Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 16, no. 7, July 2020, pp. 1058-1066
DOI: 10.23940/ijpe.20.07.p8.10581066

Differential Privacy Spatial Decomposition via Flattening Kd-Tree

Guoqiang Gong, Cedric Lessoy, Chuan Lu*, and Ke Lv

School of Computer and Information, Three Gorges University, Yichang, 443002, China

Abstract

The key problem of using differential privacy is controlling sensitivity. Almost all papers focus on processing sensitivity, but the efficiency of the algorithm is also very important. Therefore, this paper hopes to improve efficiency as much as possible under the premise of ensuring utility. In this paper, decomposition and reconstruction via flattening kd-tree (DRF) is proposed based on differential privacy, which applies a flattening kd-tree to process the adjacency matrix. Firstly, by adjusting the vertex labeling, the set of labeling form dense areas and sparse areas as much as possible in the adjacency matrix. The adjacency matrix is then decomposed by flattening kd-tree, and each sub-region is anonymously operated using differential privacy. Finally, each subregion is reconstructed to obtain a complete anonymous graph. At the end of the article, experiments are conducted over real-world datasets. According to the results, DRF has a significant improvement in efficiency, the time complexity of DRF is $\boldsymbol{O}(|\boldsymbol{V}|)$, and DRF has a good performance in degree distribution, degree centrality and cutting query.

Keywords: differential privacy; kd-tree; privacy preserving

(Submitted on April 20, 2020; Revised on May 19, 2020; Accepted on June 17, 2020)

1. Introduction

With the continuous development of computers and related technologies, graph data has been more and more widely used. It can satisfy many needs of data analysis, but many information disclosure cases in the real-world show that privacy security is a very important issue when graph data is published.

K-anonymity [1-2], l-diversity [3], t-closeness [4] and clustering [5] have their own advantages, but they are poor in resisting background knowledge attacks. Differential privacy [6] has strict mathematical protection level and leakage risk quantification, which is used to defend against background knowledge attacks. However, the direct use often leads to large errors by great sensitivity, so some processing is needed to control sensitivity. Spatial decomposition is a very effective way to control sensitivity. Spatial decomposition is categorized as data-independent and data-dependent types. The data-independent decomposition is a logical dataset partition by using the data structure directly. Xiao [7] proposed the HKD-tree partitioning method, which first uses the grid to divide the data and then separately adds noise, but this method is not suitable for inhomogeneous data. Cormode [8] analyzes various hierarchical trees in detail, and then presents a differential privacy hierarchy model for large interval count queries. Qardaji [9] proposed two methods for partitioning grid structures: UG (Uniform Grid) and AG (Adaptive Grids). Although the error equalization problem can be effectively solved, it is possible that the accuracy of the result is affected by excessive noisy. Zhang [10] proposed PrivTree, which combines the noisy count of leaf nodes with the noisy count of non-leaf nodes to respond to queries. This method solves the problem that the noise of tree structure depends on the height of the tree.

The number of Tree structures is the most in the data-depends, and those partitioning methods are affected by data distribution. Inan A.[11] proposed kd-SM based on kd-tree in 2010. The main idea is to calculate the kd-tree median value using the median noise. Comode [8] used the exponential mechanism to calculate when the kd-tree was used. [12] used H-tree to divide data and allocate less privacy budget to noise count. On the contrary, the median is given more budget, but this method is not suitable for big datasets. Efficiency and utility are both very important thing when differential privacy is used to network data, but most methods rarely consider efficiency. Therefore, this paper hopes to improve efficiency as much as

* Corresponding author.
E-mail address: 621030101@qq.com

possible under the premise of ensuring utility. Chen [13] proposed an exploration and reconstruction algorithm based on density where the adjacency matrix of the graph is divided into multiple disjoint sub-regions. Then, noise is added, and finally the exponential mechanism is used for reconstruction. However, this method has a high runtime cost in data division. Sampling can reduce a certain extent cost, but, at the same time, it would also affect the accuracy of Algorithm. Therefore, this paper proposes a method for flattening kd-tree, which can greatly improve efficiency while maintaining utility.

2. Preliminaries

Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two adjacent datasets, that is, $\mathcal{D}_1$ and $\mathcal{D}_2$ are different only in one record, and are written as $\|\mathcal{D}_1 - \mathcal{D}_2\| = 1$. Then, two techniques are used to implement ϵ-differential privacy: the Laplace mechanism [14] and the Exponential mechanism [15]. it has the following definition.

Definition 1 (ϵ-Differential Privacy). There is a random algorithm $\mathcal{A}$ that satisfies any possible output $\mathrm{O} \in \mathrm{Range}(\mathcal{A})$ for any two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ (i.e. $\|\mathcal{D}_1 - \mathcal{D}_2\| = 1$), $\Pr[\mathcal{A}(\mathcal{D}_1) \in \mathrm{O}] \leq e^{\epsilon} \times \Pr[\mathcal{A}(\mathcal{D}_2) \in \mathrm{O}]$, Then, the random algorithm $\mathcal{A}$ satisfies the ϵ-differential privacy.

