Exponential Random Graph Estimation under Differential Privacy

Wentian Lu
University of Massachusetts Amherst
wen@cs.umass.edu

Gerome Miklau
University of Massachusetts Amherst
miklau@cs.umass.edu

ABSTRACT

The effective analysis of social networks and graph-structured data is often limited by the privacy concerns of individuals whose data make up these networks. Differential privacy offers individuals a rigorous and appealing guarantee of privacy. But while differentially private algorithms for computing basic graph properties have been proposed, most graph *modeling* tasks common in the data mining community cannot yet be carried out privately.

In this work we propose algorithms for privately estimating the parameters of exponential random graph models (ERGMs). We break the estimation problem into two steps: computing private sufficient statistics, then using these to estimate the model parameters. We consider recent specifications of ERGMs and show that our perturbation method, the chain mechanism, offers provably less error than comparable methods. In addition, our redesigned estimation algorithm considers the noise distribution of the private statistics and offers better accuracy than directly performing parameter estimation on the statistics.

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Security, integrity, and protection; H.2.8 [**Database Management**]: Data Mining

Keywords

Differential privacy; Exponential random graph model

1. INTRODUCTION

The explosion in the collection of networked data has fueled researchers' interest in modeling networks and predicting their behavior. However, for important application areas such as disease transmission, network vulnerability assessment, and fraud detection (among others), networks contain sensitive information about individuals and their relationships. It is difficult for institutions to release network data and it remains difficult for reseachers to acquire data in many important application domains.

Recently, a rigorous privacy standard, *differential privacy* [8] was proposed that allows for formal bounds on the disclosure about individuals that may result from computations on sensitive data. Differential privacy provides each participant in a dataset with a strong guarantee and makes no assumptions about the prior knowledge of attackers.

Since its introduction, differentially private algorithms have been developed for a wide range of data mining and analysis tasks, for both tabular data and networked data. For networks, existing work has focused on algorithms for accurately releasing common graph statistics under differential privacy [12, 16, 25, 26, 28, 31]. However, graph statistics are only one aspect of social network analysis and are often most useful in conjunction with some paradigm for modeling structural features of graphs. Privately modeling graph data has only rarely been explored by researchers; we are aware only of work using the Kronecker model [19] under differential privacy [22].

In this work, we study the differentially private use of the classic exponential random graph model (ERGM) [21, 30, 27]. ERGMs are a powerful statistical modeling tool that allows analysts to analyze a network's social structure and formation process. In social science and related fields ERGMs have been successfully applied to many scenarios, such as co-sponsorship networks [5], friendship networks [11], and corporate and inter-organizational networks [21].

Our goal is to accurately support parameter estimation for ERGMs under differential privacy, focusing on a specific set of model parameters of recent interest to researchers: the *alternating statistics*. These sophisticated statistics represent more structural information than traditional star and triangle counts, and have been shown to lead to much better modeling results [30, 27, 13, 11].

Our adaptation of differential privacy to graphs protects relationships of individuals by limiting the influence on the output of any single relationship (edge) that is created or removed from the network.[1] A standard algorithm that implements this idea is the Laplace mechanism [8], which adds random noise to the output. The amount of noise required is related to the maximum difference in the output due to a single edge addition or removal for *any* possible network

[1] This is one of the most common interpretations of differential privacy for graphs, called *edge* differential privacy [12]. *Node* differential privacy is stronger, but often hurts utility. Our results for edge-differential privacy can easily be extended to k-edge privacy to protect multiple edges.

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623683 .

(this is the *global sensitivity* of the function producing the output). For ERGM estimation, this requires calculating the exact change in the ERGM parameter estimates as a result of changing an edge. Unfortunately, the global sensitivity for most ERGM parameters is either hard to compute in general, or too high, so that using noise calibrated to the global sensitivity is not acceptable.

To overcome this obstacle, we decompose private ERGM estimation into two separate steps. We first privately compute the sufficient statistics for ERGM estimation (typically the model statistics required by model description) and then estimate the parameters using only these sufficient statistics. Since the estimation process uses only the differentially-private statistics, and there is no additional access to the original graph, the output of estimation is also differentially private. In practice, the estimation algorithm is executed either on the server side (data owner) or client side (the analyst). In either case, it does not violate the privacy condition to release both the statistics and the derived ERGM parameters.

Challenges arise in both steps of our approach. While prior work has proposed mechanisms for various graph statistics, common ERGM models use unique statistics, e.g., alternating graph statistics [30], which are a complex aggregation of a series of basic graph statistics. We propose new approaches for privately computing these statistics. The second parameter estimation step could be implemented using standard methods [29, 3] while treating the privately-computed statistics as if they were the true statistics. Instead, we propose a novel parameter estimation method based on Bayesian inference, which considers the noise distribution from which the private statistics are drawn and produces more accurate parameter estimates.

Contributions

- We propose a novel *chain mechanism* (in Section 3) that adds noise in proportion to a bound on the local (rather than global) sensitivity. Unlike global sensitivity, local sensitivity focuses only on changes to the current network. We use the chain mechanism to compute particular graph statistics (alternating k-triangle and alternating k-twopath) but it is a general technique that can be used more broadly. Compared with competing techniques that use local sensitivity, the chain mechanism is easier to deploy and more efficient.
- We provide a formal analysis of the error of the chain mechanism (in Section 3), showing that it provably outperforms prior work [16], offering lower error and a stronger privacy standard (ϵ-differential privacy, versus relaxed (ϵ, δ)-differential privacy).
- We describe a new Bayesian method for ERGM parameter estimation (in Section 4) that is designed for the noisy sufficient statistics produced by a differentially private algorithm. While it is possible to use a standard algorithm for estimation, our inference takes the unknown network as a hidden variable and can result in estimates with lower error.
- We study a set of ERGM models based on model terms consisting of alternating graph statistics [30] (in Section 5). Our experiments on both synthetic and real graphs show that our techniques significantly reduce noise over competing techniques, for fixed ϵ.

2. BACKGROUND

2.1 Exponential random graph model (ERGM)

A graph $G = (V, E)$ is defined as a set of nodes V and relationships $E : V \times V \to \{0, 1\}$. A common representation of a graph is as an adjacency matrix x, where $x_{ij} \in \{0, 1\}$ indicating whether there is an edge from node i to j. Let $f(\cdot)$ define a vector of graph statistics called the *model terms*; the concrete values of $f(x)$ are the *model statistics*. Formally, the ERGM with parameter vector θ defines a probability distribution over graphs in the space $\mathcal{X}$ (typically the set of all simple graphs with n vertices):

$$p(x|\theta) = \frac{\exp(\theta \cdot f(x))}{Z_\theta} \tag{1}$$

Z_θ is a normalizing constant to make $p(x)$ a true probability distribution, parameterized by θ. If x_0 is the observed graph and X represents the random variable defined by the distribution above, our goal is to tune the parameter vector θ, s.t. the expected value of $f(X)$ is equal to observed statistics, meaning $\mathbb{E}_\theta(f(X)) = f(x_0)$, which intuitively puts the observed graph in the "center" of space of possible graphs implied by the model. For example, the simplest ERGM uses the number of edges as the only model term. If m_0 is the total number of edges in x_0, the θ, which enables the expected number of edges of ERGM equal to m_0, is given by [24]:

$$\theta = \log \frac{m_0}{\binom{n}{2} - m_0} \tag{2}$$

Estimating θ. The optimal θ maximizes the likelihood of x_0 given θ [24], i.e., $\arg\max_\theta p(x_0|\theta)$. Unfortunately, most ERGMs do not have an analytical or closed-form estimate for the optimal θ. Thus, numerical solutions are proposed in the literature, such as Markov chain monte carlo maximum likelihood estimation [29] and Bayesian inference [3]. An interesting property of these inference methods is that the algorithm does not require access to the input graph itself, i.e., the sufficient statistics for the parameter estimation are just the model statistics. This feature enables us to decompose the private inference problem into two steps, allowing analysts to see only the sufficient statistics.

