Hindawi
Security and Communication Networks
Volume 2019, Article ID 2518714, 9 pages
https://doi.org/10.1155/2019/2518714

Research Article

Differentially Private Release of the Distribution of Clustering Coefficients across Communities

Xiaoye Li,[1,2] **Jing Yang**,[1] **Zhenlong Sun**,[1,2] **and Jianpei Zhang**[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[2]*College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, China*

Correspondence should be addressed to Jing Yang; yangjing@hrbeu.edu.cn

Received 22 March 2018; Accepted 16 December 2018; Published 1 January 2019

Academic Editor: Emanuele Maiorana

Copyright © 2019 Xiaoye Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming to provide more information about the behaviors between groups or patterns between clusters in social networks, we propose a two-step differentially private method to release the distribution of clustering coefficients across communities. The DPLM algorithm improves a Louvain method to partition one network using an exponential mechanism. We introduce an absolute gain of modularity to sanitize neighboring communities. Otherwise, the algorithm is difficult to converge due to the randomness introduced. The DPCC algorithm charts the noisy distribution of clustering coefficients as a histogram, which presents the results in an intuitive manner. We conduct experiments on three real-world datasets to evaluate the proposed method. The experimental results indicate that the proposed method provides valuable distribution results while guaranteeing ε-differential privacy. Moreover, the DPLM algorithm can obtain better modularity for the networks.

1. Introduction

The understanding of the quantitative and qualitative characteristics of social networks has become an important challenge in the scientific research of the Internet era. The research content includes many aspects by means of the basic measures of complex networks. For example, the counts of triangles or other simple subgraphs can be used to characterize the connectivity of a graph. Meanwhile, various subgraph counts are the core data in graph analysis and also the parameters of random graph models. The clustering coefficient, a reflection of social cohesion, measures whether nodes in the graph tend to cluster together. The detected communities can assist in studying the organization and function of complex networks.

The release of graph measures may violate the privacy of individuals in social networks. Some algorithms have been proposed to address this problem. We focus on the schemes providing ε-differential privacy [1], a prominent concept, which is discussed largely in the computer field. Initially, differential privacy is used to protect the output of queries in an interactive environment. Karwa et al. [2] provided differentially private approaches for releasing k-star and k-triangle counts, respectively. A k-star subgraph has a central node with k connected nodes. A k-triangle subgraph means k triangles share one edge. The two approaches are based on smooth sensitivity [3] and a higher-order local sensitivity, respectively. Shoaran et al. [4] provided zero-knowledge private [5] methods for releasing a group-based triangle measure, which is the fraction of the number of actual triangles over the number of all possible triangles. Note that the nodes of such triangles belong to different groups. We consider the clustering coefficient as the graph measure in this study and provide an approach to protect link privacy during release.

Task et al. [6] proposed the concept of partition privacy, which provides broader protection at the level of small social groups rather than individuals. They released various graph measures in the form of histograms, such as triangle density, average shortest-path lengths, and subgraph counts. It should be noted that they ran experiments over a collection of graphs rather than one network. The accessible datasets are nonpartitioned graphs, which require developing differentially private partitioning algorithms. Mülle et al.

[7] proposed an approach for perturbing the input graph by operating on the adjacency matrix. The approach is a combination of edge sampling and edge flipping, which is essentially an edge randomization method. Then, the graph clustering algorithms are applied directly to the perturbed graph. Nguyen et al. [8] addressed the problem of detecting communities under differential privacy. They proposed two schemes, input perturbation and algorithm perturbation. In addition, another category is output perturbation. (1) They applied a high-pass filtering technique [9] to create a noisy weighted super-graph. Then, the original Louvain method [10] was run on the super-graph. (2) To heuristically detect cohesive groups in a private manner, they proposed a divisive algorithm ModDivisive by realizing an exponential mechanism via Markov Chain Monte Carlo (MCMC). The modularity was used as a score function and the global sensitivity was also demonstrated. We improve on the Louvain method, one of the most cited methods for community detection, to implement a differentially private partitioning task for one network.

In this paper, we propose a novel method for differentially private release of the distribution of clustering coefficients across communities. The method partitions one network into several communities and then releases the histogram of clustering coefficients. It is more meaningful to compare with an average clustering coefficient of an entire network, as it may provide more information about the behaviors between groups or patterns between clusters in social networks. The rest of this paper is organized as follows. Section 2 introduces the background knowledge. The proposed method is presented in Section 3. Section 4 reports the experimental results. Finally, Section 5 concludes the study and provides additional research directions.

2. Background

In this section, we first review the definition of differential privacy and some relevant concepts. Then, we introduce the calculation method of a clustering coefficient. Finally, we demonstrate the partition process of the Louvain method.

2.1. Differential Privacy. Differential privacy is based on a mathematical foundation and can provide proven security as cryptography does. The probability of the same results will not change significantly, whether a record is in the dataset or not. It is difficult to provide further reasoning for any potential adversary according to background knowledge.

Definition 1 (ε-differential privacy [1]). A randomized algorithm K satisfies ε-differential privacy if, for all neighboring datasets D_1 and D_2 differing by at most one record, and for all subsets of possible outputs $S \subseteq Range(K)$,

$$\Pr\left[K\left(D_1\right) \in S\right] \leq \exp\left(\varepsilon\right) \times \Pr\left[K\left(D_2\right) \in S\right] \tag{1}$$

where ε is a tuning parameter to make the trade-off between privacy and accuracy. It is a small positive value; a smaller value yields a higher privacy and lower accuracy, and vice versa.

