Improving Accuracy of Differentially Private Kronecker Social Networks via Graph Clustering

Arinjita Paul*, Vorapong Suppakitpaisarn†, Mitali Bafna ‡, C. Pandu Rangan*
* IIT Madras, Email: arinjita@cse.iitm.ac.in, prangan55@gmail.com
†The University of Tokyo, Email: vorapong@is.s.u-tokyo.ac.jp
‡Harvard University, Email: mitalibafna@g.harvard.edu

Abstract—**Using graph clustering, we improve accuracy of Kronecker social networks which are protected by differential privacy. Ensuring the differential privacy implicates addition of marginal changes to the network and publishing the modified network data. In many cases, it induces a large gap between the original network and the modified graph statistics, such that very little useful information can be inferred from the published graph. We use the fact that network structures in all graph clusters are similar, to improve the utility of the publication methods based on Kronecker graphs. Instead of anonymizing the social network as a whole, we anonymize each cluster of the network separately, and combine the sanitized results thereafter. We justify why this idea provides an anonymized social network with high utility and also prove that our output social network ensures rigorous differential privacy guarantees. Our experimental results show that our mechanism exhibits good agreement of the structural properties with the real graphs, and outperforms the existing anonymization techniques for certain utility measures.**

Index Terms—**anonymization, differential privacy, social network privacy, clustering**

I. Introduction

Social network datasets are complex graphs representing entities and connections between them. The entities correspond to individuals connected by edges which depict relationships such as the *friendship network* describing personal relationships, *communication network* that illustrates the interactions among the employees in an organisation and *citation network* where authors are linked by co-citation relationships. In recent years, social networks have received dramatic attention in research and development [1].

As more and more social networks are being published for knowledge discovery, privacy of individuals has become a serious concern. Naive anonymization techniques of releasing the data by just removing identifiers of individuals is not enough to preserve privacy. Several works such as [2] give a possible attack using which an adversary can learn the structure of a subgraph of the anonymized network. By knowing a subgraph, attackers can guess with large probability if two particular nodes have a link between them. They can further have users' friend list, which is an information that most social network users do not wish to reveal.

Differential privacy [3] is a notation that measures if a particular information is protected from attackers. It is one of the most commonly-used notations as it provides rigorous guarantees of privacy while allowing diverse range of queries to be answered approximately. There are three main differential privacy models for social networks: node privacy, edge privacy, and weighted edge privacy [4]–[6]. In this work, we consider the edge privacy model. In this model, two graphs that differ by just one edge are called "neighboring" graphs. We modify our original graph using some probabilistic methods, and ensure that the outputs of the methods from two neighboring graphs are not distinguishable. From the outputs, attackers then cannot distinguish between the original graph with a link e and the original graph with the link e being removed. By that, they cannot guess if the link e does exist in the original graph. The information that two persons are friends in the social network is not revealed to them.

There are several probabilistic methods proposed for the edge privacy model. Most of these methods aim at releasing a particular information of social networks. Those include the work by Hay et al. [5] which aim to output degree distribution of a graph, the work by Kenthapadi et al. [7] which aim to output a graph which shortest path lengths of two particular nodes are preserved, and the work by Ahmed et al. [8] which aim to output a graph of which adjacency matrix is has the similar eigen vectors to that of the original graph. On the other hand, several methods are proposed to output a graph that can give several graph information. Those include the work by Nguyen et al. [9] and Xiao et al. [10]. Although the general-purpose graph outputs are more desirable in practice, as far as we know, we cannot get accurate information from the outputs of most existing methods. It is therefore important to improve their accuracy.

Many general-purpose methods, including those recently proposed, are based on Kronecker graphs [11]–[14]. Due to their rich mathematical structure, Kronecker graphs have been well-studied in the literature. Leskovec et al. [15] showed a stochastic variant of Kronecker graphs which can model real world networks. Subsequently, it was shown in [15]–[17] that these graphs possess many important properties of social networks like power law degree distribution, betweenness centrality distribution, community structure, and so on. Since these graphs model real world networks so well, we use the stochastic Kronecker graph model to anonymize our social network and give rigorous privacy guarantees in the edge-privacy model for differential privacy.