Alternating statistics. A model term is usually a counting query of a specific graph pattern. Common patterns include triangles, stars and loops [21]. Recent research has introduced alternating statistics for k-star, k-triangle and k-twopath, which can represent structural properties of a graph better than traditional star and triangle counts [30]. Many works have explored these statistics since they were proposed, and they are an active and promising form of ERGM [30, 27, 13, 11]. Our work is focused on these alternating statistics (defined precisely in Section 3) which have not been studied before under differential privacy. A wide variety of other model terms are used with ERGMs; our general approach is compatible with other terms but they are beyond the scope of this work.

2.2 Differential privacy

Differential privacy is traditionally defined over a tabular based database D consisting of records, each of which describes an individual. When querying the database, differential privacy protects individuals by restricting the impact on the output of any individual who opts into or out of the database. Two such databases that differ by one record are called *neighbors*.

Definition 2.1 (Differential Privacy[7]). Let D and D' be neighboring databases and $\mathcal{K}$ be any algorithm. For any subset of outputs $O \subseteq \mathit{Range}(\mathcal{K})$, the following holds:

$$\Pr[\mathcal{K}(D) \in O] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D') \in O] + \delta$$

If $\delta = 0$, $\mathcal{K}$ is standard ϵ-differentially private. Otherwise, $\mathcal{K}$ is relaxed (ϵ, δ)-differentially private.

The input privacy parameter ϵ (and δ if using the relaxed definition) are non-negative and are used to measure the degree of privacy protection. Smaller ϵ means better privacy as $\exp(\epsilon)$ is close to one.

In this paper, our database is a graph describing relationships among individuals. Our purpose is to protect relationships among individuals so we adapt differential privacy, following [12, 16, 26, 31, 28], by defining a neighboring graph as a graph that differs by one edge.

Global sensitivity and the Laplace mechanism

Differential privacy can be achieved by adding noise to the output of algorithms according to privacy parameters and query *sensitivity.* The global sensitivity of a query is the maximum possible difference in the output when evaluating the query on two neighboring graphs. E.g., the query asking for the maximum degree of a graph has global sensitivity 1, because adding or removing one edge changes any degree by at most 1. Let $\mathit{Lap}(b)$ be a Laplace random variable with mean 0 and scale b.

Definition 2.2 (Laplace mechanism [8]). Given query f on graph x, the following algorithm $\mathcal{K}(f, x)$ is ϵ-differentially private:

$$\mathcal{K}(f, x) = f(x) + \mathit{Lap}(\mathsf{GS}_f/\epsilon)$$

where global sensitivity

$$\mathsf{GS}_f = \max_{\forall x_1, x_2 \ \mathit{neighbors}} |f(x_1) - f(x_2)|$$

A basic property we rely on is that post-processing a noisy, differentially-private output using any algorithm that does not access the original data cannot alter the privacy guarantee [18]. Past research has shown that post-processing the noisy output can, however, have significant impact on utility. In addition, composition rules for differential privacy allow us to compute the ϵ privacy standard that results from the combined release of multiple query answers or releases. Precisely, if each release is ϵ_i-differential privacy, the combined is then $\sum_i \epsilon_i$-differential privacy.

In our perturbation step, we will use the composition rule to add noise to multiple model terms. In the parameter estimation step, we run post-processing.

Local sensitivity and its smooth bound

Some common graph analyses have high global sensitivity, requiring the Laplace mechanism to add enormous amounts of noise. For example, consider the simplest ERGM model above where θ is calculated by (2). On a graph where $m_0 = 0$ or a graph where $m_0 = \binom{n}{2}$, θ can change drastically with the addition or deletion of one edge. In other words, the global sensitivity is very high for this function. But the fact is that most real graphs are nothing like these extremes. Thus, by only focusing on the input graph's neighbors, the *local sensitivity* [25] can be much smaller.

Definition 2.3 (Local sensitivity[25]). Given query f and graph x, local sensitivity $\mathsf{LS}_f(x)$

$$\mathsf{LS}_f(x) = \max_{x, x' \mathit{neighbors}} |f(x) - f(x')|$$

However, one cannot achieve differential privacy by adding noise proportional to the local sensitivity because local sensitivity itself could disclose information. The authors of [25] proposed using a smooth upper bound on the local sensitivity, the *smooth sensitivity.* Intuitively, smooth sensitivity tries to "smooth" out the difference between local sensitivities of two neighbors, so that it is itself not sensitive. Let $d(x, x')$ be the distance between two graphs, i.e. the number of edges in which they differ:

Definition 2.4 (Smooth bound and smooth sensitivity[25]). Function $\mathsf{S}_f : \mathcal{X} \Rightarrow R$ defines a β-smooth bound of local sensitivity on query f if

$$\forall x \ : \ \mathsf{S}_f(x) \geq \mathsf{LS}_f(x)$$

$$\forall x, x' \text{ neighbors } : \ \mathsf{S}_f(x) \leq \exp(\beta) \mathsf{S}_f(x')$$

The β-smooth sensitivity of f is a β-smooth bound, and

$$\mathsf{SS}_{f,\beta}(x) = \max_{x'} \left\{ \mathsf{LS}_f(x') \cdot \exp\left(-\beta d(x, x')\right) \right\}$$

Calculating the smooth sensitivity for a function may be easy (in cases like the median of a list of numbers [25]) but could be quite difficult for other functions, requiring complex proofs and nontrivial algorithms [16]. Even though smooth sensitivity may provide tight bound for local sensitivity, we show that it is NP-hard for two alternating statistics commonly used in ERGMs.

3. PERTURBING MODEL STATISTICS

In this section we provide methods for privately computing alternating graph statistics. We show alternating k-star has a constant global sensitivity which allows the Laplace mechanism to be applied with relatively small error. However, alternating k-triangle and alternating k-twopath both have high global sensitivity. It is even hard to resort to smooth sensitivity, as calculating smooth sensitivity is NP-hard in both cases. To address this challenge, we propose the novel *chain mechanism* which allows us to efficiently bound the local sensitivity. We apply it to these two alternating statistics, but note that it is a technique that is widely applicable. We defer some proofs in this section to the Appendix.

3.1 Alternating graph statistics

Three alternating graph statistics, alternating k-star, alternating k-triangle and alternating k-twopath, are essentially complex aggregations of traditional k-star, k-triangle and k-twopath statistics. Instead of considering a vector of k terms, the alternating statistics aggregate over the terms but enforce alternating signs between each consecutive term, to weaken the correlation among different terms and effectively reduce the weight on higher terms near k.

Alternating k-star. The k-star is a counting query of a star pattern in the graph, where each star contains k edges, i.e., $S_k = \sum_i \binom{d_i}{k}$ where d_i is the degree of node i.

Definition 3.1 (Alternating k-star [30])**.** With parameter $\lambda \geq 1$, alternating k-star S is defined as

$$S(x;\lambda) = S_2 - \frac{S_3}{\lambda} + \ldots + (-1)^{n-1}\frac{S_{n-1}}{\lambda^{n-3}}$$

The λ parameter here is a good way to control the geometrical weights on all k-stars.

