privacy parameter $\varepsilon = \varepsilon_1 + \varepsilon_2$ was used when the social network was processed by the HRG based differential privacy algorithm (Algorithm 2.2), where $\varepsilon_1 : \varepsilon_2 = 1 : 1$.

2) Privacy parameter $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ was used when the social network was processed by the single source shortest path based differential privacy algorithm (Algorithm 2.3), where $\varepsilon_1 : \varepsilon_2 : \varepsilon_3 = 2 : 1 : 2$.

3.3 Performance Evaluation

In this study, the performance of the two algorithms was measured by average clustering coefficient, expected distortion degree and data processing time. The average clustering coefficient [14] could reflect the structure of social network. Comparing the average clustering coefficient of the network before and after the differential processing could understand the degree of privacy protection of an algorithm; the greater the difference was, the higher the degree of privacy protection was. The formula is:

$$C = \frac{1}{n}\sum_{i=1}^{n}\frac{2E_i}{k_i(k_i-1)},$$

where n is the total number of nodes, E_i is the actual number of connections between nodes adjacent to node i, and k_i is the number of nodes adjacent to node i.

The expected distortion degree [11] could reflect the degree of distortion of the data after differential privacy processing and could measure the availability of data; the larger the value was, the lower the degree of data distortion after processing was and the higher the availability was. The calculation formula is:

$$E[d(X, X')] = \sum_{X}\sum_{X'} p(x)q(x'|x)d(x, x'),$$

where X and X' are data sets before and after differential privacy processing respectively, $d(x, x')$ is the Hamming distance of the data before and after processing, $p(x)$ is the probability distribution of data before processing, and $q(x'|x)$ is the probability of differential privacy transfer condition.

Due to the randomness of the noise added in the differential privacy algorithm, the differential privacy algorithm of each social network was repeated 10 times under different privacy budgets, and the average value was taken as the final result.

3.4 Experimental Results

The average clustering coefficient could reflect the degree of clustering among nodes in the network, and it could reflect the structural distribution of the network to some extent. In this study, two differential privacy algorithms were applied to deal with four kinds of social networks under different privacy budgets. The average clustering coefficient before and after the processing is shown in Figures 2 and 3. Algorithm 2.2 represents the HRG based differential privacy algorithm; Algorithm 2.3 represents the single source shortest path based differential privacy algorithm, and numbers in brackets after the algorithm represent the privacy budget adopted. It was seen from Figures 2 and 3 that the average clustering coefficients of different social networks before and after differential privacy processing were different; the larger the scale of social networks was, the larger the average clustering coefficient was; the average clustering coefficient of real Weibo networks was significantly larger than that of artificial networks, which was because that connections between users in real networks are more close and frequent in addition to the reason of larger scale.

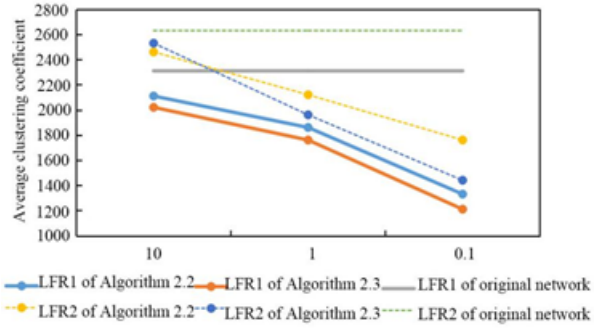

Figure 2: Average clustering coefficients of two LFR obtained by two algorithms under different privacy budgets

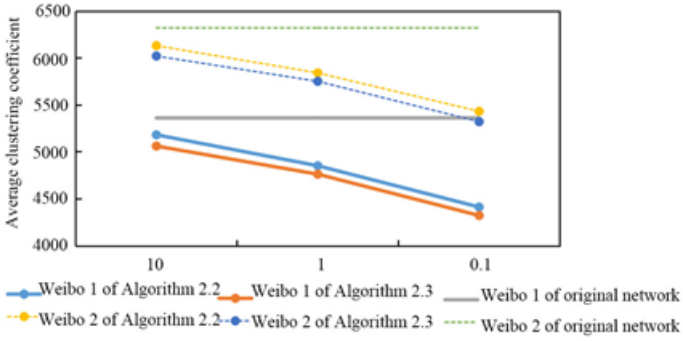

Figure 3: Average clustering coefficients of two Weibo networks obtained by two algorithms under different privacy budgets