Definition 2 (Global Sensitivity). For any two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, the global sensitivity of the function $f: \mathrm{D} \to \mathbb{R}^d$ is $\mathrm{GS}(f) = \max_{\mathcal{D}_1, \mathcal{D}_2} \|f(\mathcal{D}_1) - f(\mathcal{D}_2)\|$.

Definition 3 (Laplace mechanism). For any function $\mathrm{f}: \mathrm{D} \to \mathbb{R}^d$, mechanism $\mathcal{A}$: $\mathcal{A}(D) = f(D) + Laplace(GS(f)/\epsilon)$.

Definition 4 (Exponential mechanism). Given a utility function $\mathrm{q}: (D \times \mathcal{R}) \to \mathbb{R}$, mechanism $\mathcal{A}$: $\mathcal{A}(D, q) = \left\{return\ r\ with\ probability\ \propto exp\left(\frac{\epsilon q(D,r)}{2GS(q)}\right)\right\}$.

In some different cases, the privacy budget ϵ will also have different calculation schemes. According to paper [16], there are the following two theorems.

Theorem 1. A series of mechanisms $\mathcal{A}_i$ provide ϵ_i-differential privacy, respectively, then the mechanism $\mathcal{A}_i(D)$ on dataset D satisfies $\sum \epsilon_i$-differential privacy.

Theorem 2. If a series of datasets D_i do not intersect, a series of mechanisms $\mathcal{A}_i$ provide ϵ_i-differential privacy, respectively, to satisfy $\max(\epsilon_i)$-differential privacy.

To facilitate the calculation of the regional density, the counting matrix [13] is introduced.

Definition 5 (Counter Summary Matrix). Given the adjacency matrix A of the simple graph $\mathrm{G} = (V, E)$, the count summary matrix C of A is a matrix of $|V| \times |V|$, where $\forall 1 \leq \mathrm{i}, \mathrm{j} \leq |V|$, $\mathrm{C}[I, J]$ is equal to the number of 1 in the region $\mathrm{A}[1, i; 1, j]$, that is $\mathrm{C}[i, j] = \sum_{m=1}^{i} \sum_{l=1}^{j} A_{ml} = C[i-1, j-1] + \mathrm{C}[i, j-1] - \mathrm{C}[i-1, j-1] + A_{ij}$. If $\mathrm{i} < 1$ or $\mathrm{j} < 1$, then $\mathrm{C}[i, j] = 0$.

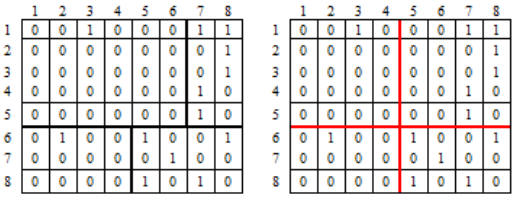
The density of $\mathrm{A}[k, l; m, n]$ is

$$\mathrm{den}(A) = \{C[l, n] - \mathrm{C}[l, m-1] - \mathrm{C}[k-1, n] + \mathrm{C}[k-1, m-1]\}/(n-m+1)(l-k+1)$$

Spatial decomposition is the division of data into smaller areas or fewer points, it can be divided into two categories: data-independent and data-dependent. Cormode [8] proposed a flattening kd-tree in order to better compare the utility of quadtree and kd-tree (shown in Figure 1), flattening kd-tree decomposition is divided into two steps: the adjacency matrix is divided into two parts in the horizontal direction; then the two parts are vertically divided to obtain four areas, as shown in Figure 1. Each split of the flattening kd-tree is constructed using two kd-tree segmentation methods. In the lower right area of the two figures, the effect of the apparently flattening kd-tree is better than the quadtree.

3. Decomposition and Reconstruction

Based on the analysis in Section III, this paper proposes DRF. The whole process can be divided into four steps.



(a) Flattening kd-tree (b) Quadtree

Figure 1. Flattening kd-tree and quadtree

Step1. Adjusting Vertex Labeling. In order to reduce the error of decomposition and reconstruction, the data should form dense regions and sparse regions in the adjacency matrix. This step adjusts the vertex labeling to form dense and sparse areas in the adjacency matrix. The detailed process is shown in algorithm 1.

Algorithm 1 Adjusting_vertex_labeling

Input: The original graph G
Output: Adjacency matrix A
1. Generate a random vertex labeling $\mathcal{L}$
2. i = 0;
3. **while** i < t:
4. Generate a candidate set $\mathcal{N}$ of swaps;
5. **for** (v_m, v_n) in $\mathcal{N}$:
6. **if** $(degree_m - degree_n) \cdot \left(\left|v_m - \left\lceil\frac{|V|}{2}\right\rceil\right| - \left|v_n - \left\lceil\frac{|V|}{2}\right\rceil\right|\right) > 0$:
7. exchange labeling(v_m, v_n);
8. remove (v_m, v_n) in $\mathcal{N}$;
9. i++