In the context of social networks, differential privacy is adapted to edge-differential privacy and node-differential privacy in the literature [11]. We adopt the former conception, edge-DP for short, to protect individual edges from being disclosed. In the definition, a neighboring graph is produced either by adding or removing an edge or by adding or removing an isolated node.

To achieve differential privacy, noise mechanisms need to be introduced. The magnitude of noise required is dependent on the global sensitivity. The common techniques are the Laplace mechanism and the exponential mechanism. Furthermore, differential privacy contains two important combination properties, sequential composition and parallel combination. The relevant definitions are formally described as below.

Definition 2 (global sensitivity [1]). For a function $f : D \longrightarrow R^d$, the global sensitivity of f is

$$\triangle f = \max_{D,D'} \left\| f\left(D\right) - f\left(D'\right) \right\|_1 \tag{2}$$

where R^d is d dimensional real vector and D and D' are neighboring datasets. Global sensitivity represents the largest change that a single record could have on the outputs.

Definition 3 (Laplace mechanism [12]). For a function $f : D \longrightarrow R^d$, the randomized algorithm M satisfies ε-differential privacy,

$$M\left(D\right) = f\left(D\right) + \left(Lap\left(\frac{\Delta f}{\varepsilon}\right)\right)^d \tag{3}$$

where $Lap(\Delta f/\varepsilon)$ is a random variable sampled from the Laplace distribution with mean 0 and scale parameter $\Delta f/\varepsilon$. The Laplace mechanism perturbs the numerical outputs by adding noise to guarantee ε-differential privacy.

Definition 4 (exponential mechanism [13]). Let $q : (D \times O) \longrightarrow R$ be a scoring function, and the randomized algorithm M satisfies ε-differential privacy,

$$M\left(D, q\right) = \left\{r \mid \Pr\left[r \in O\right] \propto \exp\left(\frac{\varepsilon q\left(D, r\right)}{2\Delta q}\right)\right\} \tag{4}$$

where the probability that r is selected is proportional to $\exp(\varepsilon q(D, r)/2\Delta q)$; a higher score means a greater probability of being selected. The exponential mechanism is applicable to discrete outputs, which ensures that an output is selected in a differentially private manner.

Proposition 5 (sequential composition [14]). *Let each algorithm A_i provide ε_i - differential privacy. The combination algorithm A (A_1 (D), A_2 (D),…) over the entire dataset D provides $\Sigma\varepsilon_i$ - differential privacy.*

Proposition 6 (parallel composition [14]). *Let each algorithm A_i provide ε_i - differential privacy. The combination algorithm A (A_1 (D_1), A_2 (D_2),…) over the disjoint subsets of dataset D provides* max ε_i *- differential privacy.*

In solving complex privacy problems, we need to combine several differentially private mechanisms and properly allocate a privacy budget ε to every portion based on the combination properties.

2.2. Clustering Coefficient. There are two versions of calculation methods of clustering coefficients. The global approach is to measure the clustering of the entire network. The local approach provides a measure of the embedding of a single node, and the mean is for measuring the entire network.

(1) Global clustering coefficient is

$$C = \frac{\#\ of\ triangles}{\#\ of\ triangles + \#\ of\ wedges} \tag{5}$$

(a) The wedge subgraph ($\vee$) is a chain of 3 nodes connected by 2 edges.

(b) The triangle subgraph ($\triangle$) is a clique of 3 nodes connected by 3 edges.

(2) Local clustering coefficient is

$$C_i = \frac{2 * \left|\{e_{st}\}\right|}{\left|N_i\right| * \left(\left|N_i\right| - 1\right)}: \quad v_s, v_t \epsilon N_i, e_{st} \epsilon E,$$
$$\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{6}$$

(a) N_i is the set of direct neighboring nodes of v_i.

(b) E is the set of edges.

2.3. Louvain Method. The Louvain method [10] is a heuristic algorithm based on modularity optimization to implement a community detection task. The algorithm can discover high quality partitions in a short time and unfold a complete hierarchical community structure for large networks.

The algorithm is divided into two phases for each pass, the first phase optimizes modularity until a local maximum is attained, and the second phase aggregates communities to build a new weighted network. In the initial partition, each node is treated as a different community. The passes are repeated iteratively to reach the final partition, the top level of the hierarchy. The Louvain method is shown in Figure 1. The algorithm runs two passes in this example. In the first pass, the network is partitioned into three communities with a modularity of 0.3291. The algorithm continues for the second pass based on the new weighted network. The modularity of the final partition is 0.3571, and then the algorithm ends.

(1) Modularity. The modularity is used to measure the quality of the partitions and also as an objective function to optimize. The definition of modularity Q [15] for a weighted network is as below:

$$\begin{aligned} Q &= \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta\left(c_i, c_j\right) \\ &= \frac{1}{2m}\left[\sum_{i,j} A_{ij} - \frac{\sum_i k_i \sum_j k_j}{2m}\right]\delta\left(c_i, c_j\right) \\ &= \frac{1}{2m}\sum_{C}\left[\sum in - \frac{\left(\sum tot\right)^2}{2m}\right] \end{aligned} \tag{7}$$

where A_{ij} represents the edge-weight between nodes i and j, k_i is the sum of the adjacent edge-weights of node i, m is the sum of the edge-weights in the network, c_i is the community where node i is located, the function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise, $\sum in$ represents the sum of the edge-weights in community C, and $\sum tot$ represents the sum of the adjacent edge-weights of nodes in community C.