978-1-7281-5628-6/20/$31.00 ©2020 IEEE

A. Our Contribution

In this work, we aim to improve the accuracy of results obtained from previous kronecker graph generation algorithms.

> We propose a pre and post-processing technique based on graph clusterings that generates Kronecker graphs that are closer to the original social network.

Typically, most social networks have a few clusters. In this work, we propose to provide each of these clusters as inputs to the differential privacy methods. We then obtain respective Kronecker graphs from the methods. Our output is an average of those Kronecker graphs.

We observe that each of the clusters usually have similar network structures and, without considering differential privacy models, the resulting Kronecker graphs are also very similar. However, on applying differential privacy, the resulting published Kronecker graph are altered. Intuitively, we can think that the differential privacy methods add some noise to the Kronecker graphs. Taking an average of a few Kronecker graphs can then reduce the magnitude of noise we add during the privacy process.

On the other hand, in this paper, we can show that taking an average of the outputs does not increase the risk of being attacked. If the method applies to each of the clusters guaranteeing (ε,δ)-differential privacy, publishing the average output can also guarantee the same (ε,δ)-differential privacy.

We show by our experimental results that we can significantly improve accuracy of the social network publications. The distribution of degrees and shortest path lengths in our published graphs is closer to the original graph than that obtained without the pre- and post-processing. Some of our published graphs have even better utility than the networks without differential privacy.

B. Related Works

There are several results that use graph clustering to obtain social network privacy. Before the notion of differential privacy was introduced, methods for privacy notions, such as k-anonymity [18] or ℓ-diversity [19] were used, that involved placing individuals into groups, then replacing information of each individual with information of the group they are assigned to. By that, it is natural to put individuals with similar information to the same group. Because we can have individuals with similar friends by clustering algorithms, there are many results using clusters to improve the accuracy of the published information [20]–[23]. On the other hand, it is not natural to use graph clusters for differential privacy notions. Most of methods for such privacy notions usually focus on how to add minimal noise to the original network, and it is not trivial to find relationships between the noise and clusters of graphs. Also, it is required to prove mathematically that such a clustering method produces a graph that satisfies the notions, but it is not easy to link graph clusters with the proof.

To the best of our knowledge, all existing results for differential privacy notions mentioning graph clusters (such as [8]) do not aim to use graph clusters as a tool to achieve better accuracy. They aim to have a resulting social network which has similar graph clusters to the original graph. In other words, we are the first to use graph clustering as a tool and not as an expecting output.

Some of our ideas are adopted from the work by Smith [24]. In their work, they also propose to divide information to several pieces. However, they do not mention a way to divide the information nor provide experimental results. We found that dividing an information in an arbitrary way does not improve an accuracy of our publication. We then significantly modify the algorithm. As a result, we also need to provide a new differential privacy guarantee proof for the new algorithm.

II. Preliminaries

In this section, we give required notations and background for this paper. Throughout this paper, we denote a set of possible graphs by $\mathscr{G}$ and a set of possible outputs from our algorithm by Y.

A. Differential Privacy

Differential privacy is a well-known and rigorous way to quantify how much private a data publication is. A formal definition of differential privacy under edge privacy model is given as follows:

Definition 1 (Differential Privacy under Edge Privacy Model [5]). *Let $\mathscr{A} : \mathscr{G} \to Y$ be a randomized algorithm that takes as input a graph $G \in \mathscr{G}$ and outputs values in the range Y, where $\mathscr{G}$ is the space of all graphs on n nodes. $\mathscr{A}$ is (ε,δ)-differentially private on $\mathscr{G}$ if for all graphs $G, G' \in \mathscr{G}$ which differ in one edge and for all subsets $S \subseteq Y$, we have,*

$$\mathbb{P}[\mathscr{A}(G) \in S] \leq \exp(\varepsilon)\mathbb{P}\left[\mathscr{A}(G') \in S\right] + \delta,$$

$$\mathbb{P}[\mathscr{A}(G) \in S] \geq \exp(-\varepsilon)\mathbb{P}\left[\mathscr{A}(G') \in S\right] - \delta,$$

where the probability is taken over the randomness of $\mathscr{A}$.