Return $\bar{\mathcal{L}}$

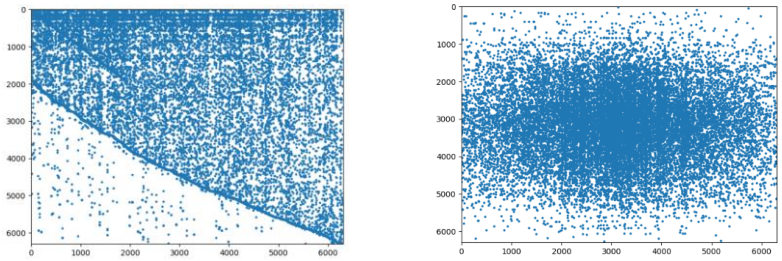
Definition 6 For the graph $G = (V, E)$, the corresponding center point in the adjacency matrix is $\left(\left\lceil\frac{|V|}{2}\right\rceil, \left\lceil\frac{|V|}{2}\right\rceil\right)$, and a set of labeling is measured by the following formula:

$$q(\mathcal{L}) = \sum_{i,j}^{|V|} A_{ij} \cdot \left(\left|i - \left\lceil\frac{|V|}{2}\right\rceil\right| + \left|j - \left\lceil\frac{|V|}{2}\right\rceil\right|\right) \tag{1}$$

Theorem 3 If (v_m, v_n) vertex pairs are satisfied

$$(degree_m - degree_n) \cdot \left(\left|v_m - \left\lceil\frac{|V|}{2}\right\rceil\right| - \left|v_n - \left\lceil\frac{|V|}{2}\right\rceil\right|\right) > 0 \tag{2}$$

Figure 2 shows the effect of process 1 of the data set p2p-Gnutella08 (described in section 4 about p2p-Gnutella08). It can be seen that after processing, it is as close as possible to the center point in the adjacency matrix. In each iteration of the whole process, each point involves only one exchange, and each iteration only needs to exchange $\left\lceil\frac{|V|}{2}\right\rceil$ times. So, the time complexity is $O(|V|)$.



(a) Before adjustment (b) After adjustment

Figure 2. Effect of Algorithm 1

Step2. Decomposition. The privacy budget ϵ_D is divided into two parts ϵ_D^d and ϵ_D^c (line 2). ϵ_D^d is the privacy budget for finding the decomposition point, and ϵ_D^c is the budget for calculating the noise count. For the split point budget

ϵ_D^d, flattening kd-tree decomposition is divided into two steps (shown in algorithm 2). When the maximum height of $\mathcal{KT}$ is h_{max}, for each split, the split point budget is $\frac{\epsilon_D^d}{2h_{max}}$. For the noise count budget ϵ_D^c, because of the nature and form of the flattening kd-tree, a standard quadtree geometry budget scheme can be used [8]. In a quadtree with a maximum height of h_{max}, the node with depth i has a privacy budget $2^{i/3}\left(\sqrt[3]{2}-1\right)\epsilon_D^c/(2^{\frac{(h_{max}+1)}{3}}-1)$.

Then, calculate the maximum height of the flattening kd-tree (line 4). For a standard quadtree, the area of the leaf area is $\frac{|V|^2}{4^{h_{max}}}$. Based on the error considerations, the size of the leaf area is greater than μ times the noise standard deviation, so

$$\frac{|V|^2}{4^{h_{max}}} \geq \frac{\mu\sqrt{2}GS(f)}{\epsilon} = \frac{\mu\sqrt{2}GS(f) * \left(2^{(h_{max}+1)/3}-1\right)}{2^{h_{max}/3}\left(\sqrt[3]{2}-1\right)\epsilon_D^c} \Rightarrow \quad \sqrt[3]{2}(2^h)^2 - (2^h)^{5/3} \leq \frac{\left(\sqrt[3]{2}-1\right)|V|^2\epsilon_D^c}{\mu\sqrt{2}GS(f)} \tag{3}$$

Where $GS(f)$ is the sensitivity of the noise count and $GS(f)=1$. If the graph $\mathrm{G}=(V,E)$ and the noise count budget ϵ_D^c are given, and h_{max} is an integer, h_{max} can be calculated by the inequality. For the decomposition of lines 9 to 15, in order to separate dense areas and sparse areas, region density is introduced.

Algorithm 2 Decomposition

Input: Adjacency matrix A
Input: Privacy budget ϵ_D
Output: Noisy flattening kd-tree $\mathcal{KT}$

```
1.  i = 0;
2.  ε_D = ε_D^d + ε_D^c
3.  KT → ∅;
4.  Calculate the max height h_max of KT;
5.  while i < h_max:
6.      if i = 0:
7.          Root node representing A in KT;
8.      while node v ∉ leaf:
9.          Generate candidate regions for split points
10.         Finding the split point for the first time;
11.         Decomposition (v, ε_D^d) → two subregions R_1, R_2;
12.         for each R ∈ R_1, R_2:
13.             Generate candidate regions for split points
14.             Finding the split point for the second time;
15.             Decomposition (R, ε_D^d) → two subregions R'_1, R'_2;
16.             for each R'_1, R'_2:
17.                 Calculate NoisyCount(R', ε_D^c);
18.                 a node u representing R' in KT;
19.                 if u = stop conditon:
20.                     u is a leaf
21.  i++
Return KT
```