The comparison of the average clustering coefficient under the same network data set suggested that the average clustering coefficient after differential privacy processing reduced; the smaller the privacy budget was, the more the reduction was. The comparison of the average clustering coefficient under the same privacy budget suggested that the average clustering coefficient of Algorithm 2.3 in the same social network was smaller. Overall, Algorithm 2.3 was better in the differential privacy protection

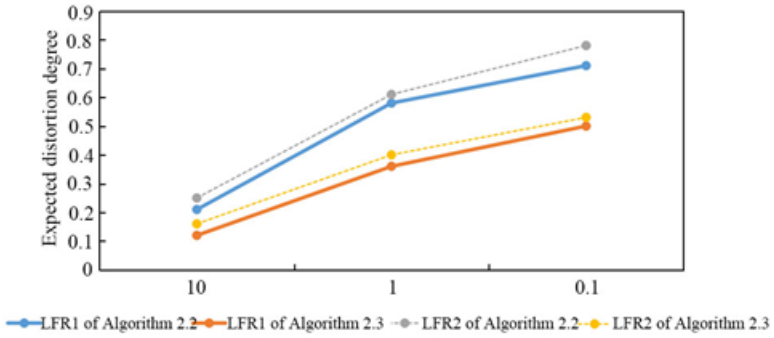

Figure 4: Expected distortion degrees of two LFR obtained by two algorithms under different privacy budgets

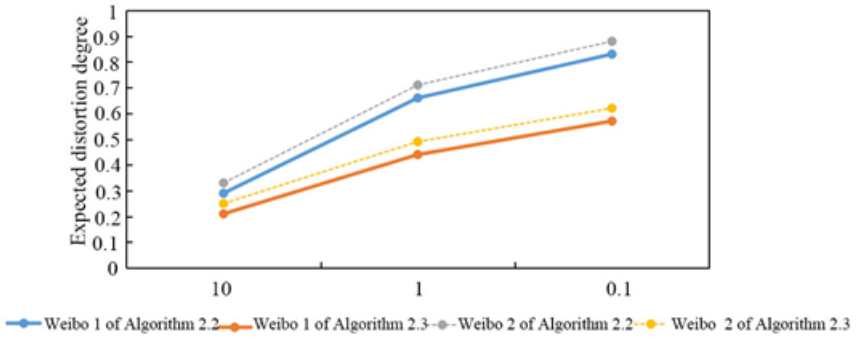

Figure 5: Expected distortion degrees of two Weibo networks obtained by two algorithms under different privacy budgets

of social networks.

The expected distortion degree could reflect the average degree of distortion between the original data set and the data set after differential privacy processing. This index measured the loss degree of effective information in the process of differential privacy processing of social networks. Once the loss degree of effective information was too large, social networks would not have the value of information mining. Under different privacy budgets, the expected distortion degree of the four social networks processed by the two differential privacy algorithms is shown in Figures 4 and 5. It was seen from Figures 4 and 5 that the expected distortion degree increased after differential privacy processing with the reduction of privacy budget no matter what kind of network it was; under the same privacy budget, no matter what kind of network it was, the expected distortion degree of Algorithm 2.3 was smaller; moreover, the expansion of social network scale also increased the expected distortion degree of networks after processing by algorithms.

The purpose of applying differential privacy algorithm to social network is to add noise to the privacy information, so as to achieve the effect of privacy protection. Therefore, in addition to the encryption effect, the execution efficiency of its encryption is also an important performance index. The average time of the two algorithms in processing differential privacy of four networks is shown in Figure 6. It was seen from Figure 6 that the expansion of social network scale and the existence of weights significantly increased the time required for differential privacy processing; under the same social network, the average time required by Algorithm 2.3 was significantly less than that of Algorithm 2.2, *i.e.* the single source shortest path based differential privacy algorithm was more efficient for differential privacy processing of social networks. The HRG based difference privacy algorithm needed to generate neighbor trees constantly in constructing the most matched HRG and sampled after converging to stability; in this process, it takes some time to converge to stability and sample. The single source shortest path constraint model completed at one time without repeated generation and convergence, so it took less time.

4 Conclusion

This paper briefly introduces the differential privacy algorithm based on HRG and the differential privacy algorithm based on a single-source shortest path. Then, two artificial networks without weights generated by LFR Gongzu and two real networks with weights crawled by searcher software were used to test the performance of these two algorithms. The results are as follows.

