Two-Stage Framework for Accurate and Differentially Private Network Information Publication

Maneesh Babu Adhikari[1], Vorapong Suppakitpaisarn[2](✉), Arinjita Paul[1], and C. Pandu Rangan[1]

[1] Indian Institute of Technology Madras, Chennai, India
adhimaneesh1998@gmail.com
{arinjita,prangan}@cse.iitm.ac.in
[2] The University of Tokyo, Tokyo, Japan
vorapong@is.s.u-tokyo.ac.jp

Abstract. We propose a novel mechanism to release an accurate and differentially private estimate of the link set of social networks. Differential privacy is one of the most common notations to quantify the privacy level of information publication. Several methods have been proposed to publish edge set information, among which one of the notable mechanisms is based on stratified sampling. While it is very scalable, social network information can be significantly altered by this technique. In fact, when we use mechanism based on stratified sampling, a totally random network may get published even when the input network is sparse. We aim to overcome this drawback in our work. We provide an efficient two-stage mechanism to control the edge set size and quality independently. To confirm the practical utility of our proposal, we apply it to the maximum matching problem when the edge information is spread between two different bipartite networks. We validate through experiments that the error induced by our framework is at least 20 times smaller than that of the original stratified sampling based mechanism when privacy level is 5. In addition, the computation time of our framework is 3 times shorter than the original method.

Keywords: Differential privacy · Stratified sampling · Maximum matching

1 Introduction

With the expeditious increase in the usage of online services, huge amounts of data are being generated everyday. Most of the generated data can be modelled as graphs, e.g., social network data, email data, etc. In these networks, people can be modelled as nodes and their interactions can be modelled as edges. The analysis of such data is pivotal in the spheres of medical research, fraud detection, recommendation engines, to name a few. However, the data owners are reluctant to share such data over privacy concerns of the users that are part of the network.

S. Chellappan et al. (Eds.): CSoNet 2020, LNCS 12575, pp. 267–279, 2020.
https://doi.org/10.1007/978-3-030-66046-8_22

Several anonymization techniques have been proposed to preserve privacy of individuals' data, but none of them provides mathematical guarantees with the availability of auxiliary information. Narayanan and Shmatikov in [1] proposed a technique to break the anonymity of naively anonymized Netflix data set using auxiliary information resulting in privacy loss of millions of users. Hence, it is important that the data is provably anonymized (even with access to auxiliary information) before it is released for research purposes. This motivates us to work with differential privacy to overcome the above discussed privacy risks.

The concept of differential privacy was formally introduced in [2] by Dwork. Differential privacy provides a mathematically provable guarantee of privacy preservation against linkage attacks, statistical attacks and leveraging attacks. It guarantees that the outcome of any analysis is equally likely, independent of the presence of any single entity in the data. Therefore, the risk of identifying private information with the availability of auxiliary information is nullified.

In this paper, we study the problem of publishing the link set of a social network. Several techniques have been proposed to achieve this, among which one of the notable mechanisms is based on stratified sampling [3]. While the computation time of this technique is quite short, social network information can significantly be altered by this technique. The published output network may be a completely random network independent of the input network when the input network is sparse.

1.1 Our Contribution

We introduce a stratified sampling based mechanism to publish an accurate estimate of the edge set with strong privacy guarantees. We propose a two-stage [ε_1:ε_2]-differentially private algorithm to release an anonymized, accurate estimate of the true edge set. In the first stage, we develop an ε_1-differentially private algorithm to sample an estimate of the size of the un-anonymized edge set. In the second stage, we present an ε_2-differentially private algorithm to sample an edge set of specific size sampled in the first step. The two-stage algorithm gives us a better control over the size of the edge set independent of its quality. It is particularly helpful when the resultant analysis on the published anonymized graph is highly sensitive to the number of edges. To show practical utilities of our proposed differential private mechanism, we consider a private variant of the maximum matching problem. The maximum matching size of a graph is highly sensitive to the presence or absence of edges in sparse graphs, although highly invariant in dense graphs. It gives us an ideal scenario to present the utility of our algorithm. We consider the setting in which two disjoint bipartite networks are connected by a set of inter-connecting (edges connecting nodes in different networks) edges. The node information of both networks and inter-connecting edge information is public whereas the intra-edge (edges connecting nodes in the same network) information of both the networks is private. We deal with the problem of finding the size of the maximum matching of the union of both the networks when one of the networks in anonymized. Comparing with the classical stratified sampling technique in [3], we present empirical results on the publicly available Moreno Crime dataset [4]. We also show results on networks synthesized from the R-MAT model [5]. We demonstrate drastic improvements in performance in sparse graphs when one of the input networks is anonymized with our proposed mechanism.