Definition 5 (Region density): For the region $\mathrm{R} \subseteq \mathrm{A}(G)$, the size is $|R| = m \times l$, then its region density is defined as $\mathrm{den(R)} = \sum_{i=1}^{m}\sum_{j=1}^{l} A_{ij}/\mathrm{ml}$. So, the utility function in exponential mechanism looks like this:

$$q(R,p) = \max_{\forall R' \in R}\left(den(R')\right) - \min_{\forall R' \in R}\left(den(R')\right) \tag{4}$$

Where R' is a sub-area obtained from the region R according to the division point p. By using this formula, two sub-regions with the largest difference in density can be obtained, that is, dense regions and sparse regions can be obtained as much as possible.

Assuming that the area of the sub-region R' is S', the sensitivity is $\mathrm{GS}(q) = \frac{1}{S'}$. So, limiting the area of the sub-area can reach a lower sensitivity. Therefore, the area of the leaf region R' is limited to 1/4 of the original region R area S (the 9th row), and the sensitivity becomes $\mathrm{GS}(q) = \frac{4}{S}$. Therefore, for each decomposition, the split point is limited to the $\frac{1}{4} \sim \frac{3}{4}$ region of the edge, and the area of the leaf region for the original region is at least $\frac{S}{16}$. Then, select the split point according to the following formula:

$$exp\left(\frac{\epsilon_D^d}{2\times 2h_{max}GS(q)}q(R,p_i)\right)\Big/\sum_{p_j\in\mathcal{P}}exp\left(\frac{\epsilon_D^d}{2\times 2h_{max}GS(q)}q(R,p_j)\right) \tag{5}$$

The purpose of the decomposition is to get the dense and sparse areas as much as possible, so three stop conditions (line 20) are set in the Decomposition section: Dense area, Sparse zone, Reach the maximum height h_{max}.

For the first condition, the area density is used to determine whether a region forms a dense region. If the area density $\mathrm{den}(R) \geq 80\%$ of an area, it is not necessary to decompose again. For the determination of the sparse zone, the noise count is used to judge. According to [13], the threshold is set to 80% of the minimum area $\frac{|V|^2}{4^{h_{max}+1}}$ of the leaf area, that is, the noise count $\leq 80\% \times \frac{|V|^2}{4^{h_{max}+1}}$ is determined to be a sparse area. For the third condition, the maximum height of the flattening kd-tree has been calculated previously, so the maximum height h_{max} is reached and the decomposition is stopped.

It can be known from the analysis of the flattening kd-tree in Section II that the time complexity of the segmentation process is $\mathrm{O}(|V|)$.

Step3. Post Processing. In the previous step, due to the setting of the stop condition, some areas would stop decomposing in advance. So, the privacy budget for noise counting is not utilized. In this step, these problems would be solved. According to Theorem 1 and Theorem 2, the raw count represented by each child node is only related to the noise count of its parent and ancestor, and is independent of the rest of the nodes. Therefore, the noise count budget of the leaf node with depth $i < h_{max}$ can be adjusted to

$$\sum_{i}^{h_{max}} (2^{i/3}(\sqrt[3]{2}-1)\epsilon_D^c/(2^{(h_{max}+1)/3}-1)) \tag{6}$$

and then its noise count is recalculated, while the total noise count budget ϵ_D^c is unchanged.

Step4. Reconstruction. In this step, the anonymized matrix $\tilde{A}$ would be obtained. For each leaf region, the reconstruction process is divided into two steps: the first step is to select the correct number that elements 1 in the reconstructed adjacency matrix $\tilde{A}$ that matches the element 1 in the original adjacency matrix A; The second step is to specifically assign the position of the element 1 in the adjacency matrix $\tilde{A}$ according to the score of the first step. In the first step, each region has a correct number of elements 1 in the reconstructed adjacency matrix $\tilde{A}$. For each region, the score is the matched number between the reconstructed adjacency matrix $\tilde{A}$ with the original adjacency matrix A. Let the leaf area raw count be c, the noise count be $\tilde{c}$, then the score $\mathrm{q}(i)\epsilon[0, min(c,\tilde{c})]$, and the sensitivity is $GS(q)=1$. Then, the score is chosen by the exponential mechanism $exp\left(\frac{q(i)\epsilon_R}{2GS(q)}\right)/\sum_{j=0}^{min(c,\tilde{c})} exp\left(\frac{q(j)\epsilon_R}{2GS(q)}\right)$.

For each region, the split point considered is only $min(c,\tilde{c})+1$. In the second step, for each region, according to the score q in the first step, the positions of number of q are randomly selected in the position of the element 1 in the corresponding original matrix A, and the corresponding position in the corresponding anonymous adjacency matrix $\tilde{A}$ is set to 1. The remaining $\tilde{c}-q$ are randomly selected from the 0 elements on the non-main diagonal in the corresponding region, which is set to 1 in the anonymous adjacency matrix $\tilde{A}$.