Alternating k-triangle. A k-triangle is a graph pattern in which k triangles share a common edge. The k-triangle query asks for the total number of k-triangles in the graph. Define the shared partner matrix C, where each entry (i, j) in C is the count of shared partners between nodes i and j, mathematically $C_{ij}(x) = \sum_l x_{il}x_{lj}$. Formally, k-triangle T_k is defined:

$$T_k = \sum_{i<j} x_{ij}\binom{C_{ij}}{k} \quad (k \geq 2), \quad \text{and } T_1 = \frac{1}{3}\sum_{i<j} x_{ij}C_{ij}$$

Alternating k-triangle is defined similarly as alternating k-star, using parameter λ:

Definition 3.2 (Alternating k-triangle [30])**.** With parameter $\lambda \geq 1$, alternating k-triangle T is:

$$T(x;\lambda) = 3T_1 - \frac{T_2}{\lambda} + \frac{T_3}{\lambda^2} - \ldots + \left(\frac{-1}{\lambda}\right)^{n-3} T_{n-2}$$

Alternating k-twopath. A k-twopath graph pattern is very similar to k-triangle, except it does not require the shared edge required by the k-triangle statistic. Using the shared partners matrix C above, the counting query for k-twopath U_k is:

$$U_k = \sum_{i<j}\binom{C_{ij}}{k} \quad (k \neq 2), \quad \text{and } U_2 = \frac{1}{2}\sum_{i<j}\binom{C_{ij}}{2}$$

And alternating k-twopath is:

Definition 3.3 (Alternating k-twopath [30])**.** With parameter $\lambda \geq 1$, alternating k-twopath U is

$$U(x;\lambda) = U_1 - \frac{2}{\lambda}U_2 + \sum_{k=3}^{n-2}\left(\frac{-1}{\lambda}\right)^{k-1} U_k$$

Alternating k-star S is the only statistic that can be readily solved using existing privacy mechanisms. Because the degree sequence is a sufficient statistic for S, one natural approach is to use the mechanism described by Hay et al [12] to compute a private degree sequence from x, and then use it to compute S by Eq. (3). But, in fact, it can be shown that the global sensitivity of S is at most 2λ. Thus, Laplace noise may be a better choice (λ is usually set to a small integer in practice). We make empirical comparisons between these methods in Section 5.

Lemma 3.4. *The global sensitivity of alternating k-star is at most 2λ.*

3.2 Chain mechanism

As the global sensitivity of alternating k-triangle and k-twopath could be as large as $O(n)$, we would like to use smooth sensitivity for perturbation. However, the following lemma shows the NP-hardness of finding the smooth sensitivity of these two statistics:

Lemma 3.5. *Computing the smooth sensitivity for both alternating k-triangle and alternating k-twopath is NP-hard.*

Inspired by the previous work [16], we should consider other techniques for bounding local sensitivity. There are two general conditions that must be satisfied if we are to use a bound on local sensitivity can be used safely (i.e., using the bound does not violate the privacy condition).

1. The bound is not smaller than the local sensitivity.

2. The bound itself is private.

In Definition 2.4 of the smooth bound, its first requirement is exactly our first condition and its second requirement satisfies the second condition here. However, the tightness of their second requirement results in the complexity of calculation of smooth sensitivity in some applications. To make the bound itself private, by relaxing the bound a bit, a simpler process can be applied. Before we introduce the chain mechanism, we define a random variable chain.

Definition 3.6. $Y_0, Y_1, \ldots, Y_n$ is a *random variable chain*, when the following condition is satisfied: for any $i \in [0, n-2]$, Y_i is conditionally independent of $Y_{i+2}, Y_{i+3}, \ldots Y_n$ given Y_{i+1}.

From conditional independence, an important property of random variable chain is the following:

$$\Pr(Y_i|Y_{i+1}, Y_{i+2}, \ldots, Y_n) = \Pr(Y_i|Y_{i+1})$$

Let $f(x)$ be the sensitive function/query. We use $\mathsf{LS}_{f,1}(x)$ to denote the local sensitivity of f, a function of the input graph x. More generally, we use $\mathsf{LS}_{f,i}(x)$ to denote the local sensitivity of function $\mathsf{LS}_{f,i-1}(x)$. We call $\mathsf{LS}_{f,0}(x)$, $\mathsf{LS}_{f,1}(x)$,$\mathsf{LS}_{f,2}(x)$,...,$\mathsf{LS}_{f,n}(x)$ *a local sensitivity chain of f.*

To satisfy condition one above, we bound the local sensitivity by adding random *positive* noise to $\mathsf{LS}_{f,1}(x)$ (recall Laplace noise is symmetric), so that we can calibrate the noise added to $f(x)$ according to that bound. The question is now how much noise should be added into $\mathsf{LS}_{f,1}(x)$, so that it is itself private, satisfying condition two above. It is actually decided by its global sensitivity $\mathsf{GS}(\mathsf{LS}_{f,1}(x))$. Moreover, if $\mathsf{GS}(\mathsf{LS}_{f,1}(x))$ is too large, we could repeat the step by adding noise to $\mathsf{LS}_{f,1}(x)$ calibrated to local sensitivity of $\mathsf{LS}_{f,1}(x)$, $\mathsf{LS}_{f,2}(x)$, and then safely bound $\mathsf{LS}_{f,2}(x)$ by its global sensitivity $\mathsf{GS}(\mathsf{LS}_{f,2}(x))$. We describe this general process as the *chain mechanism.* Let $Expn(b)$ be an exponential random variable with density function $p(y) = \frac{1}{b}\exp(\frac{-y}{b})$ for $y \geq 0$ and $p(y) = 0$ for $y < 0$.

Algorithm 1 Chain mechanism

```
Require: input graph $x$, query $f$, $\epsilon_0, \epsilon_1, \ldots, \epsilon_n$
$y_n = \mathsf{LS}_{f,n}(x) + Expn(\mathsf{GS}(\mathsf{LS}_{f,n})/\epsilon_n)$
for $i$ in $n-1$ to $1$ do
    $y_i = \mathsf{LS}_{f,i}(x) + Expn(y_{i+1}/\epsilon_i)$
$\tilde{y} = f(x) + Lap(y_1/\epsilon_0)$
return $\tilde{y}, \ldots, y_n$
```

Theorem 3.7. *Chain mechanism (Algorithm 1) is $\sum_i \epsilon_i$-differential privacy.*

In Algorithm 1, we refer to n as the size of the chain mechanism, i.e., the n-chain mechanism. Note that the chain mechanism does not specify how to distribute the privacy parameters ϵ_i among steps. An optimal distribution will require the knowledge of local sensitivity chain, which is sensitive information itself. A simple way is to distribute ϵ evenly, i.e., $\forall i\ \epsilon_i = \epsilon/(n+1)$.

Error analysis. We use mean squared error (MSE) as the measurement of error. MSE of $\tilde{y}$ in Algorithm 1 can be written as $\mathbb{E}[(\tilde{y} - f(x))^2] = \mathbb{V}[\tilde{y}] + (\mathbb{E}[\tilde{y}] - f(x))^2$. Since $\tilde{y}$ is always unbiased (Laplace noise in the last step with mean zero), $\text{MSE}(x) = \mathbb{V}[\tilde{y}]$. It is easy to see that $\tilde{y}, y_1, \ldots, y_n$ is actually a random variable chain. Specifically,

$$y_i - \mathsf{LS}_{f,i}(x) \sim Expn(y_{i+1}/\epsilon_i)\Big|y_{i+1},\ \forall i \in [1, n-1]$$

$$\tilde{y} - f(x) \sim Lap(y_1/\epsilon_0)\Big|y_1$$

Without knowing the true value of the local sensitivity chain, it is quite hard to compute the MSE. That is to say, we cannot compute the error of the chain mechanism like we do for the Laplace mechanism, since the noise in the latter is independent of input graph x. But, by exploring properties of the random variable chain, it is possible to utilize the following Lemma as a closed form calculation tool for the MSE of the chain mechanism. In fact, we generalize law of total expectation/variance [32] for random variable chains.