(2) Relative Gain. In the Louvain method, the gain of modularity is computed by moving an isolated node i into a community C. It considers all communities where the neighbors of node i are located. A node i is placed in the community with the maximum gain, only if the gain is positive. A node i may stay in the original community if no positive gain is obtained. The operation is repeated iteratively and sequentially for all nodes until no more moving can improve the modularity, and then the first phase is complete. The formula for calculating ΔQ is as below:

$$\begin{aligned} \Delta Q &= \left[\frac{\sum in + k_{i,in}}{2m} - \left(\frac{\sum tot + k_i}{2m}\right)^2\right] \\ &\quad - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \\ &= \frac{\sum in}{2m} + \frac{k_{i,in}}{2m} - \left(\frac{\sum tot}{2m}\right)^2 - \frac{\sum tot \bullet k_i}{2m^2} - \left(\frac{k_i}{2m}\right)^2 \\ &\quad - \frac{\sum in}{2m} + \left(\frac{\sum tot}{2m}\right)^2 + \left(\frac{k_i}{2m}\right)^2 \\ &= \frac{k_{i,in}}{2m} - \frac{\sum tot \bullet k_i}{2m^2} = \frac{1}{2m}\left[k_{i,in} - \frac{\sum tot \bullet k_i}{m}\right] \end{aligned} \tag{8}$$

where $k_{i,in}$ is the sum of the edge-weights between node i and nodes in community C, and other notations represent the same meaning as formula (7).

3. Methods

In this section, we first analyze the concept of absolute gain and its significance to our improved method. Then, we provide the DPLM algorithm to partition one network based on the exponential mechanism. Finally, we provide the DPCC algorithm to release the distribution of clustering coefficients based on the Laplace mechanism. The data flowchart of the proposed method is shown in Figure 2.

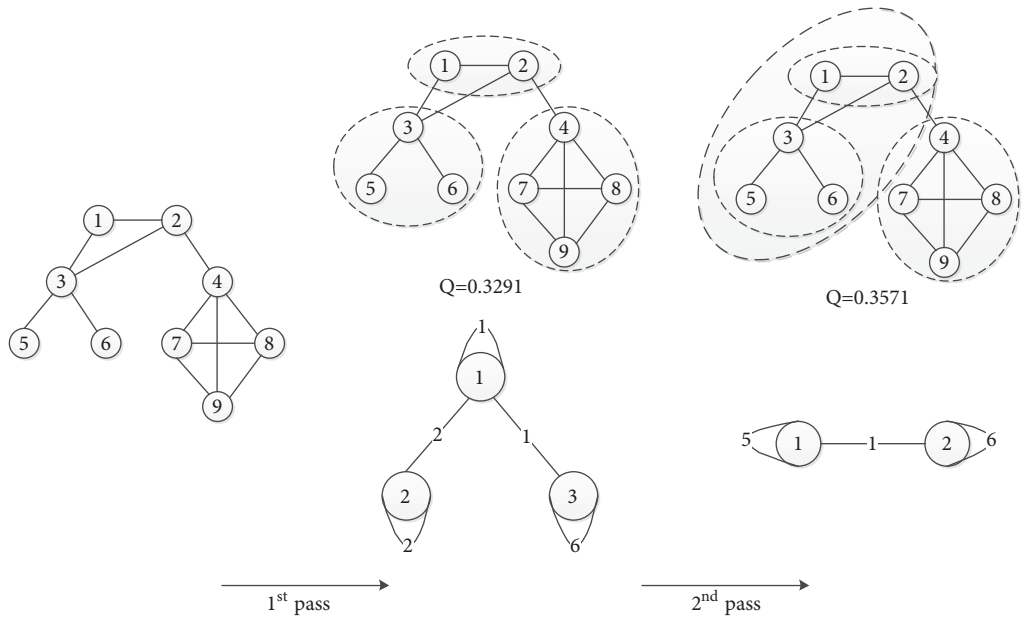


Figure 1: The Louvain method.

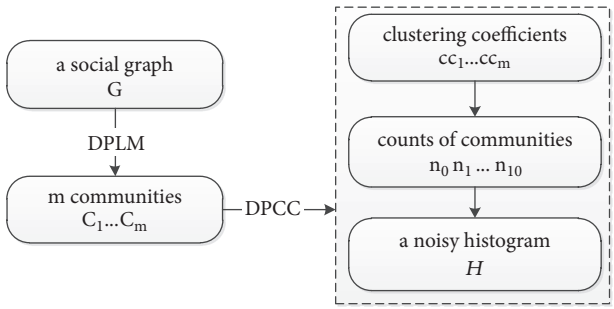


Figure 2: The data flowchart of the proposed method.

3.1. Absolute Gain. In the Louvain method, the modularity gain is relative. It considers the maximum gain by moving one node into a neighboring community, regardless of the change of modularity by removing the node from its own community. In order to simplify the calculation, we treat each node as an isolated node, which does not belong to any community. The change can be evaluated with the gain ΔQ_{ori} by supposing to move the node into its original community. We consider the absolute gain of modularity, which is the difference between two gains, $\Delta Q - \Delta Q_{ori}$, not just ΔQ.

A positive value of absolute gain guarantees that the modularity gain increases absolutely for the entire network. The formula $\Delta Q - \Delta Q_{ori} > 0$ is equivalent to $\Delta Q > \Delta Q_{ori}$, where ΔQ_{ori} is a computable threshold. This simple heuristic is beneficial for the exponential mechanism. The randomness introduced dramatically increases the number of iterations in the first phase. The constraint sanitizes neighboring communities and reduces the range of optional communities, which avoids the situation where one node may be moved multiple times in vain.