DRF consists of four parts. In first step, no matter how the vertex label is adjusted, the characteristic and structure of the graph will not change. Therefore, no privacy budget is allocated in this part, and the time complexity is $\mathrm{O}(|V|)$. For the Decomposition, the privacy budget ϵ_D is divided into two parts ϵ_D^d and ϵ_D^c. ϵ_D^d is the privacy budget for finding the decomposition point, and ϵ_D^c is the budget for calculating the noise count. Because each sub-region does not intersect, according to Theorem 1 and Theorem 2, it satisfies:

$$\epsilon_D = \epsilon_D^d + \epsilon_D^c = \sum_{i=0}^{h_{max}} \frac{\epsilon_D^d}{2h_{max}} + \sum_{i=0}^{h_{max}} \frac{2^{i/3}(\sqrt[3]{2}-1)\epsilon_D^c}{2^{(h_{max}+1)/3}-1} \tag{7}$$

In this part, due to the characteristic of the flattening kd-tree, the time complexity is only $\mathrm{O}(|V|)$. The third part is based on the noise count after differential privacy, so there is no need to allocate a privacy budget and it will not affect the overall

result. The final step is Reconstruction. The whole process is based on an exponential mechanism with a privacy budget of ϵ_R. Therefore, the DRF algorithm satisfies the differential privacy of the privacy budget $\epsilon = \epsilon_D + \epsilon_R = \epsilon_D^d + \epsilon_D^c + \epsilon_R$. For the whole experiment, the distribution ratio of the privacy budget is $\epsilon_D^d : \epsilon_D^c : \epsilon_R = 3:7:3$.

4. Experiment

In this section, DRF would be tested through a series of experiments. For reference, DRF will be compared with the DER algorithm of Chen [13], kd-h (kd-hybrid) of Cormode [8], Random in Degree Distribution, Degree Centrality and Cut Query. Random is, in the fourth step, without the exponential mechanism. 1 element is randomly arranged in the adjacency matrix. This experiment was done in Python on a 16GB RAM Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz, and each data is averaged over multiple runs.

Table 1 shows the time of processing four data sets using DRF at $\epsilon = 1$. Figures 3 and 4 show the time comparison of DER and KD-H with DRF. The ordinates of the two figures are $T(DER)/T(DRF)$ and $T(kd-h)/T(DRF)$ respectively. It can be seen that DRF is very efficient, and it has good performances in the experiments. The time taken by kd-h is similar to DRF, and DER takes more time. Compared with DER, when the data is small or the step is large enough, the time spent is similar to DRF. However, as the amount of data increases, the increment of time spent is greater than the increment of the amount of data. For example, in the dataset CollegeMsg, the data amount of DER with step = 10 is twice that of step = 20, and the time is 4.72 times. This is very disadvantageous when DER deals with big data.

Table 1. Datasets and Runtimes of DRF

Datasets	Nodes	Edges	Time(s)
CollegeMsg	1899	20296	5.4165
OpenFlights	2939	30501	14.2045
p2p-Gnutella04	10876	39994	195.8137
p2p-Gnutella08	6301	20777	61.53489826

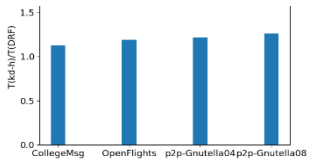
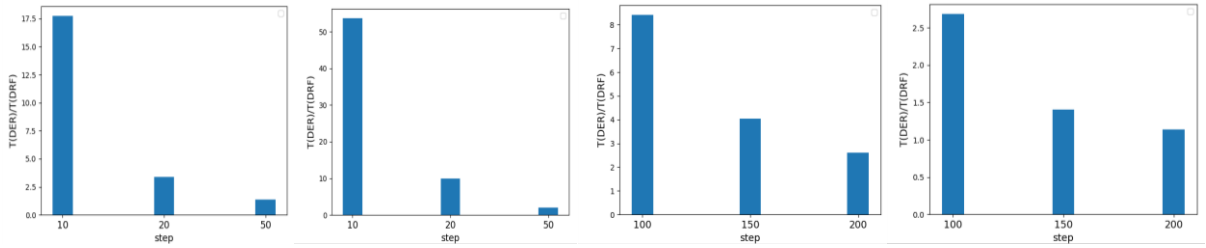


Figure 3. Efficiency comparison $T(kd-h)/T(DRF)$



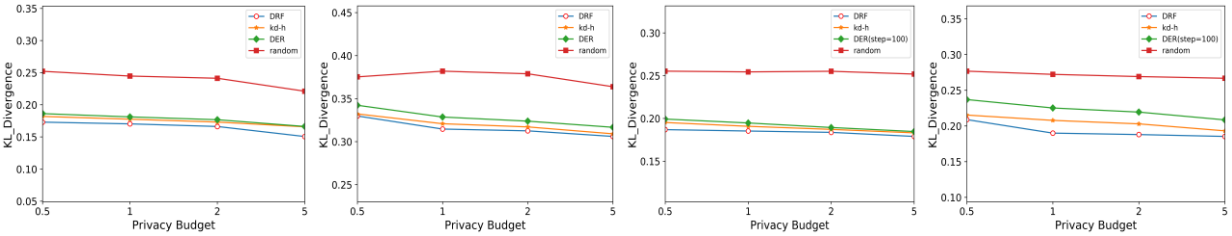
(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 4. Efficiency comparison $T(DER)/T(DRF)$