Lemma 3.8. *$Y_0, Y_1, \ldots, Y_n$ is a random variable chain. Write $\bigsqcup_{n,i} \mathbb{E}[\cdot]$ as a shortcut for $\mathbb{E}_{Y_n}[\mathbb{E}_{Y_{n-1}|Y_n}[\cdots \mathbb{E}_{Y_i|Y_{i+1}}[\cdot]]]$. Then*

$$\mathbb{E}[Y_0] = \bigsqcup_{n,0} \mathbb{E}[Y_0]$$

$$\mathbb{V}[Y_0] = \bigsqcup_{n,1} \mathbb{E}[\underset{Y_0|Y_1}{\mathbb{V}}[Y_0]] + \sum_{i=2}^{n-2} \left(\bigsqcup_{n,i} \mathbb{E}[\underset{Y_{i-1}|Y_i}{\mathbb{V}}[\bigsqcup_{i-2,0} \mathbb{E}[Y_0]]] \right) + \underset{Y_n}{\mathbb{V}}[\bigsqcup_{n-1,0} \mathbb{E}[Y_0]]$$

Theorem 3.9 (MSE of chain mechanism)**.** *Given the output of an n-chain mechanism, $\tilde{y}, y_1, \ldots, y_n$, the mean squared error (MSE) is*

$$\text{MSE}_{f,n} = \bigsqcup_{n,1} \mathbb{E}[\underset{\tilde{y}|y_1}{\mathbb{V}}[\tilde{y}]]$$

Let $l_1, \ldots, l_n$ be the local sensitivity chain, with $l_i = \mathsf{LS}_{f,i}(x)$. Writing $\mathsf{GS}(\mathsf{LS}_{f,n}(x))$ as g_n, we have:

$$\text{MSE}_{f,1} = \frac{2}{\epsilon_1^2 \epsilon_0^2} \left[g_1^2 + (l_1\epsilon_1 + g_1)^2 \right] \tag{3}$$

$$\text{MSE}_{f,2} = \frac{2}{\epsilon_2^2 \epsilon_1^2 \epsilon_0^2} \left[2g_2^2 + (l_2\epsilon_2 + g_2)^2 + (l_2\epsilon_2 + l_1\epsilon_2\epsilon_1 + g_2)^2 \right] \tag{4}$$

Theorem 3.9 is verified by applying Lemma 3.8 to Algorithm 1. The general MSE for any size n chain mechanism can be written as a lengthy closed-form equation, which we omit here. In experiments, $n = 2$ performs well for alternating statistics. In general, n should not be a very big number otherwise each part gets smaller ϵ_i, but users should also consider the difficulty of computing the local sensitivity chain, as well as the amount of global sensitivity at the tail (it should not be too large).

Comparison with RLSB [16]. Karwa et al. proposed a similar idea for bounding local sensitivity by adding specially generated noise [16]. We call their technique relaxed local sensitivity bounding (RLSB) because their bound does not strictly satisfy the first condition above for safely bounding local sensitivity. Consequently, only relaxed (ϵ, δ)-differential privacy is supported. More importantly, because they add too much noise to the local sensitivity, our work, which supports stronger privacy definition (ϵ-differential privacy), still offers better utility, as stated in Lemma 3.10.

Lemma 3.10. *Assuming ϵ is evenly distributed, for any function f, chain mechanism offers lower mean squared error than RLSB.*

3.3 Alternating k-triangle and k-twopath

Now we apply the chain mechanism to alternating k-triangle and alternating k-twopath. Let $\beta = 1 - 1/\lambda$. By binomial coefficients, we can rewrite alternating k-triangle $T(x;\lambda)$ as

$$T(x;\lambda) = \lambda \sum_{i<j} x_{ij} \left\{ 1 - \beta^{C_{ij}} \right\} \tag{5}$$

Lemma 3.11. *Set $C'_{iv} = C_{iv} - x_{ij}$ and $C'_{vj} = C_{vj} - x_{ij}$. Let N_{ij} be all shared partners of node i and j and $C_{max} = \max_{i<j} C_{ij}$. The local sensitivity of T is*

$$\mathsf{LS}_{T,1}(x) = \max_{i<j}\ \lambda \left\{ 1 - \beta^{C_{ij}} \right\} + \sum_{v \in N_{ij}} \left\{ \beta^{C'_{iv}} + \beta^{C'_{vj}} \right\} \tag{6}$$

$$\leq \lambda + 2C_{max} \tag{7}$$

As C_{max} has global sensitivity 1, $\mathsf{LS}_{T,1}$ has global sensitivity at most 2. So we can construct a 1-chain mechanism using $\mathsf{LS}_{T,1} = \lambda + 2C_{max}$ to compute private alternating k-triangle.

For alternating k-twopath $U(x;\lambda)$, we can rewrite it as

$$U(x;\lambda) = \lambda \sum_{i<j} \left\{ 1 - \beta^{C_{ij}} \right\} \tag{8}$$

Lemma 3.12. *Let N_i be the set of neighbors of node i and d_{max} be the maximum degree. Set $C'_{iv} = C_{iv} - x_{ij}$ and $C'_{vj} = C_{vj} - x_{ij}$. We have local sensitivity*

$$\mathsf{LS}_{U,1}(x) = \max_{i<j} \left\{ \sum_{v \in N_i, v \neq j} \beta^{C'_{vj}} + \sum_{v \in N_j, v \neq i} \beta^{C'_{iv}} \right\} \tag{9}$$

$$\leq 2d_{max} \tag{10}$$

$$\mathsf{LS}_{U,2}(x) \leq \frac{\max(4, 1 + C_{max})}{\lambda} \tag{11}$$

From Lemma 3.12 above, $2d_{max}$ has global sensitivity 2, since d_{max} will change by at most 1 by adding or removing an edge. $\frac{\max(4,1+C_{max})}{\lambda}$ has global sensitivity $1/\lambda$ for $C_{max} > 3$. Therefore, we can construct either a 1-chain or 2-chain mechanism. We will compare the resulting error empirically in Section 5.

4. ERGM PARAMETER ESTIMATION

The parameter estimation step in our workflow takes the private sufficient statistics $\tilde{y}$ from the previous perturbation step and finds the best parameter vector θ. As stated above, this step is essentially post-processing a differentially private output, so the output θ is also differentially private. In this section, we discuss different ways of estimating θ given $\tilde{y}$.

4.1 Standard estimation

Current estimation techniques [29, 3] provide a baseline solution for parameter estimation with private statistics. As these procedures essentially only need access to model statistics, our sufficient statistics in $\tilde{y}$ take the place of the true model terms. The semantics is now to search for θ that defines a probability distribution on graphs with expected model statistics equal to $\tilde{y}$. Intuitively, the utility of this method depends on the amount of noise added into y_0 and how θ reacts to those changes in y_0.

Prior to applying standard estimation, we post-process $\tilde{y}$ to cope with some of the difficulties of the perturbed model statistics. As the output of perturbed $\tilde{y}$ might not be *graphical* (i.e., no graph has statistics equal to $\tilde{y}$), standard estimation may fail to converge. We propose generating a graph that has the closest statistics to $\tilde{y}$ and use the statistics from that graph to replace $\tilde{y}$, in order to avoid non-converging situations and to potentially remove noise from $\tilde{y}$ simultaneously. We use simulated annealing for this purpose and, in practice, we often see big improvements in the accuracy of estimates.