3.2. Differentially Private Partitioning. We improve the Louvain method to partition one network under the premise of ε-differential privacy. As shown in Algorithm 1, lines (1) ~ (9) describe the first phase, and line (10) describes the second phase. There are two major improvements in the first phase. The neighboring communities are firstly sanitized according to $\Delta Q_j - \Delta Q_{ori} > 0$ (see line (5)). Then, a community is randomly selected based on the exponential mechanism (see line (6)). The score function is ΔQ and the global sensitivity is smaller than $1/m$, which is proved by Theorem 7.

Note that the algorithm consumes the privacy budget ε_1 at each iteration of the first phase. In the second phase, no privacy budget is consumed, similar to the Louvain method. As described in lines (11)~(12), the algorithm will perform some passes repeatedly and finally return the partition P with a maximum Q.

Theorem 7. *The global sensitivity of the gain of modularity,* $\Delta(\Delta Q)$*, is smaller than* $1/m$*.*

Proof. In edge-DP, G and G' are neighbors if $|V \oplus V'| + |E \oplus E'| = 1$. There are two cases to consider separately.

Case 1. G' is produced by adding or removing an isolated node. There is no effect on ΔQ according to the definition. That is, the maximum change is 0.

Case 2. G' is produced by adding or removing an edge. Then,

$$\Delta Q = \frac{1}{2m}\left[k_{i,in} - \frac{\sum tot \bullet k_i}{m}\right] = \frac{k_{i,in}}{2m} - \frac{\sum tot \bullet k_i}{2m^2} \leq \frac{k_{i,in}}{2m} \tag{9}$$

Input: the graph dataset G, the privacy budget ε_1
Output: a private partition $P\ (C_1 \dots C_m)$
(1) do until no change about all communities
(2) for each node i in graph G
(3) find the neighboring communities $C_1 \dots C_k$
(4) compute the gains $\Delta Q_1 \dots \Delta Q_k$, ΔQ_{ori}
(5) remain C_j with $\Delta Q_j - \Delta Q_{ori} > 0$, $j = 1 \dots k$
(6) select community C with probability $\exp(\varepsilon_1 \Delta Q / (2/m))$
(7) move node i into community C
(8) end
(9) end
(10) aggregate the found communities to build a new weighted graph G'
(11) repeat lines (1) ~ (10) until no improvement of Q
(12) return $P\ (C_1 \dots C_m)$

Algorithm 1: The DPLM algorithm.

Since adding or removing operation is similar, only the former is considered in the proof. The maximum impact on ΔQ is that a new edge is incident to the node i within the community C. The degrees of the two nodes of the new edge increase by 1, respectively. In this case, the value of $k_{i,in}$ increases by 2, and the total number of edges is $m + 1$. Therefore, the maximum amount of change for ΔQ is

$$\frac{k_{i,in}}{2m} = \frac{2}{2(m+1)} = \frac{1}{m+1} < \frac{1}{m} \tag{10}$$

In summary, the global sensitivity of the gain of modularity, Δ(ΔQ), is smaller than $1/m$.

Note that the time complexity of the improved algorithm DPLM remains unchanged. The time complexity in each iteration is O (E), where E is the number of edges in the graph. The most time-consuming computation is the first iteration, which is the lowest layer of community detection. After a few iterations, the number of communities decreases dramatically in the graph. □

3.3. Releasing a Noisy Histogram. Next, we release a noisy histogram of clustering coefficients based on the previous partitioning results. In the previous process, the consumption of privacy budget is $t \bullet \varepsilon_1$, where t is the number of iterations. The remaining privacy budget ε_2 is $\varepsilon - t \bullet \varepsilon_1$, where ε is the total privacy budget of the proposed method.

As shown in Algorithm 2, we firstly count communities with the calculated clustering coefficients in lines (1)~(2). Then, the counts are disturbed based on the Laplace mechanism in line (3). The global sensitivity of the released histogram is 2, which is proved by Theorem 8. Finally, the algorithm charts the noisy distribution of clustering coefficients across communities in line (4).

Theorem 8. *The global sensitivity of the released histogram is 2.*

Proof. As the proof of Theorem 7, there are two cases to consider separately.

Case 1. G' is produced by adding or removing an isolated node. The isolated node only changes the first bin n_0 of the histogram by 1.

Case 2. G' is produced by adding or removing an edge. The edge only changes the clustering coefficient of one community after partitioning. It is like the community shifting to a different bin of the histogram. The value of the bin adds 1 and another one subtracts 1. Therefore, the total change of the histogram is 2.

In summary, the global sensitivity of the released histogram is 2. □

Theorem 9. *The proposed method guarantees ε- differential privacy.*

Proof. The proposed method includes two steps, differentially private partitioning and releasing a noisy histogram.

Specifically, in the first step, the consumption of the privacy budget is ε_1 at each iteration according to Proposition 6. The number of iterations t is numerable; the total consumption of the privacy budget is $t \bullet \varepsilon_1$ according to Proposition 5. For the second step, the assigned privacy budget ε_2 is $\varepsilon - t \bullet \varepsilon_1$, as seen in Algorithm 2. According to Proposition 5, the total privacy budget is the sum of the two parts, $t \bullet \varepsilon_1 + \varepsilon_2 = \varepsilon$. Therefore, the proposed method guarantees ε- differential privacy. □

4. Experiments

We conduct experiments on three real-world datasets: CA-GrQc, CA-HepTh, and CA-HepPh, which can be accessed from the Stanford Large Network Dataset Collection (SNAP http://snap.stanford.edu/data/) without any restrictions. The file CA-GrQc.txt contains the collaboration network of Arxiv General Relativity category. The file CA-HepTh.txt contains the collaboration network of Arxiv High Energy Physics Theory category. The file CA-HepPh.txt contains the collaboration network of Arxiv High Energy Physics category. In

Input: the private partition P $(C_1 \ldots C_m)$, the privacy budget $\varepsilon_2 = \varepsilon - t \bullet \varepsilon_1$
Output: a noisy histogram H
(1) Calculate clustering coefficients $cc_1 \ldots cc_m$ for each community $C_1 \ldots C_m$
// The clustering coefficient is rounded to one decimal place.
(2) Count communities with given clustering coefficients *0,0.1,0.2,. . ., 1*
// The expression $n_0 + n_1 + \ldots + n_{10} = m$ is valid.
(3) Add $Lap(2/\varepsilon_2)$ to the counts of communities $n_0, n_1 \ldots n_{10}$
(4) Chart the distribution of results as a histogram H

Algorithm 2: The DPCC algorithm.