B. Kronecker Graphs [15]

Stochastic Kronecker graphs is a random graph model that obeys common network properties. The model uses a standard matrix operation, called the *Kronecker product*. The Kronecker graph model is based on recursive construction of self-similar graphs, starting with an initiator probability matrix K_1 on N_1 vertices, where each entry in the adjacency matrix is the probability that the edge is present in the graph. Using Kronecker product, larger graphs $K_2, \cdots K_p$ are generated, where K_p is a graph with N_1^p nodes. We define Kronecker product as follows:

Definition 2 (Kronecker product of graphs). *The Kronecker product of two matrices A and B of dimensions $n \times m$ and $n' \times m'$ is given by:*

$$C = A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,m}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}B & a_{n,2}B & \dots & a_{nm}B \end{bmatrix} \quad (1)$$

We let M^p denote the p^{th} Kronecker product of M, that is, M multiplied with itself for p times.

Definition 3 (Stochastic Kronecker graphs)**.** *Let $\mathscr{P}_1$ be a $N_1 \times N_1$ matrix, such that each entry $\theta_{ij} \in \mathscr{P}_1$ belongs to $[0,1]$. Let $\mathscr{P}_p$ be the p^{th} Kronecker power of $\mathscr{P}_1$, i.e. $\mathscr{P}_p = \mathscr{P}_1^p$. Interpret matrix $\mathscr{P}_p$ as the edge probability matrix of a graph with N_1^p nodes, i.e. matrix $\mathscr{P}_p$ defines a distribution on graphs with N_1^p nodes, such that, to sample an instance $K = R(\mathscr{P}_p)$, we include edge (u,v) in K with probability $P_{uv} \in \mathscr{P}_p$.*

The KronFit Algorithm: Leskovec et al [15] gave an efficient algorithm called KronFit which fits a stochastic Kronecker graph model to a given social network. That is, they use maximum likelihood estimation to find the values of an initiator matrix $\mathscr{P}_1$, which maximizes the probability of G begin generated from $\mathscr{P}_1^p$. Since node labellings do not matter, the likelihood of G is the average of the likelihood of G over all permutations of its vertices. We have that, the likelihood of G equal to $l(\theta)$ is,

$$\begin{aligned} l(\theta) = \log \mathbb{P}\left[G \mid \theta\right] &= \log \sum_{\sigma} \mathbb{P}\left[G \mid \theta, \sigma\right] \mathbb{P}\left[\sigma \mid \theta\right] \\ &= \log \sum_{\sigma} \mathbb{P}\left[G \mid \theta, \sigma\right] \mathbb{P}[\sigma], \end{aligned}$$

where σ is a permutation of the vertices of G. Using a gradient descent algorithm they find $\hat{\theta}_{MLE} = \arg\max_{\theta} l(\theta)$.

The speed of KronFit algorithm is significantly improved in the work by Kim and Leskovec [25]. However, as the work provides a similar result as KronFit in term of Kronocker graph results and speed is not the main issue of this work, we choose to use the original KronFit algorithm.

Utility Preservation by Kronecker Graphs: Leskovec et al. [26] show an empirical analysis of how Kronecker graphs exhibit statistical properties of social networks, such as degree distribution, small world phenomenon, and communities.

Mahdian and Xu [27] provide a theoretical study of the various properties exhibited by stochastic Kronecker graph model. A comparison between real-world networks and stochastic Kronecker graphs fitted using a 2×2 matrix $\mathscr{P}_1$ is demonstrated in [15]. The analysis is shown against the following real world networks - the autonomous systems (AS) network obtained from the University of Oregon Route Views project [28], a blog network (Blog-nat06all) and a large online social network (Epinions). They compare the fitted Kronecker graphs to the social networks for global statistics like degree distribution, hop plot and densification power law.

