

---

# Differentially Private Learning of Undirected Graphical Models Using Collective Graphical Models

---

Garrett Bernstein<sup>1</sup> Ryan McKenna<sup>1</sup> Tao Sun<sup>1</sup> Daniel Sheldon<sup>1,2</sup> Michael Hay<sup>3</sup> Gerome Miklau<sup>1</sup>

## Abstract

We investigate the problem of learning discrete, undirected graphical models in a differentially private way. We show that the approach of releasing noisy sufficient statistics using the Laplace mechanism achieves a good trade-off between privacy, utility, and practicality. A naive learning algorithm that uses the noisy sufficient statistics “as is” outperforms general-purpose differentially private learning algorithms. However, it has three limitations: it ignores knowledge about the data generating process, rests on uncertain theoretical foundations, and exhibits certain pathologies. We develop a more principled approach that applies the formalism of collective graphical models to perform inference over the true sufficient statistics within an expectation-maximization framework. We show that this learns better models than competing approaches on both synthetic data and on real human mobility data used as a case study.

## 1. Introduction

Graphical models are a central tool in probabilistic modeling and machine learning. They pair expressive probability models with algorithms that leverage the graphical structure for efficient inference and learning. However, with data collection and modeling growing in importance in nearly all domains of society, there is increasing demand to apply graphical models in settings where the underlying data is sensitive and must be kept private. For example, consider applying graphical models to analyze electronic health records, with the goal of guiding public health policy. How can we derive these useful population-level outcomes without compromising the privacy of individuals?

---

<sup>1</sup>University of Massachusetts Amherst <sup>2</sup>Mount Holyoke College <sup>3</sup>Colgate University. Correspondence to: Garrett Bernstein <gbernstein@cs.umass.edu>.

*Differential privacy* is a widely studied formalism for private data analysis (Dwork et al., 2006). It provides a statistical privacy guarantee to individuals: the output of a differentially private algorithm is statistically nearly unchanged even if any single individual’s record is added to or removed from the input data set. The general idea is to carefully randomize the algorithm so that the (random) output does not depend too much on any individual’s data.

Differentially private machine learning cleanly addresses the problem of extracting useful population-level models from data sets while protecting the privacy of individuals. Indeed, this is an active and important research area (see Section 1.1), which includes private learning algorithms for a variety of general frameworks and specific machine learning models. This paper addresses the problem of privately learning parameters in a widely used class of probabilistic models: discrete, undirected graphical models. Although our problem can be cast in terms of general private learning frameworks, these do not lead to practical algorithms. Previous work also addresses private learning for *directed* graphical models (J. Zhang et al., 2014; Z. Zhang et al., 2016). Our problem of learning in undirected models, which are not locally normalized, is more general and substantially harder computationally.

To learn accurate models under differential privacy, it is critical to randomize the algorithm “just enough” to achieve the desired privacy guarantee without diminishing the quality of the learned model too much. This is usually done by modifying a learning algorithm to add noise to some intermediate quantity  $X$ , with the noise magnitude calibrated to the *sensitivity* of  $X$ , a measure of how much  $X$  can depend on any single individual’s data in the worst case (Dwork et al., 2006). The randomization renders the noisy estimate of  $X$  safe for release; all subsequent calculations using the noisy  $X$ , but not the original data, are also safe. Where should noise be injected into a machine learning algorithm to achieve the best utility? We highlight two high-level goals: (1) Noise should be added at an “information bottleneck”, so the sensitivity is as small as possible relative to the information being sought,<sup>1</sup> (2) noise should be

---

<sup>1</sup>Sensitivity scales with the number of measurements; all else equal, a lower dimensional quantity will have lower sensitivity.

added to a quantity for which the sensitivity can be bounded tightly, so the noise magnitude can be kept as small as possible. These two principles are often at odds. For example, adding noise to the final learned parameters  $\theta$  (known as *output perturbation*; Dwork et al., 2006), is appealing from the information bottleneck standpoint, but if the learning algorithm is complex we may not be able to analyze the sensitivity and would have to rely on a coarse bound. Indeed, general private learning frameworks bound the sensitivity using quantities such as Lipschitz, strong-convexity, and smoothness constants (Bassily et al., 2014; Wu et al., 2016) or diameter of the parameter space (Smith, 2008), which may be loose in practice.

In this paper we will take the approach of adding noise to the sufficient statistics of a graphical model using the Laplace mechanism, a high-level approach that has also been applied recently for directed models (Zhang et al., 2016; Foulds et al., 2016). This has a number of advantages. First, sufficient statistics, by definition, are an information bottleneck. Second, it is very easy to exactly analyze the sensitivity of sufficient statistics in graphical models, which are contingency tables. Third, adding Laplace noise to contingency tables prior to release is very simple, so it is reasonable to imagine adoption in practice, say, by public agencies.

However, it is not entirely clear how to learn parameters of a graphical model with *noisy* sufficient statistics. One option, which we will refer to as *naive MLE*, is to ignore the noise and conduct maximum-likelihood estimation as if we had true sufficient statistics. This works reasonably well in practice, and is competitive with or better than state-of-the-art general-purpose methods. In fact, we will show that naive MLE is consistent and achieves the same *asymptotic* mean-squared error as non-private MLE. However, at reasonable sample sizes the error due to privacy is significant, and the approach has several pathologies (see also Yang et al., 2012; Karwa et al., 2014; 2016), some of which make it difficult to apply in practice. Therefore, we adopt a more principled approach of performing *inference* about the true sufficient statistics within an expectationmaximization (EM) learning framework.