1.2 Related Works

There are several studies in literature aiming to protect if a particular person is in a social network (node differential privacy) [6–8]. Our discussion in this work focuses on hiding the information if an edge is in a graph or edge differential privacy [9]. Though we are not currently dealing with node differential privacy, we strongly believe that our framework can be extended to this privacy notation as well.

Several differential privacy works have been introduced to publish specific information of social networks. For example, Hay et al. [9] aims to output the degree distribution of an input graph, Kenthapadi et al. [10] aims to publish the shortest path length for all pairs of nodes, whereas Ahmed et al. [11] aims to output the eigenvectors of the input network's adjacency matrix. Although we experiment with the maximum matching size, our framework is not limited to any particular graph property. It outputs an accurate edge set of the social network and users can derive any network information from it.

There are some works which, similar to us, aim to publish an accurate estimate of the edge set to the users. Many of them are based on Kronecker graph [12–14]. While Kronecker graph can effectively capture global information of a social network like diameter, clusters, or degree distribution, we believe that it cannot capture properties that need local knowledge such as shortest path or maximum matching size. Therefore, techniques based on Kronecker graph and this work could be complementary to each other.

We rely on maximum matching size to show the accuracy of our proposal. Differentially private maximum matching is considered in several works such as [15–17]. They consider a situation in which one party holds all the information, and wants to release the maximum matching information from the entire social network they have. On the other hand, our experimental results focus on a situation when two parties want to exchange private information in order to calculate the maximum matching size.

2 Preliminaries

2.1 Differential Privacy [2]

We work with the concept of private databases to define differential privacy. A database is viewed as a collection of records with each record describing private information of some entity. Distance between two databases is defined as the number of records in which these two databases differ. Two databases d, d' are neighboring databases if they differ in exactly 1 record i.e, $\|d - d'\|_1 = 1$.

Definition 1. *Consider a randomized algorithm A which takes a database as input and outputs a member of Range(A). We say that A is ε-differentially private if, for all neighboring databases d, d', and, for any $S \subseteq$ Range(A), we have* $\Pr[A(d) \in S] \leq e^{\varepsilon} \Pr[A(d') \in S]$. *where e is Euler's number.*

Informally, differential privacy bounds the amount by which output of an algorithm can change when a single record is either added or removed to/from a database. When a publication is under differential privacy, one cannot distinguish between two tables different by one record by the publication, and, hence, cannot know information of a particular person even though all other persons disclose their information.

We discuss an algorithm that satisfies the differential privacy notation. We begin by introducing l_1-sensitivity. Suppose D is the set of all possible databases, and the original information to be published from table $d \in D$ is $f(d)$.

Definition 2. *Let f be a function from a set of databases D to the set $\mathbb{R}^k$. The l_1-sensitivity of f is $\Delta f = \max\limits_{\|d-d'\|_1=1} \|f(d) - f(d')\|_1$.*

The l_1 sensitivity of a function f is the maximum change in f obtained by the inclusion or exclusion of a single entity. In other words, it bounds the margin by which the output needs to be altered to hide the participation of a single entity.

Exponential Mechanism. Exponential mechanism was first introduced by McSherry and Talwar in [18]. It preserves ε-differential privacy when the output range S of an algorithm is discrete. It maps every (database, output) pair to a quality score given by the quality function $Q : D \times S \rightarrow \mathbb{R}$. Then, it samples and outputs an element $s \in S$ with likelihood proportional to $g(s) = \exp\left(\frac{\varepsilon Q(d,s)}{2\Delta Q}\right)$. Let $C = \sum_{s \subseteq S} g(s)$. The probability with which we output $s \subseteq S$ is then equal to $\Pr[s] = g(s)/C$. Intuitively, for some fixed database d, it is likely that the algorithm outputs an element of S with large quality score. We assign higher scores to preferred outputs and lower scores to bad outputs.