The KL-divergence of DRF, kd-h, DER and Random under different privacy budgets is given in Figure 5. It can be seen that DRF has the best effect under different privacy budgets. Kd-h and DER are worse than DRF under various privacy budgets. Random has the worst performance because the fourth step is random. Although the difference between DER and DRF is not too much, DER takes much more time than DRF. If DER wants to improve efficiency, it will increase the step size. In Figure 6, DRF and kd-h are compared with DER for different step in KL-divergence. As the step size increases, the error of DER also increases. According to Figure 5(c), (d) and Figure 6(c), (d), in the case of large data, if DER wants to improve efficiency, it will use a large step size, and the utility will be poor.

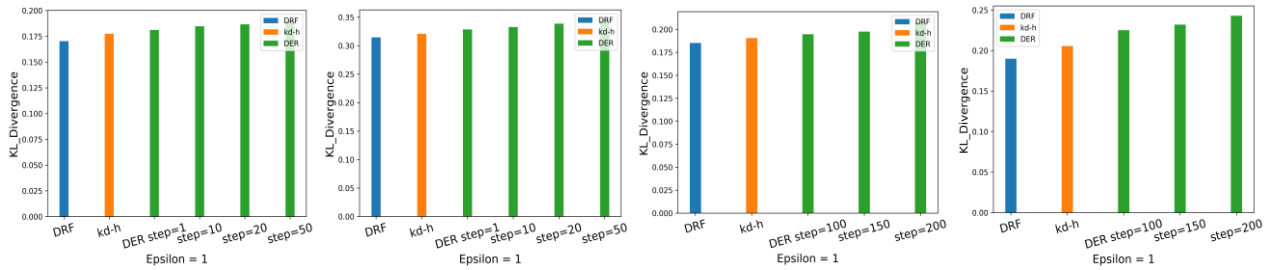
Figures 7 and 8 give various results of RRDC. DRF still has the best performance in different datasets, while DER does not perform well in RRDC. And if large steps are used to process large data, DER has the same problem that the error will increase further. Unexpectedly, Random performs well in DDRC and is similar to DRF. The reason is that RRDC is mainly affected by the second step Decomposition and the third step Post_Processing. Therefore, in ε = 1, DRF, kd-h, Random, and DRF-2 are given in Figure 9 (in the third step Post_Processing, DRF-2 only processes the noise count to the [0, S]). It can be

seen that DRF-2 performs significantly worse than DRF and RANDOM without the third step of improving accuracy. Therefore, when comparing DRF, kd-h, Random and DRF-2, the second step Decomposition of DRF and the third step Post_Processing are helpful for improving utility.



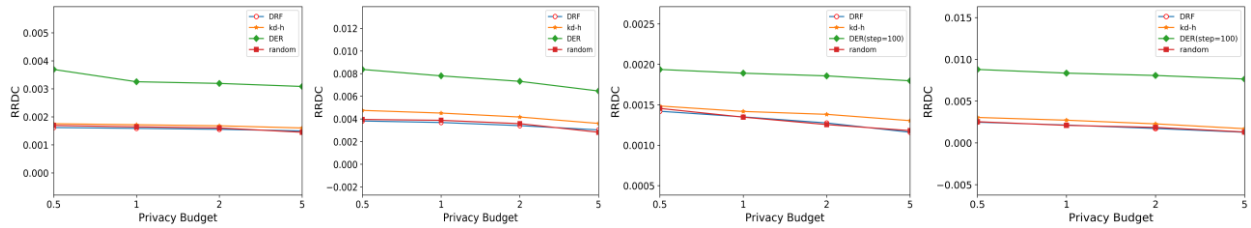
(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 5. KL-divergence vs. Privacy budget



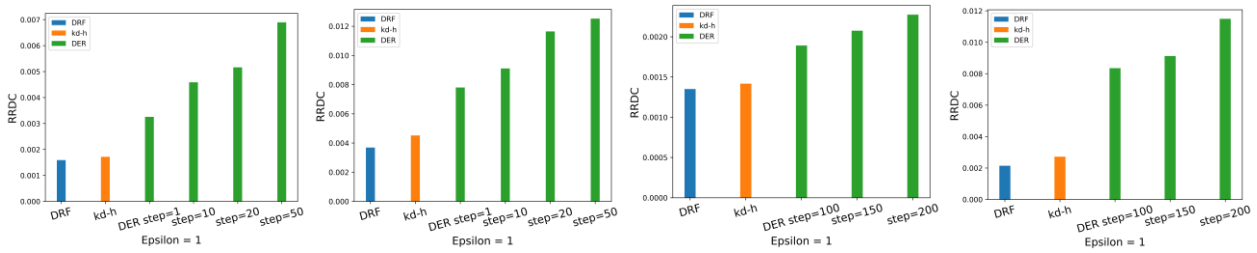
(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 6. KL-divergence vs. Step