1) After the two differential privacy algorithms process the community network, the average clustering coefficient is reduced; the lower the privacy budget, the

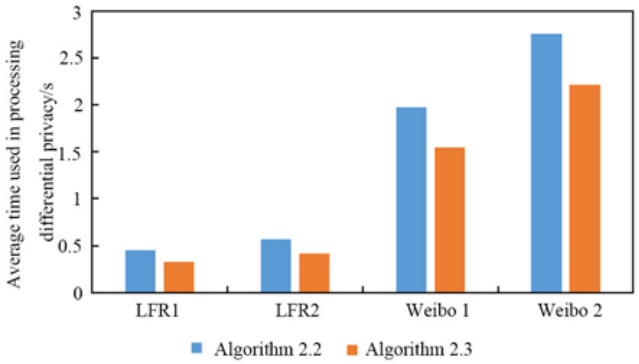

Figure 6: The average time of two algorithms for differential privacy processing

greater the reduction; under the same privacy budget, the single-source shortest path algorithm can reduce more.

2) After the community network is processed by the differential privacy algorithm, the smaller the privacy budget of the algorithm, the greater the expected distortion of the network; under the same privacy budget, the expected distortion of the network processed by the algorithm based on the single-source shortest path is smaller.

3) As the scale of social networks increases, the time required for the two algorithms to process social networks also increases, and the algorithm based on the single-source shortest path requires less time to process the same social network.

References

[1] A. Bhardwaj, V. Avasthi, S. Goundar, "Impact of Social Networking on Indian Youth: A Survey," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 41-51, 2017.

[2] L. Chen, T. Yu, R. Chirkova, "Wave cluster with differential privacy," *Computer Science*, vol. 11, no. 2, pp. 191–198, 2015.

[3] L. Chen, P. Zhu, "Preserving network privacy with a hierarchical structure approach," in *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'15)*, 2015.

[4] G. Eibl, D. Engel, "Differential privacy for real smart metering data," *Computer Science Research & Development*, vol. 32, no. 1-2, pp. 173-182, 2016.

[5] A. Friedman, S. Berkovsky, M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, 2016.

[6] K. Kalantari, L. Sankar, A. D. Sarwate, "Robust privacy-utility tradeoffs under differential privacy and hamming distortion," *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 11, pp. 1-1, 2018.

[7] H. Li, L. Xiong, Z. Ji, X. Jiang, "Partitioning-based mechanisms under personalized differential privacy," pp. 615-627, 2017.

[8] R. Rogers, A. Roth, A. Smith, O. Thakkar, "Max-information, differential privacy, and post-selection hypothesis testing," 2016.

[9] T. Steinke, J. Ullman, "Between pure and approximate differential privacy," *Computer Science*, vol. 8096, no. 2, pp. 363-378, 2015.

[10] X. C. Wang, Y. D. Li, "Geo-social network publication based on differential privacy," *Frontiers of Computer Science*, vol. 12, no. 6, 2018.

[11] X. Wu, Y. Wei, Y. Mao, L. Wang, "A differential privacy DNA motif finding method based on closed frequent patterns," *Cluster Computing*, vol. 21, pp. 1-13, 2018.

[12] B. Yang, I. Sato, H. Nakagawa, "Bayesian differential privacy on correlated data," in *ACM Sigmod International Conference on Management of Data*, 2015.

[13] D. Zhang, D. Kifer, "LightDP: Towards automating differential privacy proofs," *ACM Sigplan Notices*, vol. 52, no. 1, pp. 888-901, 2016.

[14] G. Q. Zhou, S. Qin, H. F. Zhou, "A differential privacy noise dynamic allocation algorithm for big multimedia data," *Multimedia Tools & Applications*, vol. 78, no. C, pp. 1-19, 2018.

[15] T. Zhu, P. Xiong, G. Li, W. Zhou, "Correlated differential privacy: Hiding information in non-IID data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229-242, 2015.

Biography

Jian Liu, born in 1975, has received the doctor's degree. He is a lecturer in Chengdu Technological University. He is interested in Internet of vehicles, big data analysis and risk management.

Feilong Qin, born in 1983, has received the doctor's degree. He is a lecturer in Chengdu Technological University. He is interested in mathematical geology and applied statistics.