C. Kronecker Graphs with Differential Privacy

Mir et al [11] resort to the "moment matching method" of Gleich *et al* [29] towards constructing a private estimator of Kronecker parameters. Their algorithm matches four statistics of the real graph data to their expected values over the probability distribution of the networks defined by these parameters. The four parameters include computing: number of edges (E), number of triangles (Δ), number of hairpins (2-stars) and number of tripins (3-stars). They compute differentially private estimations of these four statistics. Further, they adapt the Kronecker moment estimation in [29] with the differentially private statistics to compute the private Kronecker matrix.

III. Our Algorithm

Our algorithm is formally described in Algorithm 1. We first divide our social network G into k partitions using graph partition algorithms. As we expect all clusters to have similar size, we believe that techniques based on spectral clustering such as [30] are the most appropriate methods.

Algorithm 1 Pre- and post-processing method to improve differential privacy algorithm based on Kronecker graphs

```
Input: Social network $G$
Output: Kronecker graph $\bar{z}$ which satisfies the notion of
    differential privacy
1: Partition the graph $G$ into $k$ clusters with similar sizes.
   Let the set of nodes in each of the clusters be $V_1, \ldots V_k$.
2: Let $A$ be an adjacency matrix of $G$ and $A[V_i, V_j]$ be a sub-
   matrix of $A$ of which set of rows are rows in $A$ correspond
   to $V_i$ and set of columns are columns in $A$ correspond to
   $V_j$.
3: For all $i, j$, give the matrix $A[V_i, V_j]$ as an input of a
   private Kronecker graph algorithm. Let the Kronecker
   graph obtained from the algorithm be $z[V_i, V_j]$.
4: Output $\bar{z} = \sum_{i,j} z[V_i, V_j]/k^2$.
5: return $\bar{z}$.
```

Next, we consider the adjacency matrix A of G. We divide the matrix based on the clustering results. As nodes in the graph are divided into k sets, columns and rows of the matrix are divided into k sets, denoted by $S_1, \ldots, S_k$. We then have a sub-matrix $A[V_i, V_j]$ from columns S_i and rows S_j of A, and the number of sub-matrices obtained from the division is k^2. Those k^2 sub-matrices are results of our pre-processing step.

We execute a differential privacy method for k^2 times and give a different sub-matrix to the method for every different execution. Then, we get a different output $z[V_i, V_j]$ at each iteration. The output of our post-processing is the average of those outputs.

We guarantee privacy of our pre- and post-processing in the following theorem.

Theorem 1. *If the differential privacy method we use in Line 3 of Algorithm 1 is (ε, δ)-differentially private, then our algorithm is also (ε, δ)-differentially private.*

Proof. Consider two neighboring graphs G and G' with adjacency matrices A and A'. Suppose we divide the graph into $V_1, \ldots, V_k$. The inputs to the differential privacy method, denoted by f, is then $A[V_i, V_j]$ and $A'[V_i, V_j]$. As G and G' are different by one edge, only one element in A and A' is different. We assume without loss of generality that the differing element is in $A[V_1, V_1]$ and $A'[V_1, V_1]$. By that, for $(i, j) \neq (1,1)$, we have $A[V_i, V_j] = A'[V_i, V_j]$ and for any possible z,

$$\mathbb{P}[f(A[V_i, V_j]) = z] = \mathbb{P}[f(A'[V_i, V_j]) = z].$$

As there is only one differing element between $A[V_1,V_1]$ and $A'[V_1,V_1]$, a graph corresponds to them are neighbors. By the fact that f is (ε,δ)-differentially private, we have

$$\mathbb{P}[f(A[V_1,V_1])=z] \leq \exp(\varepsilon)\,\mathbb{P}[f(A'[V_1,V_1])=z]+\delta.$$

Let $\mathscr{S}(\bar{z}) := \{[z[V_i,V_j]]_{1\leq i,j\leq k} : \sum_{i,j} z[V_i,V_j]/k^2=\bar{z}\}$. Denoting Algorithm 1 by g, we have