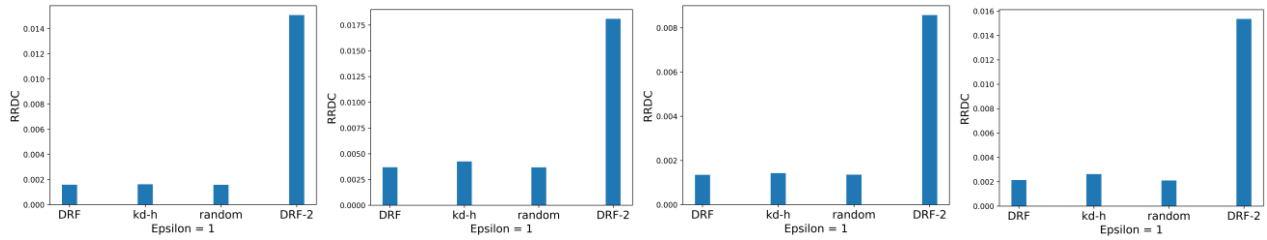
(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 7. RRDC vs. Privacy budget



(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 8. RRDC vs. Step



(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 9. DRF, Random, DRF-2 with RRDC

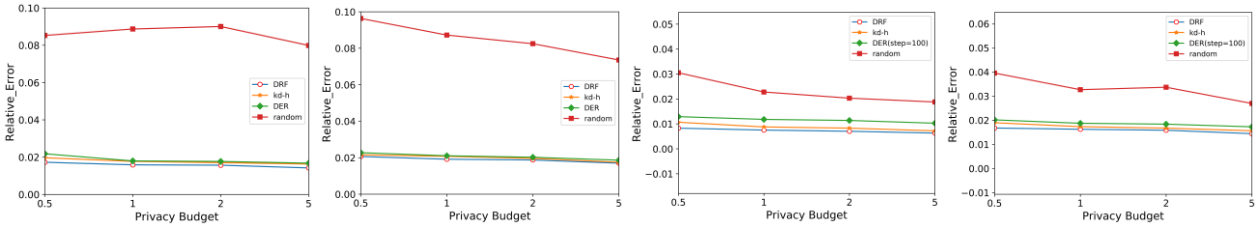
For the cutting query, Table 2 gives the performance with $\epsilon = 1$ for the four data sets under the query size of $0.2|V|$,

$0.4|V|, 0.6|V|, 0.8|V|$. Then, it compares the performance of DRF, kd-h, DER and RANDOM with $\epsilon = 1$ and query size of $0.2|V|$ in Figure 10. As can be seen from Table III and Figure 10, DRF and kd-h perform well under various query sizes. Random performed poorly. DER performs well when the step is small, but in Figure 11, DER will have the same problems as KL-divergence and DDRC when it deals with large data sets with large steps.

As can be seen from the whole experiment, DRF performs best both in terms of utility and efficiency. Kd-h is slightly worse than DRF in performance. At step = 1, the performance of the DER is comparable to that of the DRF, but it takes several times longer than the DRF. When improving efficiency, the performance of the DER will decrease as the step size increases. Therefore, DER is very disadvantageous when it deals with large data volumes. In the experiment, the performance of DRF, kd-h, Random and DRF-2 in RRDC were also compared. It can be seen that the third step Post Processing and the fourth step Reconstruction effectively improve the accuracy. Therefore, DRF can not only ensure the utility but also maintain high efficiency when dealing with large data volume or small data volume.

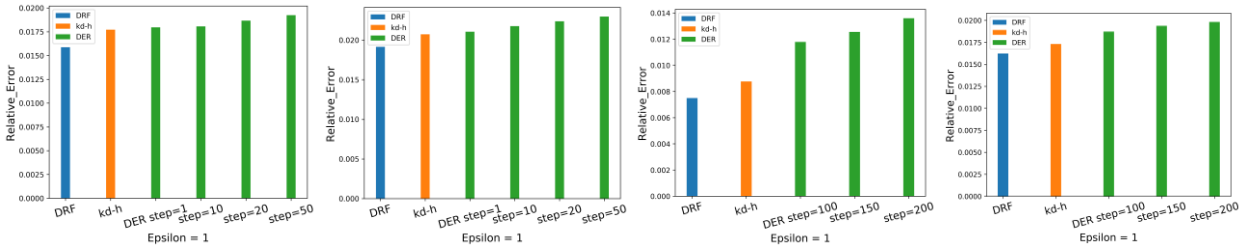
Table 2. Error $\left(Q_{S,T}(\tilde{G})\right)$ in cut query

Datasets	$0.2\|V\|$	$0.4\|V\|$	$0.6\|V\|$	$0.8\|V\|$
CollegeMsg	0.01587	0.00947	0.00684	0.00554
OpenFlights	0.01914	0.01888	0.01449	0.01035
p2p-Gnutella04	0.00751	0.00658	0.00632	0.00589
p2p-Gnutella08	0.01623	0.01466	0.01352	0.01158



(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 10. Cut query vs. Privacy budget