2.2 Differential Privacy vs Social Network Privacy

Social Networks like Facebook, Twitter, or Tinder can be modelled as graphs. Accounts (people) can be modelled as nodes and their relations can be modelled as edges. The analysis of these graph structures is pivotal to provide relevant recommendations and improve user experience. Such graph data has to be anonymized before publishing it for research purposes to preserve the privacy of users. Differential privacy can be extended to graph anonymization and provide a provable social network privacy.

We discuss the equivalence of neighboring databases in relation to graphs.

Definition 3 *(Edge Differential Privacy [14])***.** *Graphs G and G' are said to be neighbors with respect to edge differential privacy if $\|E - E'\|_1 = 1$ where $G = (V, E)$ and $G' = (V, E')$.*

Based on the previous definition, differential privacy notion for graphs is similarly defined as follows:

Definition 4 *(Differential Privacy on Graphs)***.** *Consider a randomized algorithm A which takes a graph as input and outputs a member of Range(A). We say that A is ε-differentially private if, for all neighboring graphs G, G', and, for any $S \subseteq$ Range(A), we have* $\Pr[A(G) \in S] \leq e^{\varepsilon} \Pr[A(G') \in S]$.

Differential privacy hides the presence or absence of a data entity in neighboring databases by limiting the change in output introduced by that entity. In this case, neighboring databases are graphs that differ by one edge i.e., G and G' are neighboring graphs if G' can be obtained from G by adding or removing an edge. Because of this property, attackers cannot distinguish between a graph with a link and a graph without. The information whether two people are linked to each other is then protected if we have an algorithm that satisfies the notion of edge differential privacy.

2.3 Stratified Sampling

To sample an element according to a probability distribution, standard sampling technique [19] runs in linear time in the number of possible outcomes. If the number of possible outcomes is exponential in the input size, standard sampling is no longer tractable. Stratified sampling [3], however, takes advantage of the presence of homogeneous subgroups where the probability of any two outcomes in a homogeneous subgroup is same.

Let R be the finite set of possible outcomes and X be a discrete random variable defined on probability space (p, R), i.e., $p : x \in R \mapsto \Pr(X = x)$. Let $(R_0, R_1, ..., R_k)$ be a partition of R into k homogeneous subgroups. For any subgroup R_i and any two elements $x, x' \in R_i$, we have $p(x) = p(x')$. Now, according to stratified sampling, sampling from original distribution is same as:

1. Sampling a subgroup according to the relative subgroup probabilities using the standard sampling technique.
2. Sampling an element uniformly from the chosen subgroup in step 1.

The running time of sampling is greatly enhanced depending on the number of subgroups (denoted by k).

2.4 Matching Theory in Graphs (c.f. Chapter 7 of [20])

In this section, we introduce some basic graph definitions used in later sections.

Definition 5 *(Bipartite Graph)***.** *A graph $G = (V, E)$ is a bipartite graph if there exists a partition of V into X and Y such that there is no edge between two vertices in X and there is no edge between two vertices in Y.*

Definition 6. *A matching M of a graph $G = (V, E)$ is a subset of edges E such that no two edges have a common vertex. A matching M is said to be a maximum matching if there exists no other matching M' of G with larger number of edges.*