$$\begin{aligned}
&\mathbb{P}[g(A)=\bar{z}] \\
=& \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} \mathbb{P}[f(A[V_i,V_j])=z[V_i,V_j] \text{ for all } i,j] \\
=& \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} \prod_{i,j}\mathbb{P}[f(A[V_i,V_j])=z[V_i,V_j]] \\
=& \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} [\prod_{(i,j)\neq(1,1)} \mathbb{P}[f(A[V_i,V_j])=z[V_i,V_j]] \\
&\qquad \cdot\mathbb{P}[f(A[V_1,V_1])=z[V_1,V_1]]] \\
\leq& \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} [\prod_{(i,j)\neq(1,1)} \mathbb{P}[f(A'[V_i,V_j])=z[V_i,V_j]] \\
&\qquad \cdot(\exp(\varepsilon)\,\mathbb{P}[f(A'[V_1,V_1])=z[V_1,V_1]]+\delta)] \\
\leq& \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} [\prod_{(i,j)\neq(1,1)} \mathbb{P}[f(A'[V_i,V_j])=z[V_i,V_j]] \\
&\qquad \cdot\exp(\varepsilon)\,\mathbb{P}[f(A'[V_1,V_1])=z[V_1,V_1]]+\delta] \\
\leq& \exp(\varepsilon) \sum_{[z[V_i,V_j]]_{1\leq i,j\leq k}\in\mathscr{S}(\bar{z})} [\prod_{i,j}\mathbb{P}[f(A'[V_i,V_j])=z[V_i,V_j]] \\
&\qquad +\delta] \\
=& \exp(\varepsilon)\,\mathbb{P}[g(A')=\bar{z}]+\delta
\end{aligned}$$

By the same argument as above, we can also show that

$$\mathbb{P}[g(A)=\bar{z}] \geq \exp(-\varepsilon)\,\mathbb{P}[g(A')=\bar{z}]-\delta.$$

□

IV. Experimental Analysis

In this section, we empirically evaluate the performance of our proposed algorithm on real-world networks. We compare our technique with the different results used for fitting the parameters of the Kronecker model.

A. Settings

We use the following four social networks from the Stanford large dataset collection [31].

- CA-GrQC and CA-HepPh: These are co-authorship networks from arXiv [15]. The nodes of the graph represent authors, and an edge between two nodes exists if the authors jointly wrote a paper. The networks are undirected and the edges are unweighted.
- AS20: It is a technological infrastructure network [15], where each node represents a router on the internet. Edges indicate the presence of virtual or physical connection between the routers. The network is undirected and the edges are unweighted.
- Wiki-Vote: It is a who-votes-whom network, created from Wikipedia adminship election data. The nodes in the network represents volunteers and administrators, either casting a vote or being voted on respectively. An edge indicates a vote cast by the volunteer towards an administrator. The underlying network is directed, but for our experiments, we have converted it into an undirected network by dropping the direction of the edges.

In order to demonstrate the correctness of our proposed method, we compare it with other two approaches towards fitting Kronecker graph model to real-world networks. We generate the synthetic Kronecker networks from Kronfit [15] and private [11] moment-based estimators of the network using the provided library [31] and the code provided by Gleich et al. [32] respectively apart from our algorithm for all the experiments. Table 2 enumerates the results of our algorithm alongside that of Mir et al. [11] and Leskovec et al. [15].

We set the number of clusters in our algorithm to 2, and also set the size of Kronecker matrix, denoted by N_1, in all of the algorithms to 2. The results are then 2 × 2 matrices. As social networks in our interest are undirected, the output Kronecker matrices are symmetric. We denote the symmetric 2 × 2 matrices in the following form:

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

B. Results

The results in Table 2 are the values of a,b,c of the Kronecker matrices obtained from all of the datasets and algorithms.

TABLE 2: Kronecker matrices obtained from all algorithms and datasets when assuming that the matrices are in the form of $\left[\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right]$