(a) CollegeMsg (b) OpenFlights (c) p2p-Gnutella04 (d) p2p-Gnutella08

Figure 11. Cut query vs. Step

5. Conclusion

A scheme for publishing anonymous graphs based on differential privacy is given in this article. In the first step, the adjacency matrix form is adjusted to dense and sparse areas as much as possible, which is beneficial for later partitioning and reconstruction. Second, a flattening kd-tree was introduced. When decomposing the adjacency matrix, the flattening kd-tree can effectively decompose the adjacency matrix. So, DRF is very advantageous when the amount of data is large. Then, there is post-processing. In this step, privacy budget is fully utilized. Finally, the adjacency matrix after anonymity is obtained by the adjusted flattening kd-tree $\widetilde{\mathcal{KT}}$. Experiments show that DRF has a good performance in terms of degree distribution, degree of centrality and cutting query. At the same time, this paper also made a certain comparison in terms of efficiency. DRF has a great advantage in efficiency. In experiments with a large amount of data processing, DRF can not only ensure utility, but also have processing time advantages.

Acknowledgements

This work is supported by The National Key Research and Development Program of China (2016YFB0800403). Thanks to the data sets provided by SNAP (Stanford Network Analysis Project) and KONECT (The Koblenz Network Collection).

References

1. J. J. Jia and G. L. Yan, "A Personalized(p,k)-Anonymity Privacy Protection Algorithm," *Computer Engineering*, Vol. 44, No. 1, pp. 176-181, 2018
2. A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, and J. Parra-Arnau, "Does k-Anonymous Microaggregation Affect Machine-Learned Macrotrends?" *Access IEEE*, Vol. 6, pp. 28258-28277, 2018
3. K. Oishi, Y. Tahara, Y. Sei, and A. Ohsuga, "Proposal of l-Diversity Algorithm Considering Distance Between Sensitive Attribute Values," in *Proceedings of IEEE Symposium Series on Computational Intelligence* (*SSCI*), 2017
4. Z. Tu, K. Zhao, F. L. Xu, Y. Li, L. Su, and D. P. Jin, "Protecting Trajectory From Semantic Attack Considering k-Anonymity, l-Diversity, and t-Closeness," *IEEE Transactions on Network and Service Management*, Vol. 16, No. 1, March 2019
5. D. Huang, C. D. Wang, and J. H. Lai, "Locally Weighted Ensemble Clustering," *IEEE Transactions on Cybernetics*, Vol. 51, pp. 1-14, 2017
6. C. Dwork, "Differential Privacy," in *Proceedings of the 33rd International Conference on Automata, Languages and Programming*, Vol. Part II, Springer, Berlin, Heidelberg, 2006
7. Y. H. Xiao, L. Xiong, and C. Yuan, "Differentially Private Data Release Through Multidimensional Partitioning," in *Proceedings of VLDB Conference on Secure Data Management*, Springer-Verlag, 2010
8. G. Cormode, M. Procopiuc, E. Shen, D. Srivastava, and T. Yu, "Differentially Private Spatial Decompositions," in *Proceedings of the 27th IEEE International Conference on Data Engineering* (*ICDE*), pp. 20-31, 2012
9. W. Qardaji, W. Yang, and N. Li, "Differentially Private Grids for Geospatial Data," in *Proceedings of 2013 IEEE 29th International Conference on Data Engineering* (*ICDE*), 2013
10. J. Zhang, X. Xiao, and X. Xie, "Privtree: A Differentially Private Algorithm for Hierarchical Decompositions," in *Proceedings of the 2016 International Conference on Management of Data*, pp. 155-170, 2016
11. A. Inan, M. Kantarcioglu, G. Ghinita, and E. Bertino, "Private Record Matching using Differential Privacy," in *Proceedings of International Conference on Extending Database Technology*, ACM, 2010
12. H. To, L. Fan, and C. Shahabi, "Differentially Private h-Tree," in *Proceedings of the 2nd Workshop on Privacy in Geographic Information Collection and Analysis*, pp. 1-8, 2015
13. R. Chen, B. C. M. Fung, P. S. Yu, and B. C. Desai, "Correlated Network Data Publication via Differential Privacy," *The VLDB Journal*, Vol. 23, No. 4, pp. 653-676, 2014
14. C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Proceedings of the 3rd Theory of Cryptography Conference* (*TCC*), pp. 265-284, 2006
15. F. McSherry and K. Talwar, "Mechanism Design via Differential Privacy," in *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science* (*FOCS*), pp. 94-103, 2007
16. F. McSherry, "Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis," in *Proceedings of the 35th ACM SIGMOD International Conference on Management of Data* (*SIGMOD*), pp. 19-30, 2009

Guoqiang Gong is an Associate professor at the China Three Gorges University. His research interests include privacy preservation, advanced signal processing.

Cedric Lessoy is a Master's candidate at the China Three Gorges University. His research interests include deep learning and social network data analysis.

Chuan Lu is a Master's candidate at the China Three Gorges University. His research interests include social network and privacy protection.

Ke LV is a Professor at the China Three Gorges University. His research interests include information processing and big data analysis.