The remaining problem is how to conduct inference over sufficient statistics of a graphical model given noisy observations thereof. This is exactly the goal of inference in *collective graphical models* (CGMs; Sheldon & Dietterich, 2011), and we will adapt CGM inference techniques to solve this problem. Put together, our results significantly advance the state-of-the-art for privately learning discrete, undirected graphical models. We clarify the theory and practice of naive MLE. We show that it learns better models than existing state-of-the-art approaches in most scenarios across a broad range of synthetic tasks, and in experiments

modeling human mobility from wifi access point data. We then show the more principled approach of conducting inference with CGMs is superior to competing approaches in nearly all scenarios.

## 1.1. Related Work

Differential privacy has been applied to many areas of machine learning, including learning specific models such as logistic regression (Chaudhuri & Monteleoni, 2009), support vector machines (Rubinstein et al., 2009), and deep neural networks (Abadi et al., 2016); privacy in general frameworks such as empirical risk minimization (ERM; Chaudhuri et al., 2011; Kifer et al., 2012; Jain & Thakurta, 2013; Bassily et al., 2014), gradient descent (Wu et al., 2016), and parameter estimation (Smith, 2011); and theoretical analysis of what can be learned privately (e.g., Blum et al., 2005; Kasiviswanathan et al., 2011).

A key aspect of our work is conducting probabilistic inference over data or model parameters given knowledge of the probabilistic privacy mechanism and its output. Karwa et al. (2014; 2016) take a similar approach but for exponential random graph models, as do Williams & McSherry (2010), but for the factored exponential mechanism. Because sufficient statistics of graphical models are contingency tables, our work connects to the well-studied problem of releasing differentially private contingency tables (Barak et al., 2007; Yang et al., 2012; Hardt et al., 2012); we adopt the Laplace mechanism because it is simple and fits well within our learning framework.

We highlight connections between CGMs and differential privacy and adopt existing inference techniques for CGMs. In general, the inference problems we wish to solve are NP-hard (Sheldon et al., 2013), but a number of efficient approximate inference algorithms are available (Liu et al., 2014; Sun et al., 2015; Vilnis et al., 2015). In a paper that was primarily about CGM inference, Sun et al. (2015) conducted a case study using CGMs to privately learn Markov chains; we build on this approach, which was limited in scope and did not address general graph structures.

Our work connects to an active current line of work on private probabilistic inference, some of which directly addresses learning in directed graphical models, but not the more challenging problem of learning in undirected graphical models. Several closely related approaches, which we refer to as One Posterior Sampling (OPS), show that a single sample drawn from a posterior distribution is differentially private (Dimitrakakis et al., 2014; Wang et al., 2015; Zhang et al., 2016). This can be understood as applying the exponential mechanism to the log-likelihood function, and can provide a point estimate for graphical model parameters (Zhang et al., 2016). To apply OPS, one must sample from the posterior over parameters,  $p(\Theta|X)$ , which is

straightforward for directed graphical models with conjugate priors, but not in undirected models, where posteriors over parameters are usually intractable. Zhang et al. (2016) and Foulds et al. (2016) also developed fully Bayesian methods using Laplace noise-corrupted sufficient statistics to update posterior parameters. Similar considerations apply to this approach, which matches ours in that it uses the same data release mechanism, but, like OPS, requires conjugate priors and thus easily applies only to directed graphical models. Wang et al. (2015) also describe MCMC approaches to draw many private samples from a posterior distribution; this is another general framework that could apply to our problem, but, it relies on loose sensitivity bounds and since we only request point estimates, it would waste privacy budget by drawing many samples.

## 2. Background and Problem Statement

We consider data sets consisting of  $T$  discrete attributes associated with each individual. Let  $x_t \in \mathcal{X}$  denote the value of the  $t$ th attribute of an individual; we assume for simplicity of notation that all variables take values in the same finite set  $\mathcal{X}$ . Let  $\mathbf{x} = (x_1, \dots, x_T)$  denote the complete vector of attributes for an individual, and let  $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)})$  denote a data set for an entire population of  $N$  individuals.

### 2.1. Differential Privacy

Differential privacy offers strong privacy protection by imposing constraints on any algorithm that computes on the private dataset. Informally, it requires that an individual’s data has a bounded effect on the algorithm’s behavior. The formal definition requires reasoning about all pairs of datasets that are otherwise identical except one dataset contains one additional individual’s data vector. Let  $\text{nbrs}(\mathbf{X})$  denote the set of datasets that differ from  $\mathbf{X}$  by at most one individual’s vector—i.e., if  $\mathbf{X}' \in \text{nbrs}(\mathbf{X})$ , then  $\mathbf{X}' = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i-1)}, \mathbf{x}^{(i+1)}, \dots, \mathbf{x}^{(n)})$  for some  $i$  or  $\mathbf{X}' = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}, \mathbf{x}', \mathbf{x}^{(i+1)}, \dots, \mathbf{x}^{(n)})$  for some  $i$  and some  $\mathbf{x}' \in \mathcal{X}^T$ .