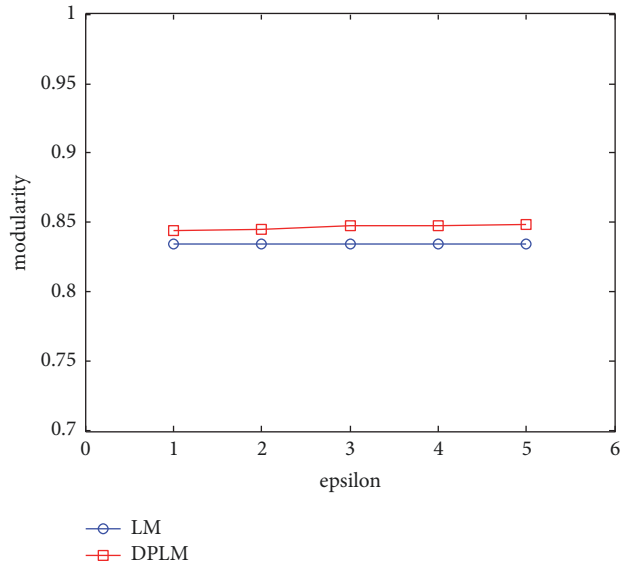


Figure 3: The modularity metric (CA-GrQc).

these collaboration networks, every author is considered as a node, and there is an edge if authors coauthored at least one paper. The statistics of the datasets are shown in Table 1.

In this section, we evaluate two methods for releasing the distribution of clustering coefficients across communities. The proposed method includes two algorithms, DPLM and DPCC. The comparative method directly calls the Louvain method (LM for short) to partition one network and releases the results without noise. With a slight abuse of concepts in Figures 5, 8, and 11, we will use LM and DPLM to denote the two methods, respectively.

In the experiments, the total privacy budget ε is between 1 and 5, and ε_1 is set to one percent of ε. That is, ε_1 is between 0.01 and 0.05 correspondingly. The number of iterations is shown in Table 2 in the first partitioning step. The privacy budget is allocated as follows: for example, when ε=3 is set, the number of iterations is 24 for CA-HepTh. In the first partitioning step, the consumption of the privacy budget is 24∗0.03= 0.72. For the second releasing step, the remaining privacy budget ε_2 is 3- 0.72=2.28.

The modularity of the partitioning results is shown in Figures 3, 6, and 9. The two curves are almost parallel, which indicates that the privacy budget ε has little impact on modularity. The modularity obtained by DPLM is better

Table 1: The statistics of datasets.

Datasets	Nodes	Edges	Average clustering coefficients
CA-GrQc	5242	14496	0.5296
CA-HepTh	9877	25998	0.4714
CA-HepPh	12008	118521	0.6115

Table 2: The number of iterations.

Datasets	LM	DPLM				
		ε=1	ε=2	ε=3	ε=4	ε=5
CA-GrQc	12	19	19	24	18	20
CA-HepTh	19	24	29	24	28	26
CA-HepPh	14	28	25	27	29	24

than LM, especially for CA-GrQc and CA-HepTh. As for CA-HepPh, there is little difference in modularity between the two methods.

As shown in Figures 4, 7 and 10, the number of communities partitioned by DPLM is about 100 less than LM. Then, the size of the community in DPLM is relatively larger than LM. As shown in Table 2, the number of iterations performed in DPLM is more than LM, or even twice in some cases. Undoubtedly, DPLM consumes more running time than LM. In DPLM, the program can iterate more times to detect larger communities and obtain better modularity for the networks. The reason may be that the exponential mechanism provides more testing opportunities to avoid falling into local optimizations. Note that it is not feasible to simply introduce the exponential mechanism into the Louvain method. The algorithm will be difficult to converge if the absolute gain of modularity is not guaranteed to be positive.

As shown in Figures 5, 8, and 11, nearly half of the communities are located at both ends of the histogram, while the remaining half are normally distributed in the middle part of the histogram. Intuitively, the relative difference is uniform between the two methods. Therefore, the two methods are consistent with the released results. In conclusion, the proposed method provides valuable distribution results while guaranteeing ε- differential privacy.

5. Conclusions and Future Work

In this paper, we addressed a differentially private method for releasing the distribution of clustering coefficients across

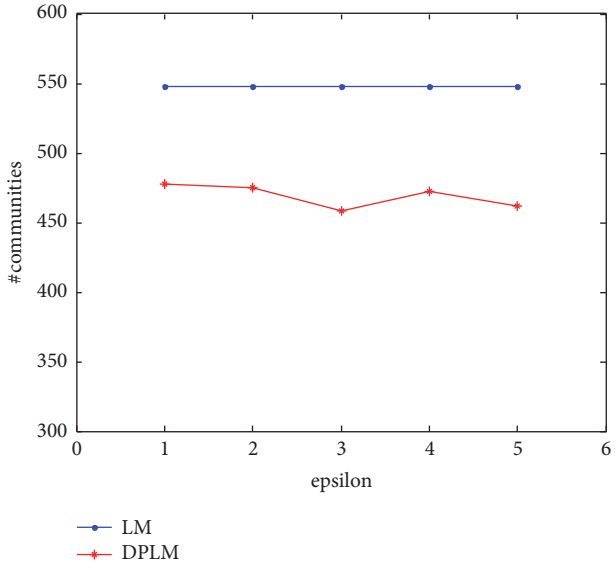


Figure 4: The number of communities (CA-GrQc).