2.5 Previous Work on Differential Privacy by Stratified Sampling

In the forward message of [3], the goal is to communicate a differentially private estimate of the adjacent nodes of a specific node u in the graph $G = (V, E)$. Let U be the set of adjacent nodes of node u in graph $G = (V, E)$, i.e., for all $a \in V$, $(a, u) \in E \iff a \in U$ and let U^* be some approximation of U. The quality function for this mechanism is then defined as $Q(U^*) = |U \cap U^*| + |\overline{U \cup U^*}|$. The probability distribution is then calculated over the power set R of all nodes $V^- = (V \setminus \{u\})$ according to the above quality function. A set U^* is then sampled from R according to this probability distribution and is communicated as a differentially private estimate of U. From the definition of $Q(U^*)$, it can be observed that $Q(U^*) \in [0, |V^-|]$. The number of possible quality values is $|V|$, although the number of possible node sets U^* is exponential in $|V|$. Hence, stratified sampling can be used to sample U^* from R efficiently. The probabilities are calculated in log space since the exponents may otherwise blowup. The log space probability is given by $\ln(\Pr[U^*]) = \frac{\varepsilon Q(U^*)}{2\Delta Q} - \ln(C)$. The normalizing constant C can be calculated efficiently by calculating the number of node sets for each possible quality value. The normalizing constant C in this case turns out to be $C = (1 + \exp(\varepsilon/(2\Delta Q)))^{|V|-1}$ as described in [3]. C being in the form $(1 + A)^B$ makes it feasible to work in log space. The running time of this framework is linear in the total number of nodes, i.e. $\mathcal{O}(|V|)$.

3 Proposed Frameworks

In this section, we describe the working methodology of two differentially private frameworks. Each of the two frameworks can independently be used for computing the differentially private estimate of the edge set of any graph. Framework 1 is a direct extension of the mechanism in [3]. Framework 2 is a further extension of Framework 1 which gives us better control over the size of the differentially private edge set independent of its quality. We call Framework 2 as $[\varepsilon_1 : \varepsilon_2]$-differentially private algorithm indicating the dependence of the size of the private edgeset on ε_1 and its quality on ε_2.

3.1 Framework 1

In our framework, the goal is to communicate a differentially private estimate of the edges of graph $G = (V, E)$ when node information V is public. The mechanism described in [3] can easily be extended to reach that goal. Let E^* be the differentially private estimate of E. Then, the quality function is similarly defined as $Q(E^*) = |E \cap E^*| + |\overline{E \cup E^*}|$. The probability distribution is then calculated over power set S of all edges possible with vertex set V according to this quality function. We adopt the same sampling technique as described [3] and obtain differentially private estimate E^*. The running time of Framework 1 is linear in the total number of possible edges with $|V|$ nodes, which is $\mathcal{O}(|V|^2)$.

3.2 Framework 2

One of the main problems we encountered with Framework 1 was that

> the cardinality of $(E\Delta E^*)$ is often very high even when the privacy level is very high (ε is set to a very large value). If the analysis to be performed on private graph is very sensitive to addition or deletion of edges, then the private graph $G = (V, E^*)$ is not an ideal estimate of $G = (V, E)$.

So, we now propose a two stage mechanism to control the cardinality and quality of the differentially private edge set E^* independently.

- Algorithm Stage 1: Mechanism to output a differentially private estimate x of the size of the true edge set $|E|$.
- Algorithm Stage 2: Mechanism to output the differentially private estimate E^*, where $|E^*| = x$.

Algorithm Stage 1. This stage of the algorithm can be viewed as answering a query when the true answer is $|E|$. One might think that Laplace mechanism [2], one of the most common mechanisms, can be applied here. However, it cannot be applied to achieve differential privacy in this case, since it might output non-integral values whereas we need the output to be an integer that can be utilized in Stage 2 of the algorithm. Let E^t be the set of all possible edges in the graph $G = (V, E)$, i.e. $E^t = \{(u, v) : u, v \in V\}$. As the integer output x is always in $[0, |E^t|]$, the set of possible outputs is finite. The quality function for this mechanism is defined as $Q_1(x) = |E^t| - \text{abs}(x - |E|)$ where abs denotes the absolute value. The quality value $Q_1(x)$ deteriorates as we move away from the true value $|E|$ and is uniquely maximized when $x = |E|$. We adopt the same method as described in Framework 1 and work in log space. To calculate the log space probabilities, we need to obtain a closed form expression for the normalizing constant $C = \sum_{x=0}^{|E^t|} \exp\left(\frac{\varepsilon_1 . Q_1(x)}{2.\Delta Q_1}\right) = \frac{a^{|E^t|}}{a-1} \cdot \left[a + 1 - a^{-|E|} - a^{|E|-|E^t|}\right]$, where $a = \exp(\frac{\varepsilon_1}{2.\Delta Q_1})$. We ignore last two terms in RHS in the above expression as they are much smaller than $a + 1$. We now calculate the log space probabilities and sample the differentially private estimate x. Note that $\Delta Q_1 = 1$.