**Definition 1** (Differential Privacy; Dwork et al., 2006). *A randomized algorithm  $\mathcal{A}$  satisfies  $(\epsilon, \delta)$ -differential privacy if for any input  $\mathbf{X}$ , any  $\mathbf{X}' \in \text{nbrs}(\mathbf{X})$  and any subset of outputs  $S \subseteq \text{Range}(\mathcal{A})$ ,*

$$\Pr[\mathcal{A}(\mathbf{X}) \in S] \leq \exp(\epsilon)\Pr[\mathcal{A}(\mathbf{X}') \in S] + \delta.$$

When  $\delta = 0$ , we say that the algorithm satisfies  $\epsilon$ -differential privacy. All of the algorithms we propose satisfy  $\epsilon$ -differential privacy but we compare against some algorithms that satisfy the weaker condition of  $(\epsilon, \delta)$ -differential privacy with non-zero  $\delta$ .

We achieve differential privacy by injecting noise into the

statistics that are computed on the data. Let  $f$  be any function that maps datasets to  $\mathbb{R}^d$ . The amount of noise depends on the *sensitivity* of  $f$ .

**Definition 2** (Sensitivity). *The sensitivity of a function  $f$  is defined as  $\Delta_f = \max_{\mathbf{X}} LS_f(\mathbf{X})$  where  $LS_f$  denotes the local sensitivity of  $f$  on input  $\mathbf{X}$  and is defined as  $LS_f(\mathbf{X}) = \max_{\mathbf{X}' \in \text{nbrs}(\mathbf{X})} \|f(\mathbf{X}) - f(\mathbf{X}')\|_1$ .*

We drop the subscript  $f$  when it is clear from context.

Our approach achieves differential privacy through the application of the Laplace mechanism.

**Definition 3** (Laplace Mechanism; Dwork et al., 2006). *Given function  $f$  that maps datasets to  $\mathbb{R}^d$ , the Laplace mechanism is defined as  $\mathcal{L}(\mathbf{X}) = f(\mathbf{X}) + \mathbf{z}$  where  $\mathbf{z} = (z_1, \dots, z_d)$  and each  $z_i$  is an i.i.d. random variable from  $\text{Laplace}(\Delta_f/\epsilon)$ .*

An important property of differential privacy is that any additional post-processing on the output cannot weaken the privacy guarantee.

**Proposition 1** (Post-processing; Dwork & Roth, 2014). *Let  $\mathcal{A}$  be an  $(\epsilon, \delta)$ -differentially private algorithm that maps datasets to  $\mathbb{R}^d$  and let  $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$  be an arbitrary function. Then  $g \circ \mathcal{A}$  is also  $(\epsilon, \delta)$ -differentially private.*

### 2.2. Problem Statement

Our goal is to learn a probabilistic model  $p(\mathbf{x})$  from the data set  $\mathbf{X}$  while protecting the privacy of individuals. We will learn probability distributions  $p(\mathbf{x})$  that are *undirected discrete graphical models* (also called Markov random fields; Koller & Friedman, 2009). These are defined by a set of local *potential functions* of the form  $\psi_C(\mathbf{x}_C)$ , where  $C \subseteq \{1, \dots, T\}$  is an index set or *clique*,  $\mathbf{x}_C$  is a sub-vector of  $\mathbf{x}$  corresponding to  $C$ , and  $\psi_C : \mathcal{X}^{|C|} \rightarrow \mathbb{R}^+$  assigns a *potential* value to each possible  $\mathbf{x}_C$ . The probability model is  $p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{x}_C)$  where  $\mathcal{C}$  is the collection of cliques that appear in the model, and  $Z = \sum_{\mathbf{x}} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{x}_C)$  is the normalizing constant or *partition function*. The graph  $G$  with node set  $V = \{1, \dots, T\}$  and edges between any two indices that co-occur in some  $C \in \mathcal{C}$  is the *independence graph* of the model; therefore, each index set  $C$  is a clique in  $G$ .

For learning, it is most convenient to express the model in log-linear or exponential family form as:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \exp \left\{ \sum_{C \in \mathcal{C}} \sum_{i_C \in \mathcal{X}^{|C|}} \mathbb{I}\{\mathbf{x}_C = i_C\} \theta_C(i_C) - A(\boldsymbol{\theta}) \right\}. \quad (1)$$

In this expression:  $\mathbb{I}\{\cdot\}$  is an indicator function; the variable  $i_C \in \mathcal{X}^{|C|}$  denotes a particular setting of the variables  $\mathbf{x}_C$ ; the *parameters*  $\theta_C(i_C) = \log \psi_C(i_C)$  are log-potential values; the vector  $\boldsymbol{\theta} \in \mathbb{R}^d$  is the concatenation of

all parameters; and  $A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$  is the log-partition function, with the dependence of  $Z$  on the parameters now made explicit. Note that, for any  $\boldsymbol{\theta} \in \mathbb{R}^d$ , the density is strictly positive:  $p(\mathbf{x}; \boldsymbol{\theta}) > 0$  for all  $\mathbf{x}$ . This is true because the potential values  $\psi_C(i_C)$  are strictly positive, so the log-potentials are finite.