4.2 Bayesian inference

Standard estimation is the direct way of post-processing $\tilde{y}$, but since we know the distribution of the noise added to $\tilde{y}$, we can "guess" the true values and incorporate them into the estimation algorithm. This idea naturally fits into *Bayesian inference* based post-processing. While based on earlier work [3] on Bayesian inference for non-private estimation, our method deals with the extra hidden variable of graph x in our setting. And later we will see, by introducing the unknown x, our method can utilize more information from private statistics, such as the local sensitivity chain from the chain mechanism. In particular, we search for θ given $\tilde{y}$, represented as the posterior distribution of ERGM parameter θ:

$$\begin{aligned} p(\theta|\tilde{y}) \propto p(\tilde{y}|\theta)p(\theta) &= \sum_x p(\tilde{y}|x)p(x|\theta)p(\theta) \\ &= \sum_x p(\tilde{y}|x)q(x;\theta)p(\theta)/Z_\theta \end{aligned} \tag{12}$$

where x is our guess about x_0, but the fact is that we need to summarize over all possible x to get to the posterior. In (12), $p(\tilde{y}|x)$ is the *privacy distribution*, defined by the differential privacy mechanism applied on sufficient statistics. $p(x|\theta)$ is the *ERGM distribution*, as shown in (1) and $q(x;\theta)$ represents the unnormalized distribution.

$$q(x;\theta) = \exp(\theta \cdot f(x)) \tag{13}$$

The probability distribution (12) is hard to calculate or even sample from directly due to summarization over all graphs and normalizing constant Z_θ. Using the exchange algorithm [23], we introduce extra variables x, θ' and x' to bypass the difficult terms (12). By carefully choosing the probability distribution of these new random variables, the posterior distribution is now augmented as shown in (14). The key is that the marginal posterior distribution for θ in (14) is equivalent to (12). Thus, if we are able to sample from the distribution in (14), the marginal posterior distribution for θ can be obtained by summarizing over all samples.

$$p(\theta, x, \theta', x'|\tilde{y}) \propto p(\tilde{y}|x)p(x|\theta)p(\theta)p(\theta'|\theta)p(x'|\theta') \tag{14}$$

θ' is sampled from *proposal distribution* $p(\theta'|\theta)$, where, for a given θ, a new θ' can be proposed according to $p(\theta'|\theta)$. A common choice is a multivariate normal distribution or a multivariate t distribution, with mean equal to θ. x, x' are sampled graphs under the ERGM with parameter θ and θ'.

Algorithm 2 ERGM parameter estimation with private model statistics

```
Require: $\tilde{y}$, initial $\theta, x$
1: for $i$ in 1 to $T$ do
2:     Sample $\theta' \sim p(\theta'|\theta)$
3:     Sample $x' \sim p(x'|\theta')$
4:     Replace $\theta$ with $\theta'$ and $x$ with $x'$, with probability
       $\min(1, H)$    $//H$ by (15) below
5: return average of multiple samples of $\theta$.
```

A MCMC based sampling process for (14) is shown in Algorithm 2. In particular, the initial input θ and x could be any parameters and any graph. In Line 3, we need a separated MCMC chain to sample $x' \sim p(x'|\theta')$. In such MCMC algorithms, at each iteration, we propose adding or removing edges in the current state of graph, calculate the new model statistics, compare the probability of new state x_{new} to that of old state x_{old}, and with probability $p(x_{new}|\theta')/p(x_{old}|\theta')$ the change is accepted. This process should be run long enough so that final sample x' is truly from $p(x'|\theta')$.

H in Line 4 is the ratio of accepting the exchange, computed by comparing the probability before and after exchange. That is, we exchange θ with θ' and x with x' in (14) and calculate the ratio. Then the complex terms are cancelled out and each remaining term is easy to compute.

$$\begin{aligned} H &= \frac{p(\tilde{y}|x')p(x'|\theta')p(\theta')p(\theta|\theta')p(x|\theta)}{p(\tilde{y}|x)p(x|\theta)p(\theta)p(\theta'|\theta)p(x'|\theta')} \\ &= \frac{p(\tilde{y}|x')p(\theta')p(\theta|\theta')}{p(\tilde{y}|x)p(\theta)p(\theta'|\theta)} \end{aligned} \tag{15}$$

In practice, Algorithm 2 usually results in low acceptance rates in the exchange step in Line 4 and thus long mixing times for the MCMC process. We now propose to separate that last step, isolating simultaneously updated θ and x into two different steps, as shown in Algorithm 3, which improves the acceptance rate significantly.

Algorithm 3 Improved ERGM parameter estimation with private model statistics

```
Require: $\tilde{y}$, initial $\theta, x$
1: for $i$ in 1 to $T$ do
2:     Sample $\theta' \sim p(\theta'|\theta)$
3:     Sample $x' \sim p(x'|\theta')$
4:     Exchange $\theta$ with $\theta'$, with probability $\min(1, H_1)$
       $//H_1$ by (16) below
5:     Replace $x$ with $x'$, with probability $\min(1, H_2)$
       $//H_2$ by (17) below
6: return average of multiple samples of $\theta$.
```

H_1 and H_2 in Algorithm 3 are defined as follows.

$$\begin{aligned} H_1 &= \frac{p(\tilde{y}|x)p(x|\theta')p(\theta')p(\theta|\theta')p(x'|\theta)}{p(\tilde{y}|x)p(x|\theta)p(\theta)p(\theta'|\theta)p(x'|\theta')} \\ &= \frac{q(x;\theta')p(\theta')p(\theta|\theta')q(x';\theta)}{q(x;\theta)p(\theta)p(\theta'|\theta)q(x';\theta')} \end{aligned} \tag{16}$$

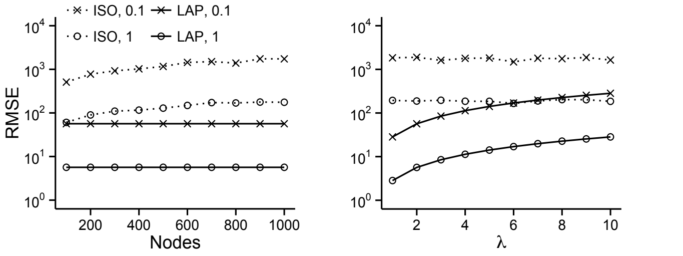


Figure 1: Perturbation error on alternating k-star on synthetic graphs. Left: $p = \log(n)/n$, varying size of graph n. Right: $n = 1000$, varying λ.

$$H_2 = \frac{p(\tilde{y}|x')p(x'|\theta)p(\theta)p(\theta'|\theta)p(x|\theta')}{p(\tilde{y}|x)p(x|\theta)p(\theta)p(\theta'|\theta)p(x'|\theta')}$$
$$= \frac{p(\tilde{y}|x')q(x';\theta)q(x;\theta')}{p(\tilde{y}|x)q(x;\theta)q(x';\theta')} \qquad (17)$$

The correctness of Algorithm 3 can be proved briefly in terms of a component-wise Metropolis-Hasting algorithm, with hybrid Gibbs updating steps. In each iteration, θ' and x' (Line 2 and 3) are drawn based on full conditional distribution, so the updating probability is always 1. In Line 4 and 5, we update θ and x with Hasting ratios. Although we may end up updating θ' and x' more times in a iteration, we still get to the detailed balance in MCMC [9].

When applying Algorithm 3 to real ERGM models, the key is correctly computing H_1 and H_2. Everything in H_1 is independent of the privacy mechanism used for the model terms. In H_2, the ratio of privacy distribution $\frac{p(\tilde{y}|x')}{p(\tilde{y}|x)}$ is mechanism dependent. Here, we illustrate the cases for both Laplace and Chain mechanism.