Algorithm Stage 2. In this stage of the algorithm, we need to output a differentially private edge set E^* from all possible edge sets of size x. We adopt the same quality function as described in Framework 1, which is $Q_2(E^*) = |E \cap E^*| + |\overline{E \cup E^*}|$. In order to sample a differentially private estimate E^*, we first need to find out the probability distribution over all possible edge sets of size x. The first task is to compute the normalizing constant efficiently in order to calculate the probabilities. We denote the quality value by q and the cardinality of intersection by i, i.e. $Q_2(E^*) = q$ and $|E \cap E^*| = i$. From the definition of Q_2, we obtain the equality $i = \frac{1}{2}(q + x + |E| - |E^t|)$. Hence, there exists an edge set of size x with quality value q if and only if $(q + x + |E| - |E^t|)$ is an even number along with other boundary conditions. If there exists an edge set with the above requirements, then such an edge set can be obtained by selecting i edges from E and remaining $(x-i)$ edges from $(E^t - E)$ respectively. Therefore, the number of such edge sets with quality q is given by $N(q) = \binom{|E|}{i}\binom{|E^t-E|}{x-i}$. The normalizing constant is then given by

$$C = \sum_q N(q) \cdot \exp\left(\frac{q\varepsilon_2}{2\Delta Q_2}\right) \tag{1}$$

The summation above is over all possible quality values q discussed in the previous paragraph. Unfortunately, Eq. (1) does not have a closed form expression. Since we are working in the log space, we approximate $\ln(C)$ instead of computing C which can later be used to compute the log space probabilities.

For all q, let us denote $s_q = N(q) \cdot \exp(\frac{q\varepsilon_2}{2\Delta Q_2})$ and $s_{\max} = \max_q s_q$. Then, $\ln(C) = \ln(s_{\max} \cdot \sum_q \frac{s_q}{s_{max}}) = \ln(s_{max}) + \ln\left(\sum_q \exp\left(\ln(s_q) - \ln(s_{\max})\right)\right)$. We only consider the terms for which $\ln(s_q) - \ln(s_{max}) \geq -\alpha$ for a large constant α and all the remaining terms are ignored. We used $\alpha = 650$ in our experiments ignoring all the values s_q for which the ratio s_q/s_{max} is less than e^{-650} which we think is a reasonably small value to neglect.

The second task is to sample an edge set according to this probability distribution. From the definition of Q_2, it can be observed that $Q_2(E^*) \in [0, |E^t|]$ i.e., the number of possible quality values is $(|E^t| + 1)$, although the number of possible edge sets is exponential. In relation to stratified sampling, a group of edge sets with same quality value can be viewed as a homogeneous subgroup since equal quality values induce equal probabilities. Sampling an edge set using stratified sampling is done as follows:

- Step 1: Sample a quality value q according to the relative homogeneous subgroup probabilities using the standard sampling technique.
- Step 2: Uniformly, sample an edge set of size x and quality value q.

Step 1 The probability $\Pr(q)$ associated with a homogeneous subgroup induced by quality q is the sum of probabilities of all edge sets with quality q. The probability associated with an edge set having quality q can be computed using $\Pr[s]$ as defined in Section II where $Q(d, s) = q$. Therefore, $\Pr(q) = N(q) \cdot \Pr[s]$ where s in an edge set with quality value q. The log space probability associated with a homogeneous subgroup with quality q is then given by

$$\ln(\Pr(q)) = \ln(N(q)) + \frac{\varepsilon_2 q}{2\Delta Q_2} - \ln(C) \tag{2}$$

The probability distribution over all possible quality values can be calculated in $\mathcal{O}(|E^t|)$ time and standard sampling can be done in $\mathcal{O}(|E^t|)$ time and space.