The goal is to learn parameters  $\hat{\boldsymbol{\theta}}$  from the data  $\mathbf{X}$  in a way that is  $\epsilon$ -differentially private and such that  $p(\mathbf{x}; \hat{\boldsymbol{\theta}})$  is as accurate as possible. We will measure accuracy as Kullback-Leibler divergence from an appropriate reference distribution (Kullback & Leibler, 1951). In synthetic experiments, we will measure the divergence  $D(p(\cdot; \boldsymbol{\theta}) \| p(\cdot; \hat{\boldsymbol{\theta}}))$ , where  $p(\mathbf{x}; \boldsymbol{\theta})$  is the true density. For real data, we will measure the holdout log-likelihood  $E_q[\log p(\mathbf{x}; \hat{\boldsymbol{\theta}})]$  where  $q$  is the empirical distribution of the holdout data, which is equal to a constant minus  $D(q \| p(\cdot; \hat{\boldsymbol{\theta}}))$ .

The problem of privately selecting which cliques to include in the model (i.e., *model selection* or *structure learning*) is interesting but not considered in this paper; we assume the cliques  $\mathcal{C}$  are fixed in advance by the modeler.

### 3. Approach

To develop our approach to privately learn graphical model parameters, we first discuss standard concepts related to maximum-likelihood estimation for graphical models.

**Log-Likelihood, Sufficient Statistics, Marginals.** From Eq. (1), the log-likelihood  $\mathcal{L}(\boldsymbol{\theta}) = \log \prod_{i=1}^N p(\mathbf{x}^{(i)}; \boldsymbol{\theta})$  of the entire data set can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = \left[ \sum_{C \in \mathcal{C}} \sum_{i_C \in \mathcal{X}^{|C|}} n_C(i_C) \theta_C(i_C) \right] - NA(\boldsymbol{\theta})$$

where  $n_C(i_C) = \sum_{i=1}^N \mathbb{I}\{\mathbf{x}_C^{(i)} = i_C\}$  is a count of how many times the configuration  $i_C$  for the variables in clique  $C$  appears in the population. The collection of counts  $\mathbf{n}_C = (n_C(i_C))$  for all possible  $i_C$  is the (population) *contingency table* on clique  $C$ . Let  $\mathbf{n}$  denote the vector concatenation of the contingency tables for all cliques. Then we can rewrite the log-likelihood more compactly as

$$\mathcal{L}(\boldsymbol{\theta}) = f(\mathbf{n}, \boldsymbol{\theta}) := \boldsymbol{\theta}^T \mathbf{n} - NA(\boldsymbol{\theta}) \quad (2)$$

The most common approach for parameter learning in graphical models is maximum likelihood estimation: find the parameters  $\hat{\boldsymbol{\theta}}$  that maximize  $\mathcal{L}(\boldsymbol{\theta})$ . The resulting parameter vector  $\hat{\boldsymbol{\theta}}$  is a *maximum-likelihood estimator* (MLE). It is clear from Eq. (2) that this problem depends on the data only through the contingency tables  $\mathbf{n}$ . Indeed, the clique contingency tables  $\mathbf{n}$  are *sufficient statistics* of the model: they measure all of the information from the data set  $\mathbf{X}$  that is relevant for estimating the parameter  $\boldsymbol{\theta}$  (Fisher, 1922).

The algorithmic approach for maximum-likelihood estimation in graphical models is standard (Koller & Friedman, 2009), and we do not repeat the details here. However, there are a few concepts that are important for our development. The *marginals* of a graphical model are the marginal probabilities  $\mu_C(i_C) = p(\mathbf{x}_C = i_C; \boldsymbol{\theta})$  for all cliques  $C$  and configurations  $i_C$ . Let  $\boldsymbol{\mu}$  be the vector concatenation of all marginals, and note that  $\boldsymbol{\mu} = \mathbb{E}_{\boldsymbol{\theta}}[\mathbf{n}]/N$ . Similarly, let  $\hat{\boldsymbol{\mu}} = \mathbf{n}/N$  be the *data marginals*—these are marginal probabilities of the empirical distribution of the data.

Marginals play a fundamental role in estimation. First, note that we can divide Eq. (2) by  $N$  to see that the MLE only depends on the data through the data marginals  $\hat{\boldsymbol{\mu}}$ . However, we leave  $\mathcal{L}(\boldsymbol{\theta})$  in the current form because it is more convenient for the CGM development in Section 3.2. Second, it is well known that  $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = N(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})$ , so maximum likelihood estimation seeks to adjust  $\boldsymbol{\theta}$  so that the data and model marginals match. Third, it can (almost) always succeed in doing so, even if the data marginals do not come from a graphical model. More formally, let  $\mathcal{M}$  be the *marginal polytope*: the set of all vectors  $\boldsymbol{\mu}$  such that there exists some distribution  $q(\mathbf{x})$  with marginal probabilities  $\boldsymbol{\mu}$ .

**Proposition 2** (Wainwright & Jordan, 2008). *For any  $\boldsymbol{\mu}$  in the interior of  $\mathcal{M}$ , there is a unique distribution  $p(\mathbf{x}; \boldsymbol{\theta})$  with marginals  $\boldsymbol{\mu}$ , i.e., such that  $\boldsymbol{\mu} = E_{\boldsymbol{\theta}}[\mathbf{n}]/N$ .*

Applying Proposition 2 to the data marginals  $\hat{\boldsymbol{\mu}}$  shows that if these belong to the interior of  $\mathcal{M}$ , we may learn a distribution with marginals that match what we observe in the data. Note that, while the *distribution*  $p(\mathbf{x}; \boldsymbol{\theta})$  is unique, the parameters  $\boldsymbol{\theta}$  are not, because our model is overcomplete. If  $\boldsymbol{\mu}$  belongs to  $\mathcal{M}$  but not the interior of  $\mathcal{M}$ , which occurs, for example, when some marginals are zero, the situation is more complex: there is no (finite)  $\boldsymbol{\theta} \in \mathbb{R}^d$  such that  $p(\mathbf{x}; \boldsymbol{\theta})$  has marginals  $\boldsymbol{\mu}$ .<sup>2</sup> Similarly, the MLE does not exist, meaning that its maximum is not attained for any finite  $\boldsymbol{\theta}$  (Fienberg & Rinaldo, 2012; Haberman, 1973). This issue will end up being significant in our understanding of the naive MLE approach in the following section.