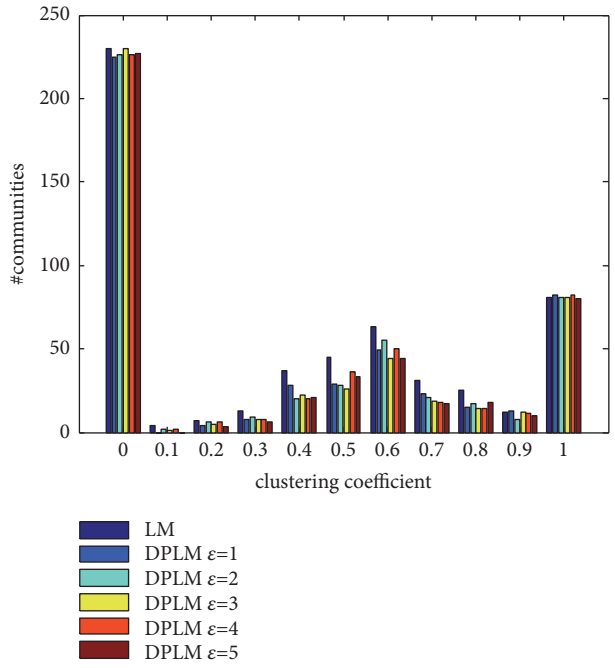


Figure 5: The histogram of clustering coefficients (CA-GrQc).

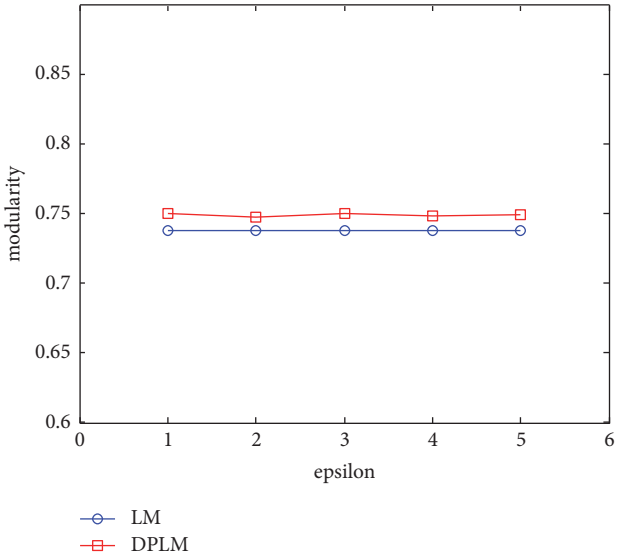


Figure 6: The modularity metric (CA- HepTh).

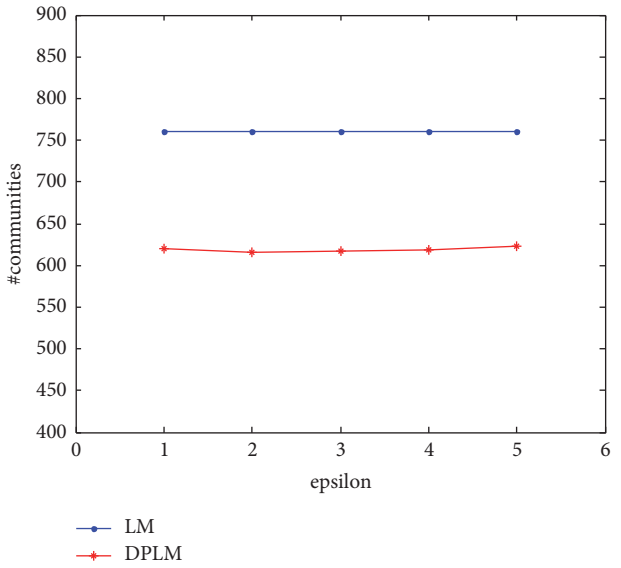


Figure 7: The number of communities (CA- HepTh).

communities. We improved the Louvain method to partition one network by using an exponential mechanism. The neighboring communities are sanitized according to the absolute gain, which is different from the relative gain used in the original algorithm. This change is crucial given the randomness of the exponential mechanism. Finally, the method outputs the histogram of clustering coefficients based on the Laplace mechanism.

By conducting a comprehensive evaluation, the proposed method was shown to provide valuable distribution results while guaranteeing ε-differential privacy. Moreover, the DPLM algorithm can obtain better modularity for the networks, which is also an improvement for community detection. Nevertheless, the accuracy of the partitioning results needs to be further tested with some auxiliary structural information. As part of further study, we plan to extend our work to other community detection algorithms [16–18] or develop new differentially private partitioning algorithms.

Data Availability

The CA-GrQc, CA-HepTh, and CA-HepPh data used to support the findings of this study can be accessed from the Stanford Large Network Dataset Collection (http://snap.stanford.edu/data/) without any restrictions.