Step 2 The challenge is to sample an edge set E^* of size x and quality q uniformly from all possible edge sets when the original edge set is E. Recall that, when $|E \cap E^*| = i$, we have $i = \frac{1}{2}(q + x + |E| - |E^t|)$. For a given q and x, the size of intersection or the number of common edges in E and E^* is fixed to i. The probability of selecting some set of i edges from E without repetition is given by $1/\binom{|E|}{i}$. The remaining $(x-i)$ edges of E^* are then uniformly chosen from the set $(E^t - E)$. The probability of selecting some set of $(x-i)$ edges from $(E^t - E)$ without repetition is given by $1/\binom{|E^t - E|}{x-i}$.

We then publish the union of i and $(x-i)$ edges as our differentially private estimate E^*. We used the Fisher Yates Shuffling Algorithm [21] to randomly pick a few items from a large set without repetition. Note that $\Delta Q_2 = 1$.

4 Use Case on Bipartite Matching

To demonstrate the performance of our algorithm, we consider a private variant of the maximum matching problem. Maximum matching in bipartite graphs is a heavily studied area with immense practical importance. The size of maximum matching of a graph is highly sensitive to the presence or absence of edges in sparse graphs whereas it is highly invariant in dense graphs. It presents us with an ideal scenario to demonstrate the comparative utilities of our framework.

4.1 Problem Setting

We work in the similar setting as described in [3]. Assume that $G_1 = (X_1, Y_1, E_1)$ and $G_2 = (X_2, Y_2, E_2)$ are two node disjoint bipartite networks owned by organizations I_1 and I_2 respectively. Although every node belongs to one of the two bipartite networks, edges may span across both the networks. We denote the set of edges spanning across the two networks by $E_{12} \subseteq \{(u,v) \mid u \in X_1, v \in Y_2\}$ and $E_{21} \subseteq \{(u,v) \mid u \in Y_1, v \in X_2\}$. Our privacy assumptions are as follows:

- public information : $X_1, Y_1, X_2, Y_2, E_{12}, E_{21}$
- private information known only to I_1: E_1
- private information known only to I_2: E_2

We enable one of the organizations to compute the maximum matching size of the union of both the networks while hiding the intra-edge information of the other network from the organization. Suppose the organization to compute the maximum matching size is I_2, then I_2 will not know E_1 but its differentially private estimation of the set, denoted by E_1^*. It then must estimate the maximum matching size of $G_u = (X_1 \cup X_2, Y_1 \cup Y_2, E_u)$ where $E_u = E_1 \cup E_2 \cup E_{12} \cup E_{21}$.

4.2 Practical Utility

The problem setting is useful in many practical situations. Consider a situation when two institutions I_1 and I_2 want to see if a collaboration between them is beneficial to them or their customers. If they do not collaborate, their benefits are the maximum matching size of G_1 added by the maximum matching size of G_2, while the benefits are the maximum matching size of G_u when they collaborate.

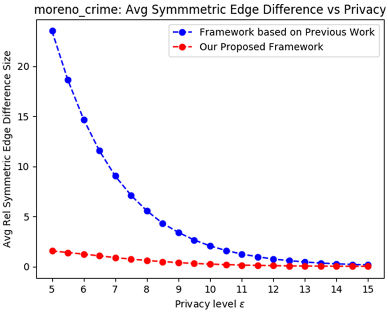

Fig. 1. Average relative symmetric edge difference ($|E_1 \varDelta E_1^*|/|E_1|$) for $\varepsilon =$ 5 to 15, moreno crime dataset

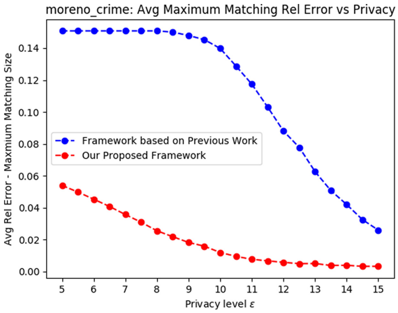

Fig. 2. Average relative maximum matching error for $\varepsilon = 5$ to 15, moreno crime dataset

We can consider I_1 and I_2 as two different university departments which are matching students with professors for graduate projects. The maximum matching size is then the number of students who are matched with their preferred professors. Two departments may then collaborate if the matching matching size is significantly increased by joining two social networks. On the other hand, we may want to protect if a link exists as students may not want to reveal if they prefer one professor over others. The same setting can be applied when two airlines wants to decide if they should have codeshare flights or when companies wants to decide if they should merge with each other.