#### 3.1. Noisy sufficient statistics

From the development so far, there are two obvious possibilities for randomizing the learning process to achieve privacy:

1. (Output perturbation) Find the MLE  $\hat{\boldsymbol{\theta}}$  and add Laplace noise proportional to its sensitivity.
2. (Sufficient statistics perturbation) Add Laplace noise to the sufficient statistics  $\mathbf{n}$ , and then conduct maximum-likelihood estimation.

<sup>2</sup>However, there is a sequence  $\{\boldsymbol{\theta}^k\}$  where  $\boldsymbol{\theta}^k \in \mathbb{R}^d$  and  $\lim_{k \rightarrow \infty} \mathbb{E}_{\boldsymbol{\theta}^k}[\mathbf{n}]/N = \boldsymbol{\mu}$ .

The two approaches are similar from an information bottleneck standpoint—the dimensionality of  $\mathbf{n}$  and  $\hat{\theta}$  is the same. However, the sensitivity of  $\hat{\theta}$  is difficult to analyze, since it requires reasoning about worst-case inputs. It also may be high due to pathological inputs whose local sensitivity is much higher than that of realistic data sets. On the other hand, the sensitivity of  $\mathbf{n}$  is very easy to analyze and the analysis is tight: the local sensitivity is the same for all data sets.

**Proposition 3.** *Let  $\mathbf{n}(\mathbf{X})$  be the sufficient statistics of a graphical model with clique set  $\mathcal{C}$  on data set  $\mathbf{X}$ . The local sensitivity of  $\mathbf{n}$  is  $|\mathcal{C}|$  for all inputs  $\mathbf{X}$ . Therefore the sensitivity of  $\mathbf{n}$  is  $|\mathcal{C}|$ .*

(Proofs can be found in the supplementary material.) So, a simple approach to achieve privacy is to release noisy sufficient statistics  $\mathbf{y}$  that are obtained after applying the Laplace mechanism:

$$y_C(i_C) = n_C(i_C) + \text{Laplace}(|\mathcal{C}|/\epsilon) \quad (3)$$

**Positive results.** How can we learn with noisy sufficient statistics  $\mathbf{y}$ ? A naive approach is to use  $\mathbf{y}$  in place of  $\mathbf{n}$  in maximum-likelihood estimation, i.e., to find  $\theta$  to maximize  $f(\mathbf{y}, \theta)$ . The validity of this approach has been debated in the literature (Yang et al., 2012). However, it is relatively easy to show that it behaves well asymptotically.

**Proposition 4.** *Assume  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$  are drawn iid from a probability distribution with marginals  $\mu$ . The marginal estimate  $\bar{\mu}_C(i_C) = \frac{1}{N}y_C(i_C)$  obtained from the noisy sufficient statistics is unbiased and consistent, with mean squared error:*

$$\text{MSE}(\bar{\mu}_C(i_C)) = \frac{\mu_C(i_C)(1 - \mu_C(i_C))}{N} + \frac{2|\mathcal{C}|^2}{N^2\epsilon^2} \quad (4)$$

Now let  $\hat{\theta} \in \text{argmax}_{\theta} f(\mathbf{y}, \theta)$  be parameters estimated using the noisy sufficient statistics  $\mathbf{y}$ . If the true distribution  $p(\mathbf{x}; \theta)$  is a graphical model with cliques  $\mathcal{C}$ , then the estimated distribution  $p(\mathbf{x}; \hat{\theta})$  converges to  $p(\mathbf{x}; \theta)$ .

**Pathologies.** Asymptotically, the noisy sufficient statistics behave as desired in terms of MSE: the  $O(1/N)$  term, which is due to sampling error and not privacy, dominates for large  $N$ . However, for practical settings of  $\epsilon$  the  $O(1/N^2)$  term, which is due to privacy, is dominant until  $N$  becomes very large, due to the large constant  $2|\mathcal{C}|^2/\epsilon^2$ . Figure 1(a) illustrates this issue.

A second pathology is that the noise added for privacy destroys some of the structure expected in the empirical marginals. The true data marginals  $\hat{\mu} = \mathbf{n}/N$  belong to the marginal polytope: in particular, this means that each clique marginal  $\hat{\mu}_C$  is nonnegative and sums to one, and

that clique marginals agree on common subsets of variables. After adding noise, the *pseudo*-marginals  $\bar{\mu} = \mathbf{y}/N$  do not belong to the marginal polytope:  $\bar{\mu}$  may have negative values, and does not satisfy consistency constraints. We find that a partial fix is very helpful empirically: project the pseudo-marginal  $\bar{\mu}_C$  for each clique onto the simplex prior to conducting MLE, which can be done via a standard procedure (Duchi et al., 2008). Let  $\tilde{\mu}$  be the projected marginals. We now have that  $\tilde{\mu}_C$  is a valid marginal for each clique  $C$ , but consistency constraints are not satisfied among cliques, and it is still the case that  $\tilde{\mu} \notin \mathcal{M}$ . Figure 1(b) illustrates the benefits of projection on the quality of the model learned by Naive MLE.