Algorithm	**AS-20**	**CA-GrQC**	**CA-HepPh**	**Wiki-Vote**

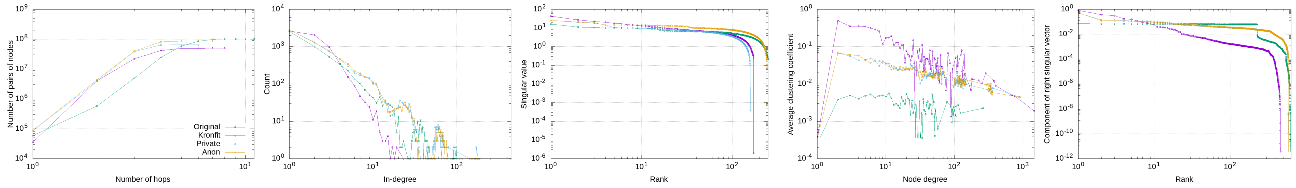
Kronfit [15]	a = 0.99
	b = 0.50
	c = 0.39
		a = 0.99
		b = 0.29
		c = 0.58
			a = 0.86
			b = 0.57
			c = 0.18
				a = 0.99
				b = 0.64
				c = 0.15

Private [11]	a = 0.99
	b = 0.56
	c = 0.01
		a = 0.99
		b = 0.54
		c = 0.14
			a = 0.87
			b = 0.60
			c = 0.11
				a = 0.99
				b = 0.55
				c = 0.01

Our work	a = 0.99
	b = 0.56
	c = 0.01
		a = 0.99
		b = 0.49
		c = 0.12
			a = 0.89
			b = 0.56
			c = 0.08
				a = 0.99
				b = 0.51
				c = 0.01

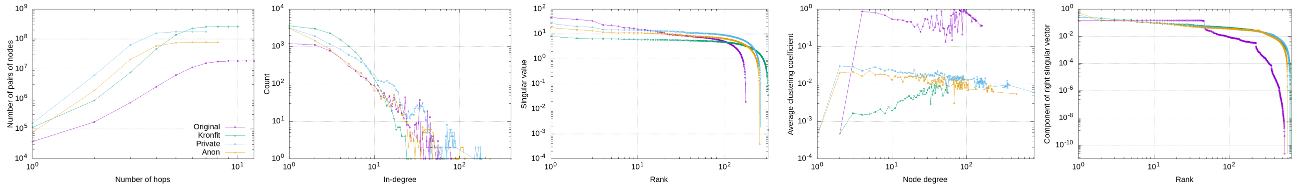
Given the parameter estimates of the graphs, we study the distribution of different graph properties of the realised Kronecker graphs along with the original graphs. All experiments pertaining to private and our method are conducted for $(0.2,0.1)$-differential privacy. We generate 100 synthetic graphs from the estimated parameters using all three methods, and compute different expected statistics [15] as enumerated next over these 100 graphs. To prove correctness of our idea, we use the most common graph clustering algorithm, spectral clustering with k-means [30], [33] in this experiment.

1) *Degree Distribution*: the distribution of the degree of the nodes.
2) *Hop-plot*: the number of reachable node pairs $g(h)$ within h hops, as a function of the number of hops h.



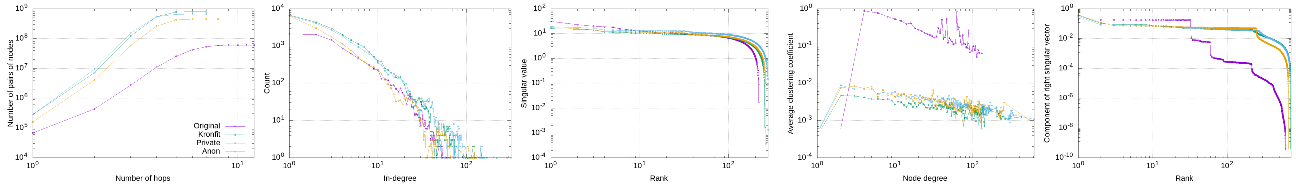
(a) Hop plot (b) Degree Distribution (c) Scree plot (d) Clustering (e) Network value

Fig. 1: AS-20: Overlayed patterns of real graph and three fitted Kronecker graphs using *Kronfit* [15], *Private* [11] and our algorithm *Anon*.