5 Experimental Results

We apply the frameworks on E_1 to obtain the differentially private estimate E_1^* and then compute the maximum matching size of $G_u^* = (X_1 \cup X_2, Y_1 \cup Y_2, E_1^* \cup E_2 \cup E_{12} \cup E_{21})$. We refer to Framework 1 and Framework 2 in the above graph plots as 'Framework based on Previous Work' and 'Our Proposed Framework' respectively. All the experiments have been executed on Lenovo y50–70 machine with 16 GB RAM, using python 3.6 without parallel computations. Results in Figs. 1, 2, 3 are based on the publicly available moreno crime dataset [4] which is randomly divided into two bipartite networks with inter and intra edge connections as described in the above sections.

We first examine the variation of relative symmetric edge difference with privacy leverage in Fig. 1. In this figure, privacy leverage ε for Framework 2 indicates the summation of privacy levels at Algorithm stage 1 and stage 2 i.e., $\varepsilon = \varepsilon_1 + \varepsilon_2$. Our framework exhibits exponential improvements in the quality of differentially private edge set for smaller values of ε. We obtain better approximations with stronger privacy guarantees i.e., average relative symmetric edge difference is 1.56 for Framework 2 when compared to 23.49 for Framework 1 when $\varepsilon = 5$. Differentially private estimate in Framework 2 for $\varepsilon < 5$ is a bad approximation (but is still exponentially better than Framework 1) with relative error close to 2. Therefore, for an edge set publication, we recommend using $\varepsilon \geq 5$.

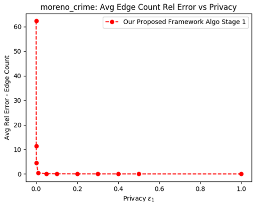

Fig. 3. Average relative edge count (*abs* $(|E_1| - |E_1^*|)/E_1$) error in algorithm Stage 1 for $\varepsilon_1 = 0.0001$ to 1, moreno crime dataset

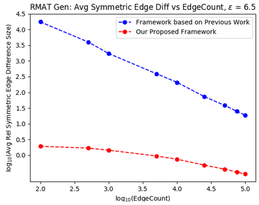

Fig. 4. Average relative symmetric edge difference $(|E_1 \Delta E_1^*|/|E_1|)$ vs total edge count $(|E^u|)$ for $\varepsilon = 6.5$, R-MAT generated graphs

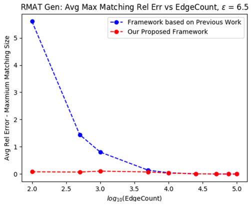

Fig. 5. Average relative maximum matching error vs total edge count$(|E_u|)$ for $\varepsilon = 6.5$, RMAT generated graphs

Figure 2 shows the relation between average relative maximum matching error and privacy leverage for ε between 5 and 15. Relative maximum matching error is as low as 0.05 for Framework 2 whereas it is 0.15 for Framework 1 when $\varepsilon = 5$. Average relative maximum matching error is quite small even for $\varepsilon < 5$ even though the symmetric edge difference is high owing to the fact that size of maximum matching can be same for completely different graphs. Figure 3 shows the variation of cardinality of differentially private edge set $|E_1^*|$ with privacy leverage ε_1 at stage 1 of the algorithm in Framework 2. We can see that the error is very small for any $\varepsilon_1 \geq 0.1$. Therefore, we propose to set value of $\varepsilon_1 = 0.1$.

We next report the results on the synthetic R-MAT [5] graphs. Multiple R-MAT graphs with constant number of nodes i.e., $|X_1 \cup X_2| = |Y_1 \cup Y_2| = 1024$ and edges ranging from 100 to 100000 have been generated. We further maintained the property that $|X_1| = |Y_1| = 512$ for the divided graphs with no further condition on the edges. We then examined the average relative symmetric edge difference and average relative maximum matching error as a function of edge cardinality of the union of graphs $|E_u|$ in Fig. 4 and Fig. 5 respectively. The relative symmetric edge difference is 0.94 for Framework 2 whereas it is 391.89 for Framework 1 when $|E_u| = 5000$ and $\varepsilon = 6.5$ proving the drastic improvement in performance we have achieved in relatively sparse graphs.