A more significant pathology has to do with zeros in the projected marginals  $\tilde{\mu}$ , which are more prevalent than in true data marginals  $\hat{\mu}$ . This is because the addition of Laplace noise creates negative values, which are then truncated to zero during projection. As discussed following Proposition 2, zero values in the marginals lead to non-existence of the MLE (Fienberg & Rinaldo, 2012; Haberman, 1973). If  $\tilde{\mu}_C(i_C) = 0$ , the likelihood increases monotonically as  $\theta_C(i_C)$  goes to negative infinity; in other words, the model attempts to drive the learned marginal probability to zero. Numerically, we can address this by regularization, e.g., adding  $\lambda\|\theta\|^2$  to the objective function for arbitrarily small  $\lambda > 0$ . However, we may still learn vanishingly small marginal probabilities, which can lead to a very large KL-divergence between the true and learned models. Figure 1(c) illustrates the effect of  $\lambda$  on KL-divergence with both noisy sufficient statistics and true sufficient statistics. At high  $\lambda$  (strong regularization), both methods underfit and yield poor KL divergence. Learning with true sufficient statistics has no tendency to overfit; it achieves good performance for a broad range of  $\lambda$  approaching zero. Naive MLE with noisy sufficient statistics overfits badly (to zeros) for small  $\lambda$ , and must be tuned “just right” to achieve reasonable performance.

### 3.2. Collective Graphical Models

Since learning with noisy sufficient statistics “as-is” has several pathologies and is less robust than maximum-likelihood estimation in the absence of privacy, we investigate a more principled approach, which matches the data generating process: We treat the true sufficient statistics  $\mathbf{n}$  as latent variables, and learn  $\theta$  to maximize the *marginal* likelihood  $p(\mathbf{y}; \theta) = \sum_{\mathbf{n}} p(\mathbf{n}, \mathbf{y}; \theta)$ . In this section, we will develop an EM approach to accomplish this.

In EM, we need to conduct inference to compute  $\mathbb{E}[\mathbf{n} | \mathbf{y}; \theta]$  for a fixed value of  $\theta$ . This is the central problem of *collective graphical models* (CGMs) (Sheldon & Dietterich, 2011). Consider the joint distribution  $p(\mathbf{n}, \mathbf{y}; \theta) = p(\mathbf{n}; \theta)p(\mathbf{y} | \mathbf{n})$ , which we use to compute  $\mathbb{E}[\mathbf{n} | \mathbf{y}; \theta]$ . The

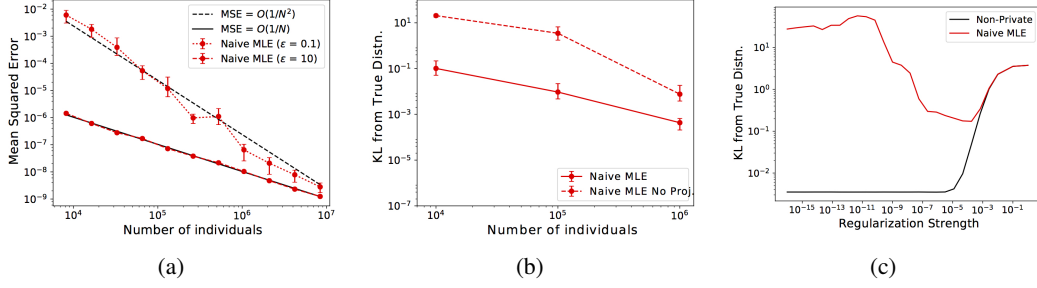


Figure 1. Sample results on synthetic data illustrating behavior of naive MLE (see Section 4.1 for experiment details): (a) MSE of learned marginals vs population size  $N$  on a chain model with  $T = 10$ ,  $|\mathcal{X}| = 10$ ; reference lines indicate predicted slope for  $O(1/N)$  and  $O(1/N^2)$  error terms, respectively (the function  $c/N^d$  has slope  $-d$  on a log-log plot), (b) effect of projecting marginals on performance of naive MLE for an Erdős-Rényi graph with  $T = 10$ ,  $|\mathcal{X}| = 20$ ,  $\epsilon = 0.5$ , (c) effect of regularization on KL-divergence for learning with and without privacy; chain model with  $T = 10$ ,  $|\mathcal{X}| = 10$ ,  $\epsilon = 0.1$ .