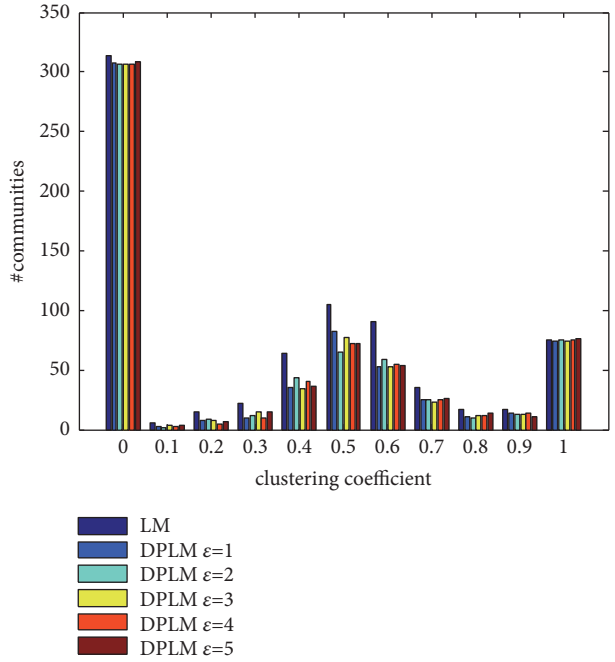


Figure 8: The histogram of clustering coefficients (CA- HepTh).

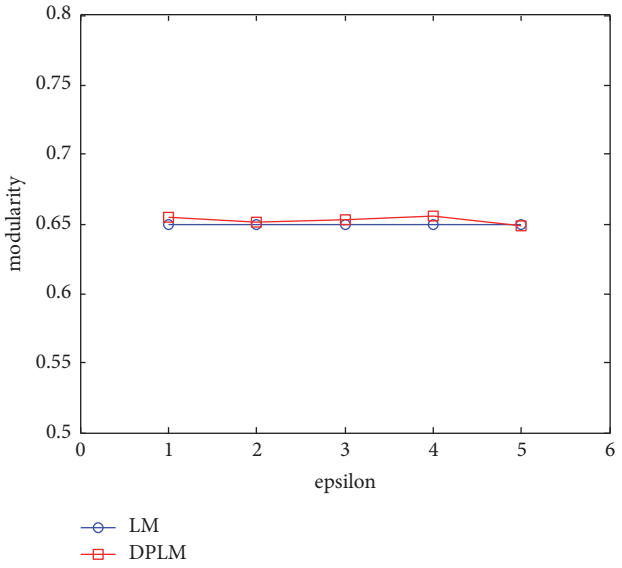


Figure 9: The modularity metric (CA- HepPh).

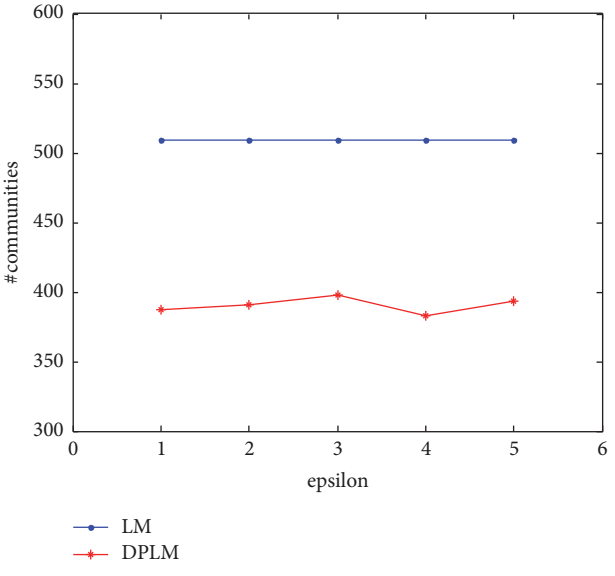


Figure 10: The number of communities (CA- HepPh).

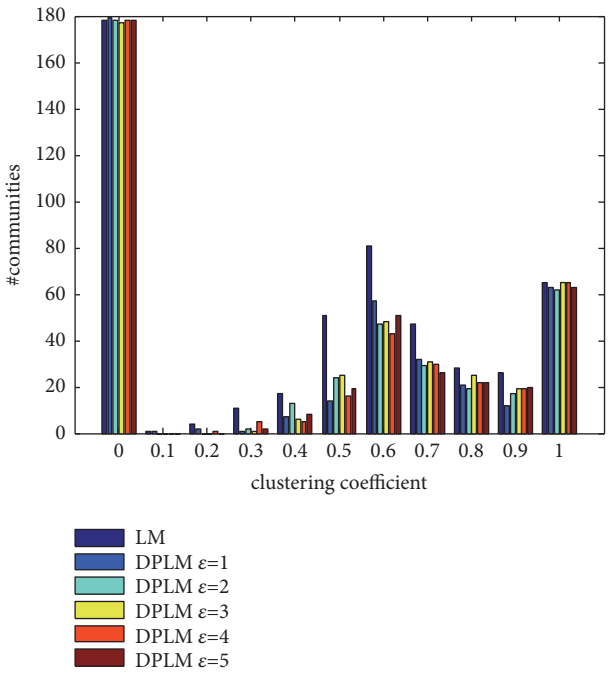


Figure 11: The histogram of clustering coefficients (CA- HepPh).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by National Natural Science Foundation of China (No. 61672179, No. 61370083, and No. 61402126), Natural Science Foundation of Heilongjiang Province (No. F2015030), Youth Science Fund of Heilongjiang Province (No. QC2016083, No. QC2017079), Postdoctoral Fellowship of Heilongjiang Province (No. LBH - Z14071), and The Fundamental Research Funds in Heilongjiang Provincial Universities (No. 135109245, No. 135109314).

References

[1] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata Languages and Programming*, pp. 1–12, Springer, Heidelberg, Berlin, Germany, 2006.

[2] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, 2011.

[3] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC '07)*, pp. 75–84, ACM, 2007.