Example 4.1 (Laplace mechanism)**.** If the Laplace mechanism is applied on all model terms (f_i for i-th model term) independently, and $\tilde{y}$, ϵ and GS are the vectors of private statistics, privacy parameters and global sensitivities respectively, $p(\tilde{y}|x)$ is then:

$$p(\tilde{y}|x) \propto \exp\left(-\sum_i |\tilde{y}_i - f_i(x)|\epsilon_i/\mathsf{GS}_i\right) \qquad (18)$$

Assume we use a symmetric proposal distribution for θ, i.e., $p(\theta'|\theta) = p(\theta|\theta')$. With Algorithm 3, ratio H_1 and H_2 can be written as (after taking logarithm)

$$\log H_1 = \log\frac{p(\theta')}{p(\theta)} + (\theta - \theta')\cdot\big(f(x') - f(x)\big) \qquad (19)$$
$$\log H_2 = (\theta - \theta')\cdot\big(f(x') - f(x)\big) +$$
$$\sum_i \frac{\epsilon_i}{\mathsf{GS}_i}\big(|\tilde{y}_i - f_i(x)| - |\tilde{y}_i - f_i(x')|\big) \qquad (20)$$

Example 4.2 (Chain mechanism)**.** Assume a single model term (multiple model terms can be adjusted accordingly), and privacy parameter ϵ. In the process of MCMC, for current sampled graph x, we write $l_1, \ldots, l_n$ as the local sensitivity chain and y_{n+1} for the global sensitivity of l_n. The chain mechanism returns $\tilde{y}, y_1, \ldots, y_n$ for the observed graph. Based on Algorithm 1, $p(\tilde{y}|x)$ is then:

$$p(\tilde{y}|x) = p(\tilde{y}|x, y_1)p(y_1|y_2, l_1)\ldots p(y_n|y_{n+1}, l_n)$$
$$\propto \exp\left(\sum_{i\in[1,n]} \frac{l_i - y_i}{y_{i+1}/\epsilon_i} - \frac{|\tilde{y} - f(x)|}{y_1/\epsilon_0}\right) \qquad (21)$$

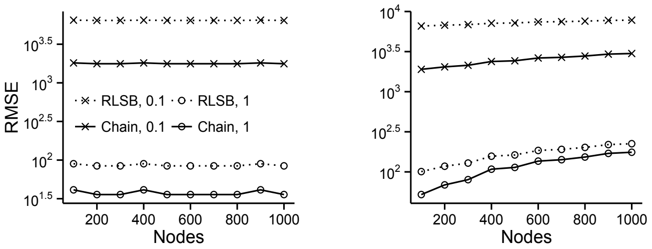


Figure 2: Perturbation error on alternating k-triangle. Left: $p = \log(n)/n$. Right: $p = 0.1$.

Calculation of $p(\tilde{y}|x)$ deals with not only the private version of local sensitivity chain $(y_1, \ldots, y_n)$, but also more statistics from the sampled graph in each iteration of MCMC $(l_1, \ldots, l_n)$. Recall in the standard estimation, none of them is incorporated in the process. In the next section, we empirically show that such extra information can benefit the estimation. As in the example above, assume a symmetric proposal distribution. With Algorithm 3, ratio H_1 is the same as (19). Before calculating H_2, we first check if all $l_i \leq y_i$, otherwise $H_2 = 0$, because exponential noise in the chain mechanism is non-negative. If the check passes, H_2 is:

$$\log H_2 = (\theta - \theta')\cdot\big(f(x') - f(x)\big) +$$
$$\sum_{i\in[1,n]} \frac{l'_i - l_i}{y_{i+1}/\epsilon_i} + \frac{|\tilde{y} - f(x)| - |\tilde{y} - f(x')|}{y_1/\epsilon_0} \qquad (22)$$

Marginal maximum a posterior. In practice, instead of returning the mean of the marginal posterior (14), using a marginal maximum a posterior (MMAP) could give analysts better estimates. Formally, MMAP of θ is defined as $\underset{\theta}{\operatorname{argmax}}\ p(\theta|\tilde{y})$. A fast method we apply is reusing the samples of θ from Algorithm 3, and performing approximate MMAP estimation by histogram or density estimation. More sophisticated solutions require further expanding (14) before MCMC sampling [6, 15].

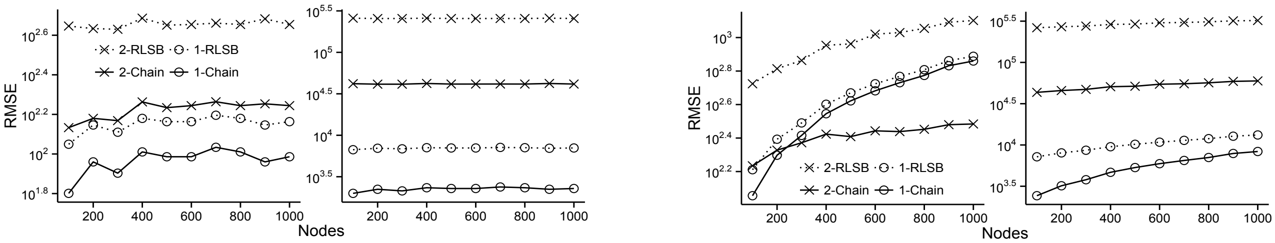
5. EVALUATION

Our evaluation has two goals. First we assess the perturbation error of our privacy mechanisms, particularly the Laplace mechanism on alternating k-star and the Chain mechanism on alternating k-triangle and k-twopath. Second, we evaluate the ERGM parameter estimation with private statistics using different approaches proposed in Section 4. All our experiments are run on Linux servers with Intel Xeon CPU and 8GB memory.

5.1 Perturbation error

Our datasets include synthetic and real graphs. Synthetic graphs are generated using a random graph model, $G(n, p)$, where parameters n and p control the size of graph and the probability of two nodes connecting respectively. We iterate n from 100 to 1000 with step 100. p is set to $\log(n)/n$ for relatively sparse graphs and then moved to 0.1 and higher by a step of 0.1. Though we only report the sparse case and $p = 0.1$, results for larger p agree with the conclusions. Error measurement is root mean square error (RMSE).

Alternating k-star As described in Section 3.1, we can apply the Laplace mechanism (LAP) directly or compute the degree distribution privately first, by isotonic regression



(1) $p = \log(n)/n, \epsilon = 1$ (2) $p = \log(n)/n, \epsilon = 0.1$ (3) $p = 0.1, \epsilon = 1$ (4) $p = 0.1, \epsilon = 0.1$

Figure 3: Perturbation error on alternating k-twopath.

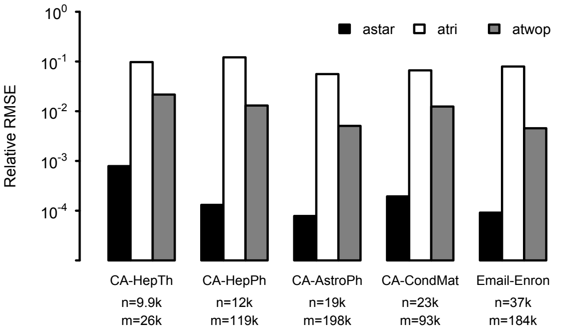


Figure 4: Perturbation error on real graphs

Network	nodes	edges	astar	atri	atwop
karate	34	78	194.0	88.7	411.7
dolphins	62	159	418.1	177.5	705.4
lesmis	77	254	756.4	426.5	1565.5
adjnoun	112	425	1292.9	452.2	3801.1
football	115	613	1992.4	922.4	3675.4

Table 1: Real networks for ERGM parameter estimation

Model	Model terms	Perturbation mech
M1	edges, astar	LAP, LAP
M2	edges, atri	LAP, 1-Chain
M3	edges, atwop	LAP, 1-Chain

Table 2: Model descriptions

(ISO) from [12] and use it as a sufficient statistic for alternating k-star. Figure 1 shows the error of the two methods by varying p and λ, with different settings of $\epsilon = 1, 0.1$, listed in the legend text. As we do not have analytical RMSE for the ISO case, it is calculated from 100 independent perturbations. We clearly see LAP significantly outperforms ISO, even when $\lambda = 10$ at both ϵ settings (and recall that the global sensitivity is 2λ). For the rest of this section, if not stated, we set $\lambda = 2$ as it is the value normally recommended [21] and usually plays a minor part in the workflow.