noise mechanism  $p(\mathbf{y} | \mathbf{n})$  arises directly from the Laplace mechanism (see Eq. (3)). The distribution of the sufficient statistics,  $p(\mathbf{n}; \boldsymbol{\theta})$ , is known as the *CGM distribution*. It can be written in closed form when the model is *decomposable*, i.e., the cliques  $\mathcal{C}$  correspond to the nodes of some junction tree  $\mathcal{T}$ . Although decomposability is a significant restriction, let us assume that such a tree  $\mathcal{T}$  exists; we will use the exact results derived for this case to develop an approximation for the general case. Let  $\mathcal{S}$  be the set of separators of  $\mathcal{T}$ , and let  $\nu(S)$  be the multiplicity of  $S \in \mathcal{S}$ , i.e., the number of distinct edges  $(C_i, C_j) \in \mathcal{T}$  for which  $S = C_i \cap C_j$ . Under these assumptions, the CGM distribution has the form (Liu et al., 2014):

$$p(\mathbf{n}; \boldsymbol{\theta}) = h(\mathbf{n}) \cdot \exp(f(\mathbf{n}, \boldsymbol{\theta})),$$

$$h(\mathbf{n}) = N! \cdot \frac{\prod_{S \in \mathcal{S}} \prod_{i_S \in \mathcal{X}^{|S|}} (n_S(i_S)!)^{\nu(S)}}{\prod_{\mathcal{C} \in \mathcal{C}} \prod_{i_{\mathcal{C}} \in \mathcal{X}^{|\mathcal{C}|}} n_{\mathcal{C}}(i_{\mathcal{C}})!} \cdot \mathbb{I}\{\mathbf{n} \in \mathcal{M}_N^{\mathbb{Z}}\}$$

The term  $\exp(f(\mathbf{n}, \boldsymbol{\theta}))$  is the probability of an ordered data set  $\mathbf{X}$  with sufficient statistics  $\mathbf{n}$ , as discussed previously. The term  $h(\mathbf{n})$  is a base measure that counts the number of ordered data sets with sufficient statistics equal to  $\mathbf{n}$ , and enforces constraints on  $\mathbf{n}$ . The *integer-valued marginal polytope*  $\mathcal{M}_N^{\mathbb{Z}}$  is the set of all vectors  $\mathbf{n}$  that are sufficient statistics of some data set  $\mathbf{X}$  of size  $N$ .

Exact inference in CGMs is intractable (Sheldon et al., 2013). Therefore, it is typical to relax the integrality constraint and apply Stirling’s approximation:  $\log n! \approx n \log n - n$ . Let  $\mathcal{M}_N$  be the feasible set with the integrality constraint removed, which is now just the standard marginal polytope scaled so that each marginal sums to  $N$  instead of one.

**Proposition 5** (Sun et al.; Nguyen et al., 2015; 2016). *For a decomposable CGM with junction tree  $\mathcal{T}$ , the following*

*approximation of the CGM log-density for any  $\mathbf{n} \in \mathcal{M}_N$  is obtained by applying Stirling’s approximation:*

$$\log p(\mathbf{n}, \mathbf{y}; \boldsymbol{\theta}) \approx \boldsymbol{\theta}^T \mathbf{n} - N A(\boldsymbol{\theta}) + H(\mathbf{n}) + \log p(\mathbf{y} | \mathbf{n}). \quad (5)$$

Here,  $H(\mathbf{n}) = -N \sum_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x})$  is the entropy of the unique distribution  $q(\mathbf{x}) = p(\mathbf{x}; \boldsymbol{\theta})$  in the graphical model family with marginals equal to  $\mathbf{n}/N$ .

Proposition 5 is the basis for *approximate MAP inference problem in CGMs*: find  $\mathbf{n}$  to maximize Eq. (5) and obtain an approximate mode of  $p(\mathbf{n} | \mathbf{y}; \boldsymbol{\theta})$ . Even though our goal is to compute the *mean*  $\mathbb{E}[\mathbf{n} | \mathbf{y}; \boldsymbol{\theta}]$ , it has been shown that the approximate mode, which is also a real-valued vector, is an excellent approximation to the mean for use within the EM algorithm (Sheldon et al., 2013). Note that for non-decomposable models, we will simply apply the same approximation as in Proposition 5, even though an exact expression for the counting measure  $h(\mathbf{n})$ , and therefore the correspondence of  $\log h(\mathbf{n})$  to an entropy  $H(\mathbf{n})$ , is not known in this case. Then, after dropping the term  $NA(\boldsymbol{\theta})$  from Proposition 5, which is constant with respect to  $\mathbf{n}$ , the approximate MAP problem can be rewritten as:

$$\mathbf{n}^* \in \operatorname{argmax}_{\mathbf{n} \in \mathcal{M}_N} \boldsymbol{\theta}^T \mathbf{n} + H(\mathbf{n}) + \log p(\mathbf{y} | \mathbf{n}) \quad (6)$$

This equation reveals a close connection to variational principles for graphical models (Wainwright & Jordan, 2008). It is identical to the variational optimization problem for marginal inference in standard graphical models, except the objective has an additional term  $\log p(\mathbf{y} | \mathbf{n})$ , which is non-linear in  $\mathbf{n}$ . Several message-passing based algorithms have been developed to efficiently solve the approximate MAP problem. For trees or junction trees, Problem (6) is convex as long as  $\log p(\mathbf{y} | \mathbf{n})$  is concave in  $\mathbf{n}$  (which is true in most cases of interest, such as Laplace noise) so it can be solved exactly (Sun et al., 2015; Vilnis et al., 2015). For loopy models, both the entropy  $H(\mathbf{n})$  and the feasible set  $\mathcal{M}_N$  must be approximated (Nguyen et al., 2016).