[4] M. Shoaran and A. Thomo, "Zero-knowledge-private counting of group triangles in social networks," *The Computer Journal*, vol. 60, no. 1, pp. 126–134, 2017.

[5] J. Gehrke, E. Lui, and R. Pass, "Towards privacy for social networks: a zero-knowledge based definition of privacy," in *Proceedings of the Theory of Cryptography Conference*, pp. 432–449, TCC, 2011.

[6] C. Task, "Privacy-preserving social network analysis," *Dissertations and Theses - Gradworks*, 2015.

[7] Y. Mülle, C. Clifton, and K. Böhm, "Privacy-Integrated Graph Clustering Through Differential Privacy," in *Proceedings of the EDBT/ICDT 2015 Joint Conference*, Brussels, Belgium, 2015.

[8] H. H. Nguyen, A. Imine, and M. Rusinowitch, "Detecting communities under differential privacy," in *Proceedings of the 15th ACM Workshop on Privacy in the Electronic Society, WPES 2016*, pp. 83–93, ACM, 2016.

[9] G. Cormode, C. Procopiuc, D. Srivastava, and T. T. Tran, "Differentially private summaries for sparse data," in *Proceedings of the International Conference on Database Theory*, pp. 299–311, ACM, New York, NY, USA, 2012.

[10] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. 155–168, 2008.

[11] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, vol. 120, pp. 169–178, IEEE, Miami, FL, USA, December 2009.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography*, pp. 265–284, Springer, New York, NY, USA, 2006.

[13] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS '07)*, pp. 94–103, IEEE, Providence, RI, USA, October 2007.

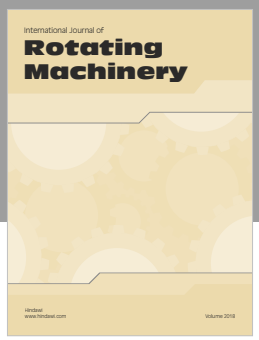
[14] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems, SIGMOD-PODS'09*, pp. 19–30, ACM, July 2009.

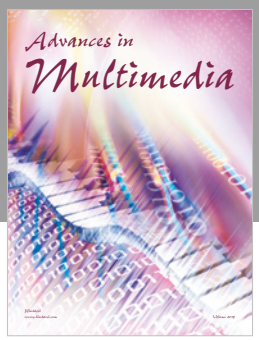
[15] M. E. Newman, "Analysis of weighted networks," *Physical Review E Statistical Nonlinear and Soft Matter Physics*, vol. 70, Article ID 056131, 2004.

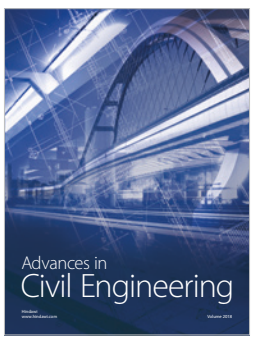
[16] D. Liu, H.-Y. Bai, H.-J. Li, and W.-J. Wang, "Semi-supervised community detection using label propagation," *International Journal of Modern Physics B*, vol. 28, no. 29, Article ID 1450208, 2014.

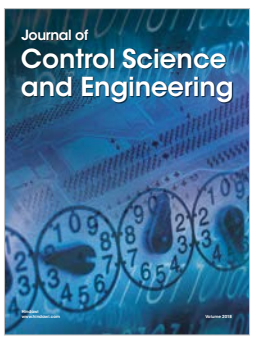
[17] H. Li, Y. Wang, L. Wu, Z. Liu, L. Chen, and X. Zhang, "Community structure detection based on Potts model and network's spectral characterization," *EPL (Europhysics Letters)*, vol. 97, no. 4, pp. 48005–48010, 2012.

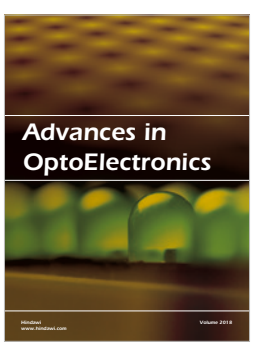
[18] H.-J. Li, Z. Bu, A. Li, Z. Liu, and Y. Shi, "Fast and accurate mining the community structure: integrating center locating and membership optimization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2349–2362, 2016.

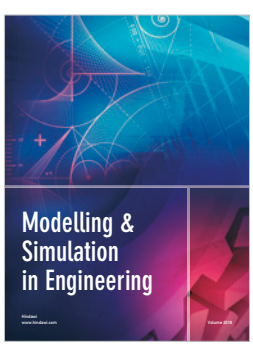


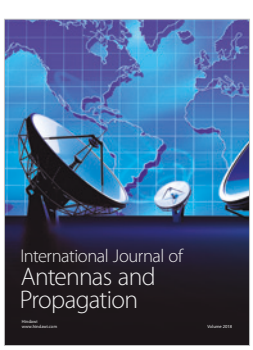


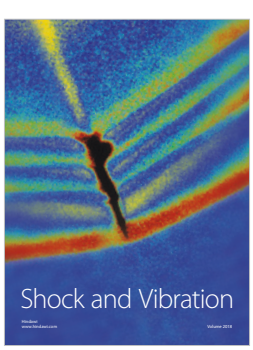


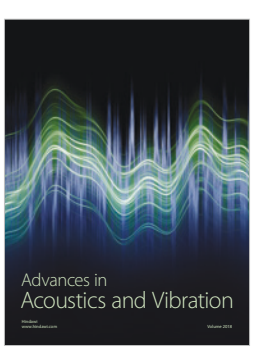












Hindawi

Submit your manuscripts at
www.hindawi.com