Alternating k-triangle The chain mechanism is applied to alternating k-triangle, with a size-one chain. We compare with RLSB [16], setting ϵ to 1 and 0.1 and fixing $\delta = 0.01$ for RLSB. Figure 2 shows that the Chain is constantly better, as stated by Lemma 3.10. Moreover, with smaller ϵ (meaning greater privacy protection), Chain can gain even more advantage over RLSB, not to mention that RLSB only offers relaxed differential privacy.

Alternating k-twopath We discussed in Section 3.3 how a 1-chain or 2-chain can be used for alternating k-twopath. Using RLSB's noise generation, we can bound local sensitivity with its global sensitivity (1-RLSB), or treating C_{max} as the deciding factor of local sensitivity and bounding it first (2-RLSB). We present results in Figure 3. As above, in all circumstances, Chain illustrates superior utility over RLSB and the difference is even more drastic in the 2-chain case. This is because RLSB adds more noise than necessary in each step of the local sensitivity chain, which accumulates in the final output. If the local sensitivity chain size is larger than 2, there will be an even greater difference. In Figure 3, with $\lambda = 2$, 1-chain generates less error than 2-chain on most cases, except on random graphs with $p = 0.1$ and $n > 300$. When increasing λ, 2-chain benefits from shrinked $\mathsf{LS}_{U.2}$. At small $\epsilon = 0.1$, with random graphs with $n = 1000$ and $p = 0.1$, 2-chain needs roughly $\lambda = 10$ to surpass 1-chain, which happens when Eq. (9) is much smaller than Eq. (10).

Real graphs For real graphs, we consider several collected networks from the SNAP collection[2] in order to figure out if our alternating statistics can be perturbed in a "meaningful" way, i.e., small relative noise that doesn't destroy the utility. Our metric is relative RMSE, which is RMSE divided by the true statistic. As shown in Figure 4, with $\epsilon = 0.1$, all three alternating statistics (with shortened names: astar, atri, atwop) are estimated with low relative error. In particular, error for alternating k-star is between 10^{-3} and 10^{-4}, alternating k-triangle at 10^{-1} and alternating k-twopath at 10^{-2}.

5.2 ERGM parameter estimation

For the evaluation of ERGM parameter estimation, we want to compare the algorithms in Section 4. In practice, the data owner will only perturb each statistic once and then release it to the analysts. As the perturbation is a randomized process, our goal is to understand how good our estimation algorithm is on *average*. So for each graph and each model description, we perturb the statistics $N = 50$ times and run the estimation algorithm on each perturbation, finally measuring their quality by RMSE with respect to estimates in the non-private case, $\sqrt{1/N \sum_{i \in [1,N]} (\hat{\theta}_i - \theta)^2}$, where θ is the "true" value, calculated from the non-private estimation algorithm from [14] or [3], $\hat{\theta}_i$ is θ from i-th perturbation.

As mentioned in [3], the estimation using the Bayesian technique has general scalability issues, where it becomes very slow for any graphs beyond a few hundred of nodes. Moreover such time cost also varies with the model terms,

[2] http://snap.stanford.edu

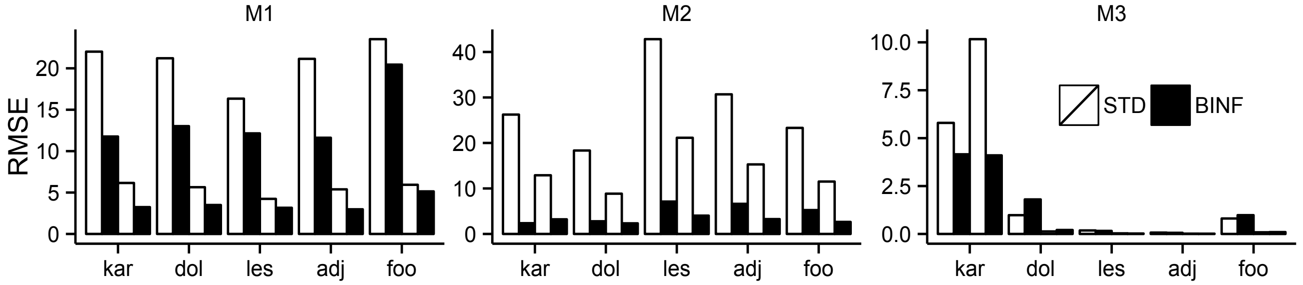


Figure 5: Parameter estimation with private statistics. Every four bars, from left to right, are $\theta_1, \theta_1, \theta_2, \theta_2$.

e.g., alternating k-twopath takes much more time than the other two alternating statistics, as calculation of the acceptance ratio in MCMC sampling of $x \sim p(x|\theta)$ is more complicated. Therefore, here we focus on smaller graphs, and this is the common practice for many ERGM works such as [3, 5, 21]. Our test networks[3] include karate, dolphins, lesmis, adjnoun and football. Detailed facts are listed in Table 1. We fix $\epsilon = 1$.

We experimented with three models, each of which corresponds to one of the alternating statistics, with the purpose of testing estimation by isolating other factors. We include the count of edges as a shared term in all models, as it is very common in ERGM applications. As shown in Table 2, each model contains two terms, with correspondingly two parameters, $\theta = (\theta_1, \theta_2)$. The estimation algorithms will be standard estimation (STD) and Bayesian inference (BINF). In all cases, the privacy budget is distributed evenly in a way such that each generation of noise uses same share of the overall ϵ. In Figure 5, each graph is represented with 4 bars, showing θ_1 of STD, θ_1 of BINF, θ_2 of STD, θ_2 of BINF. In M1 and M2, we see a significant improvement of θ from STD to BINF. Especially in M2, BINF limits all errors to around 5 or smaller where STD can go up to 40. We believe this is because BINF can utilize the extra information presented by the local sensitivity chain as shown in Example 4.2. In M3, we find BINF helps a lot on the bad case (karate graph) but not really on others as they already have low error in STD.

6. RELATED WORK

Differential privacy [8] has been actively studied in many sub-areas of computer science. Although the original focus was mainly on tabular data, the definition can be adapted to graph data [12] as well as other data models. Most research into differentially private analysis of graphs has focused on releasing graph statistics, e.g., degree sequence [12], triangle/star [16, 25], joint degree distribution/assortativity [26, 28] and clustering coefficient [31]. For modeling graphs privately, we are aware only of a private Kronecker graph modeling approach under differential privacy [22]. While our work relies on obtaining good private statistics, the ultimate goal is to allow ERGM modeling under differential privacy.

All of these works, including ours, protect relationships, i.e. they support edge-differential privacy. A stronger standard is to protect individuals, where neighbors are defined by changing a single node. Recently, researchers have developed some mechanisms for calculating private graph statistics under node differential privacy [17, 2, 4]. Our chain mechanism could be a good supporting algorithm when bounds on local sensitivity are required [17, 2].