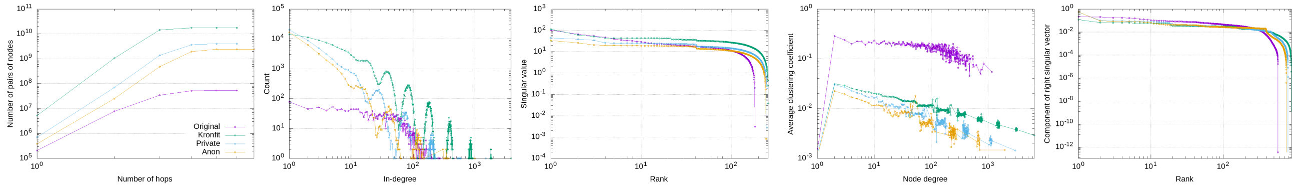
(a) Hop plot (b) Degree Distribution (c) Scree plot (d) Clustering (e) Network value

Fig. 2: CA-GrQC: Overlayed patterns of real graph and three fitted Kronecker graphs using *Kronfit* [15], *Private* [11] and our algorithm *Anon*.



(a) Hop plot (b) Degree Distribution (c) Scree plot (d) Clustering (e) Network value

Fig. 3: CA-HepPh: Overlayed patterns of real graph and three fitted Kronecker graphs using *Kronfit* [15], *Private* [11] and our algorithm *Anon*.



(a) Hop plot (b) Degree Distribution (c) Scree plot (d) Clustering (e) Network value

Fig. 4: Wiki-Vote: Overlayed patterns of real graph and three fitted Kronecker graphs using *Kronfit* [15], *Private* [11] and our algorithm *Anon*.

3) *Scree plot*: the eigenvalues (singular values) of the graph adjacency matrix, versus their rank using logarithmic scale
4) *Network values*: the distribution of eigenvector components (indicators of "network value") associated to the largest eigenvalue of the graph adjacency matrix
5) *Average clustering coefficient*: the average clustering coefficient as a function of the node degree. The clustering coefficient is a measure of the extent to which nodes in a graph tend to cluster together.

We compute and plot the above statistics for all four datasets, experimented using our approach alongside other two techniques as mentioned. In the overlayed patterns shown in Figures 1, 2, 3 and 4, *"Original"* refers to the original dataset, *"Kronfit"* refers to the Kronecker graph generated from the parameter estimates computed using *Kronfit* [15] algorithm, *"Private"* refers to the Kronecker graph generated from the parameter estimates computed using *Private* [11] algorithm and *"Anon"* refers to the Kronecker graph generated from the parameter estimates computed in our algorithm.

In all of the graphs, compared to that obtained from the other private Kronecker graph method [11], the data obtained from our method has a closer number of hops and degree distribution to the original graph. In most the graphs, our method even performs better than the original Kronfit method [15]. For the scree plot and network value, the previous two methods have already performed very well giving almost the same values as the original graphs. Our method gives the same performances with those previous works. On the other hand, our method does not perform very well for the clustering result, but that can be expected. As we divide graphs into clusters, the clustering structures in each of the clusters should be different from the original graph. However, because the previous works also cannot give a precise information for the clustering structure, we can see in the results that our method is not significantly worse than them.

By our pre- and post-processing technique, we can significantly reduce the computation time of differential privacy methods. All of the methods including Kronfit and private moment-based estimators take $\Omega(n^3)$ when n is the number of nodes in our social networks. By our technique, although we have to execute the differential privacy methods for k^2 times, as the number of nodes is now n/k, it takes $\Omega((n/k)^3)$ time in each execution. The overall computation is therefore $\Omega((n/k)^3 \cdot k^2) = \Omega(n^3/k)$. We reduce the computation time of the methods by at most k times.

V. Conclusion and Future Works

Using graph clusters, we propose a pre- and post-processing technique to improve accuracy of differential privacy method based on Kronecker graphs. Then, we show by our experimental results that, by our technique, we can significantly improve the accuracy of social network publication in terms of number of hops and degree distributions. Also, we get similar results to the previous works for other utility factors. We also significantly reduce the computation time of the methods.

As indicated by our experiments, we have improved the results of the existing method of Mir et al. [11]. It is very straightforward to see that our process can be applied also to other methods. We strongly believe that we can see similar accuracy improvements when we apply this technique to other methods, and we plan to verify that in our future works.

References

[1] J. P. Scott and P. J. Carrington, *The SAGE Handbook of Social Network Analysis*. Sage Publications Ltd., 2011.