We also validated through our experimentation that Framework 2 is around 3 times faster then Framework 1.

6 Conclusions and Future Works

In this work, we consider the case where two parties own different parts of a social network. Their goal is to calculate certain network properties while preserving the information of their users. We propose a two-stage framework that can increase the accuracy of the calculation result by up to 20 times and reduce the computation time by 3 times.

In our work, we have considered a scenario when node information is public and edge information is private. But many cases even require the node information to be hidden. Also, a link might contain other information in addition to the person who are incident on it. We believe our mechanism can be extended to protect such private information, and we plan to do that in the future. We also plan to compare our mechanism with techniques other than stratified sampling, and work on different use case.

References

1. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: SP. IEEE **2008**, 111–125 (2008)
2. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
3. Roohi, L., Rubinstein, B.I., Teague, V.: Differentially-private two-party egocentric betweenness centrality. INFOCOM IEEE **2019**, 2233–2241 (2019)
4. “Crime network dataset - KONECT,” April 2017. http://konect.uni-koblenz.de/networks/moreno_crime
5. Chakrabarti, D., Zhan, Y., Faloutsos, C.: R-MAT: a recursive model for graph mining. SDM SIAM **2004**, 442–446 (2004)
6. Ullman, J., Sealfon, A.: Efficiently estimating erdos-renyi graphs with node differential privacy. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada, 2019, pp. 3765–3775 (2019)
7. Day, W.-Y., Li, N., Lyu, M.: Publishing graph degree distribution with node differential privacy. ICDM **2016**, 123–138 (2016)
8. Kasiviswanathan, S.P., Nissim, K., Raskhodnikova, S., Smith, A.: Analyzing graphs with node differential privacy. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 457–476. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_26
9. Hay, M., Li, C., Miklau, G., Jensen, D.D.: Accurate estimation of the degree distribution of private networks. In: The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA **6–9**(2009), pp. 169–178 (2009)
10. Kenthapadi, K., Korolova, A., Mironov, I., Mishra, N.: Privacy via the Johnson-Lindenstrauss transform, arXiv preprint arXiv:1204.2606 (2012)
11. Ahmed, F., Liu, A.X., Jin, R.: Publishing social network graph eigen-spectrum with privacy guarantees. In: IEEE Transactions on Network Science and Engineering, pp. 1–14 (2019)
12. Mir, D.J., Wright, R.N.: A differentially private estimator for the stochastic kronecker graph model. EDBT/ICDT Workshops **2012**, 167–176 (2012)
13. Li, D., Zhang, W., Chen, Y.: Differentially private network data release via stochastic kronecker graph. In: Cellary, W., Mokbel, M.F., Wang, J., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2016. LNCS, vol. 10042, pp. 290–297. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48743-4_23

14. Paul, A., Suppakitpaisarn, V., Bafna, M., Rangan, C.P.: Improving accuracy of differentially private kronecker social networks via graph clustering. In: ISNCC 2020, 2020 (accepted)
15. Hsu, J., Huang, Z., Roth, A., Roughgarden, T., Wu, Z.S.: Private matchings and allocations. SIAM J. Comput. **45**(6), 1953–1984 (2016)
16. Varma, N., Yoshida, Y.: Average sensitivity of graph algorithms, arXiv preprint arXiv:1904.03248 (2019)
17. Huang, Z., Zhu, X.: Scalable and jointly differentially private packing. In: 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9–12, 2019, Patras, Greece, ser. LIPIcs, 2019, pp. 73:1–73:12 (2019)
18. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS 2007. IEEE 94–103 (2007)
19. Cochran, W.G.: Sampling techniques. Wiley (2007)
20. Kleinberg, J., Tardos, E.: Algorithm design. Pearson Education (2006)
21. Durstenfeld, R.: Algorithm 235: random permutation. Commun. ACM **7**(7), 420 (1964)