Parameter estimation for ERGMs has also evolved from pseudo likelihood estimation (MPLE) [1], to Monte Carlo maximum likelihood (MC-MLE) [10] to recent stochastic approximation [29] and Bayesian inference [3]. These advances have helped ERGMs become central to social network analysis with many successful applications [21].

7. CONCLUSION AND FUTURE WORK

In this work, we consider the problem of estimating parameters for the exponential random graph model under differential privacy. Our solution decomposes the process into two steps: releasing private statistics first and running estimation second. Our local sensitivity-based chain mechanism can offer lower error than existing methods. The redesigned Bayesian parameter estimation is flexible and more accurate than standard methods. For future work, improving scalability is an important direction as well exploring alternative model terms.

8. ACKNOWLEDGMENTS

This material is based on work supported by the NSF through awards CNS-1012748 and IIS-0964094.

9. REFERENCES

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1974.

[2] J. Blocki, A. Blum, A. Datta, and O. Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *ITCS*, 2013.

[3] A. Caimo and N. Friel. Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55, 2011.

[4] S. Chen and S. Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *SIGMOD*, 2013.

[5] S. J. Cranmer and B. A. Desmarais. Inferential network analysis with exponential random graph models. *Political Analysis*, 19(1):66–86, 2011.

[6] A. Doucet, S. J. Godsill, and C. P. Robert. Marginal maximum a posteriori estimation using markov chain monte carlo. *Statistics and Computing*, 2002.

[3] http://www-personal.umich.edu/˜mejn/netdata/

[7] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *EUROCRYPT*, 2006.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, 2006.

[9] D. Gamerman and H. F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*, volume 68. CRC Press, 2006.

[10] C. J. Geyer and E. A. Thompson. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society.*, 1992.

[11] S. M. Goodreau, J. A. Kitts, and M. Morris. Birds of a feather, or friend of a friend? using exponential random graph models to investigate adolescent social networks*. *Demography*, 46(1):103–125, 2009.

[12] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, 2009.

[13] D. Hunter. Curved exponential family models for social networks. *Social Networks*, 29(2):216–230, 2007.

[14] D. Hunter and M. Handcock. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 2006.

[15] A. M. Johansen, A. Doucet, and M. Davy. Particle methods for maximum likelihood estimation in latent variable models. *Statistics and Computing*, 2008.

[16] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev. Private analysis of graph structure. In *VLDB*, 2011.

[17] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In *TCC*, 2013.

[18] D. Kifer and B.-R. Lin. An axiomatic view of statistical privacy and utility. *Journal of Privacy and Confidentiality*, 2012.

[19] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *JMLR*, 2010.

[20] W. Lu and G. Miklau. Exponential random graph estimation under differential privacy. In *arXiv*, 2014.

[21] D. Lusher, J. Koskinen, and G. Robins. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 2012.

[22] D. Mir and R. N. Wright. A differentially private estimator for the stochastic kronecker graph model. In *EDBT/ICDT Workshops*, 2012.

[23] I. Murray, Z. Ghahramani, and D. MacKay. Mcmc for doubly-intractable distributions. *arXiv*, 2012.

[24] M. Newman. *Networks: an introduction*. Oxford University Press, 2010.

[25] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, 2007.

[26] D. Proserpio, S. Goldberg, and F. McSherry. Calibrating data to sensitivity in private data analysis. In *VLDB*, 2014.

[27] G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison. Recent developments in exponential random graph p* models for social networks. *Social networks*, 2007.

[28] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Zhao. Sharing graphs using differentially private graph models. In *SIGCOMM*, 2011.

[29] T. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 2002.

[30] T. A. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock. New specifications for exponential random graph models. *Sociological methodology*, 2006.

[31] Y. Wang, X. Wu, J. Zhu, and Y. Xiang. On learning cluster coefficient of private networks. *Social network analysis and mining*, 2013.

[32] N. A. Weiss, P. T. Holmes, and M. Hardy. *A course in probability*. Pearson Addison Wesley, 2006.

APPENDIX

We provide other proofs in the full version of this paper [20].

Proof of Theorem 3.7. Let $\tilde{Y}, Y_1, \ldots, Y_n$ be the random variables representing output of chain mechanism. By the chain property of Y_i series (i.e., Y_i is conditionally independent of $Y_{i+2}, \ldots, Y_n$ given Y_{i+1}), we have:

$$\begin{aligned} &\Pr(\tilde{Y} = \tilde{y}, Y_1 = y_1, \ldots, Y_n = y_n) \\ =&\Pr(\tilde{Y} = \tilde{y} | Y_1 = y_1) \Pr(Y_1 = y_1 | Y_2 = y_2) \\ &\ldots \Pr(Y_{n-1} = y_{n-1} | Y_n = y_n) \Pr(Y_n = y_n) \end{aligned}$$

Now we want to prove for each combination of $y_1, \ldots, y_n$, the multiplicity of probabilities above satisfies differential privacy. More precisely, each $\Pr(Y_i = y_i | Y_{i+1} = y_{i+1})$ offers ϵ_i-differential privacy.

We start with $i = n$. Let $\Pr'(\cdot)$ represent the probability on neighbor x' and $g = \mathsf{GS}(\mathsf{LS}_{f,n}(x))$

$$\begin{aligned} \frac{\Pr(Y_n = y_n)}{\Pr'(Y_n = y_n)} &= \frac{\epsilon_n/g \ \exp(-(y_n - \mathsf{LS}_{f,n}(x)) * \epsilon_n/g)}{\epsilon_n/g \ \exp(-(y_n - \mathsf{LS}_{f,n}(x')) * \epsilon_n/g)} \\ &= \exp\left(\frac{(\mathsf{LS}_{f,n}(x) - \mathsf{LS}_{f,n}(x')) * \epsilon_n}{g}\right) \end{aligned}$$

Because $-g \le \mathsf{LS}_{f,n}(x) - \mathsf{LS}_{f,n}(x') \le g$, so

$$\exp(-\epsilon_n) \le \frac{\Pr(Y_n = y_n)}{\Pr'(Y_n = y_n)} \le \exp(\epsilon_n)$$

We move to $\Pr(Y_i = y_i | Y_{i+1} = y_{i+1})$ when $i < n$.

$$\begin{aligned} &\frac{\Pr(Y_i = y_i | Y_{i+1} = y_{i+1})}{\Pr'(Y_i = y_i | Y_{i+1} = y_{i+1})} \\ =&\frac{\epsilon_i/y_{i+1} \ \exp(-(y_i - \mathsf{LS}_{f,i}(x)) * \epsilon_i/y_{i+1})}{\epsilon_i/y_{i+1} \ \exp(-(y_i - \mathsf{LS}_{f,i}(x')) * \epsilon_i/y_{i+1})} \\ =&\exp\left(\frac{(\mathsf{LS}_{f,i}(x) - \mathsf{LS}_{f,i}(x')) * \epsilon_i}{y_{i+1}}\right) \end{aligned}$$

Because $-\mathsf{LS}_{f,i+1}(x) \le \mathsf{LS}_{f,i}(x) - \mathsf{LS}_{f,i}(x') \le \mathsf{LS}_{f,i+1}(x)$ and $y_{i+1} \ge \mathsf{LS}_{f,i+1}$ (due to positive exponential noise), we have:

$$\exp(-\epsilon_i) \le \frac{\Pr(Y_i = y_i | Y_{i+1} = y_{i+1})}{\Pr'(Y_i = y_i | Y_{i+1} = y_{i+1})} \le \exp(\epsilon_i)$$

Similarly, line 4 in Algorithm 1 is ϵ_0-differential private. Therefore, the whole algorithm is $\sum_i \epsilon_i$-differential private. □