[2] L. Backstrom, C. Dwork, and J. M. Kleinberg, "Wherefore art thou R3579X?: Anonymized social networks, hidden patterns, and structural steganography," *Commun. ACM*, vol. 54, no. 12, pp. 133–141, 2011.

[3] C. Dwork, "Differential privacy," in *ICALP 2006*, 2006, pp. 1–12.

[4] A. Sealfon, "Shortest paths and distances with differential privacy," in *PODS 2016*, 2016, pp. 29–41.

[5] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *ICDM 2009*, 2009, pp. 169–178.

[6] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *TCC 2013*, 2013, pp. 457–476.

[7] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra, "Privacy via the Johnson-Lindenstrauss transform," *arXiv preprint arXiv:1204.2606*, 2012.

[8] F. Ahmed, A. X. Liu, and R. Jin, "Publishing social network graph eigen-spectrum with privacy guarantees," *IEEE Transactions on Network Science and Engineering*, pp. 1–14, 2019.

[9] H. H. Nguyen, A. Imine, and M. Rusinowitch, "Differentially private publication of social graphs at linear cost," in *ASONAM 2015*, 2015, pp. 596–599.

[10] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially private network data release via structural inference," in *KDD 2014*, 2014, pp. 911–920.

[11] D. J. Mir and R. N. Wright, "A differentially private estimator for the stochastic kronecker graph model," in *EDBT/ICDT Workshops 2012*, 2012, pp. 167–176.

[12] D. Li, W. Zhang, and Y. Chen, "Differentially private network data release via stochastic kronecker graph," in *WISE 2016*, 2016, pp. 290–297.

[13] R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala, "Optimizing error of high-dimensional statistical queries under differential privacy," *Proceedings of the VLDB Endowment*, vol. 11, no. 10, pp. 1206–1219, 2018.

[14] J. Ma, Q. Zhang, J. Lou, J. C. Ho, L. Xiong, and X. Jiang, "Privacy-preserving tensor factorization for collaborative health data analysis," in *CIKM 2019*, 2019, pp. 1291–1300.

[15] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.

[16] M. Mahdian and Y. Xu, "Stochastic kronecker graphs," *Random Struct. Algorithms*, vol. 38, no. 4, pp. 453–466, 2011.

[17] J. Kepner and J. Gilbert, *Graph Algorithms in the Language of Linear Algebra*. Society for Industrial and Applied Mathematics, 2011.

[18] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[19] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 2007.

[20] A. Campan and T. M. Truta, "Data and Structural k-Anonymity in Social Networks," in *PinKDD 2008*, 2008, pp. 33–54.

[21] F. Yu, M. Chen, B. Yu, W. Li, L. Ma, and H. Gao, "Privacy preservation based on clustering perturbation algorithm for social network," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 11 241–11 258, 2018.

[22] S. Jones and E. O'Neill, "Feasibility of structural network clustering for group-based privacy control in social networks," in *SOUPS 2010*, 2010, pp. 1–13.

[23] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on networks," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 341–366, 2017.

[24] A. D. Smith, "Efficient, differentially private point estimators," *CoRR*, vol. abs/0809.4794, 2008.

[25] M. Kim and J. Leskovec, "The network completion problem: Inferring missing nodes and edges in networks," in *SDM 2011*, 2011, pp. 47–58.

[26] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, and C. Faloutsos, "Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication," in *PKDD 2005*, 2005, pp. 133–145.

[27] M. Mahdian and Y. Xu, "Stochastic kronecker graphs," *Random Struct. Algorithms*, vol. 38, no. 4, pp. 453–466, 2011.

[28] RouteViews, "University of Oregon route views project, online data and reports," *http://www. routeviews.org*, 1997.

[29] D. F. Gleich and A. B. Owen, "Moment-based estimation of stochastic kronecker graph parameters," *Internet Mathematics*, vol. 8, no. 3, pp. 232–256, 2012.

[30] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS 2002*, 2002, pp. 849–856.

[31] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[32] D. F. Gleich, "Kronecker moment based estimation code," https://dgleich.com/gitweb/?p=kgmoments;a=summary, 2011.

[33] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.