---

**Algorithm 1** Non-Linear Belief Propagation (NLBP)
 

---

**Input:**  $\theta, \mathbf{y}$ , damping parameter  $\alpha > 0$   
**while**  $\neg$  converged **do**  
      $\theta' \leftarrow \theta + \nabla_{\mathbf{n}} \log p(\mathbf{y} | \mathbf{n})$   
      $\mathbf{n}' \leftarrow \text{STANDARD-BP}(\theta')$  {Normalized to sum to  $N$ }  
      $\mathbf{n} \leftarrow (1 - \alpha)\mathbf{n} + \alpha\mathbf{n}'$   
**end while**

---



---

**Algorithm 2** EM for CGMs
 

---

**Input:** Noisy sufficient statistics  $\mathbf{y}$   
 Initialize  $\theta_0$  arbitrarily  
**while**  $\neg$  converged **do**  
      $\mathbf{n}_t \leftarrow \text{NLBP}(\theta_t, \mathbf{y})$   
      $\theta_{t+1} \leftarrow \arg\max_{\theta} \theta^T \mathbf{n}_t - NA(\theta)$   
**end while**

---

Algorithm 1 shows pseudocode *non-linear belief propagation* (NLBP; Sun et al., 2015), which we select as our primary inference approach due to its simplicity. It is a thin wrapper around standard BP, and can be applied to trees, in which case it exactly solves Problem (6), or it can be applied to loopy graphs by using loopy BP (LBP) as the subroutine, in which case it is approximate.

Our final EM learning procedure is shown in Algorithm 2. It alternates between inference steps that solve the approximate MAP problem to find  $\mathbf{n}_t \approx \mathbb{E}[\mathbf{n} | \mathbf{y}; \theta_t]$ , and optimization steps to re-estimate parameters given the inferred sufficient statistics  $\mathbf{n}_t$ . See also (Sheldon et al., 2013; Liu et al., 2014; Sun et al., 2015).

## 4. Experiments

We conduct a number of experiments on synthetic and real data to evaluate the quality of models learned by both Naive MLE and CGM.

**Methods.** We compare three algorithms: Naive MLE, CGM, and a version of private stochastic gradient descent (PSGD) due to Abadi et al. (2016). PSGD belongs to a class of general-purpose private learning algorithms that can be adapted to our problem, including gradient descent or stochastic gradient descent algorithms for empirical risk minimization (Chaudhuri et al., 2011; Kifer et al., 2012; Jain & Thakurta, 2013; Bassily et al., 2014; Abadi et al., 2016) and the subsample-and-aggregate approach for parameter estimation (Smith, 2011). We chose PSGD because it is a state-of-the-art method and it significantly outperformed other approaches in preliminary experiments. However, note that PSGD satisfies only  $(\epsilon, \delta)$ -differential privacy for  $\delta > 0$ , which is a weaker privacy guarantee than  $\epsilon$ -differential privacy. We tune PSGD using a grid search over all relevant parameters to ensure it performs as well as

possible.

### 4.1. Synthetic data

We evaluate two types of pairwise graphical models: third-order chains with edges between two nodes  $i$  and  $j$  if  $1 \leq |i - j| \leq 3$ , and (connected) Erdős-Rényi (ER) random graphs. We report results for graphs of 10 nodes, where potentials on each edge are drawn from a Dirichlet distribution with concentration parameter of one; results are similar for smaller and larger models, models with different structures, and for different types of potentials. We vary data size  $N$  and privacy parameter  $\epsilon$ . For each setting of model type,  $N$ , and  $\epsilon$ , we conduct 25 trials. The trials are nested, with five random populations and five replications per population, i.e.:  $\mathbf{n}_i \sim p(\mathbf{n}), \mathbf{y}_{i,j} \sim p(\mathbf{y} | \mathbf{n}_i)$  for  $i \in \{1, \dots, 5\}, j \in \{1, \dots, 5\}$ . We measure the quality of learned models using KL divergence from the true distribution, and include for comparison two reference models: a random estimator and a non-private MLE estimator. The random estimator is obtained by randomly generating marginals  $\bar{\mu}$  and then learning potentials via MLE.

**Results.** Figure 2 shows the results for the two models (top: third-order chain, bottom: ER) for different values of  $N$  and  $\epsilon$ . CGM improves upon Naive MLE for all models, privacy levels, and population sizes. Recall that PSGD promises only  $(\epsilon, \delta)$ -differential privacy. While  $\delta$  is often assumed to be “cryptographically small”, e.g.,  $O(2^{-N})$ , we set  $\delta$  to a relatively large value of  $\delta = 1/N$ . Increasing  $\delta$  weakens the privacy guarantee but enables PGSD to run on a wider range of  $\epsilon$ . However, even with this setting for  $\delta$ , some of the smaller values of  $\epsilon$  are not attainable by PGSD and are omitted from those plots.

Figure 3(a) shows a qualitative comparison of edge marginals of a single graph learned by the different methods, compared with the true model marginals; it is evident that CGM learns marginals that are much closer to both the true marginals and those learned by the non-private estimator than Naive MLE is able to learn. Naive MLE is the fastest method; CGM is approximately 4x/8x slower on third-order chains and ER graphs, respectively, and PSGD is approximately 27x/40x slower.

### 4.2. Wifi data

We study human mobility data in the form of connections to wifi access points throughout a highly-trafficked academic building over a twenty-one day period. We treat each (user ID, day) combination as an “individual”, leading to 124,399 unique individuals; with this data preparation scheme, the unit of protection is one day’s worth of a user’s data. We discretize time by recording the location every 10 minutes, and assign null if the user is not connected to the network. Our probability model  $p(\mathbf{x})$  is a

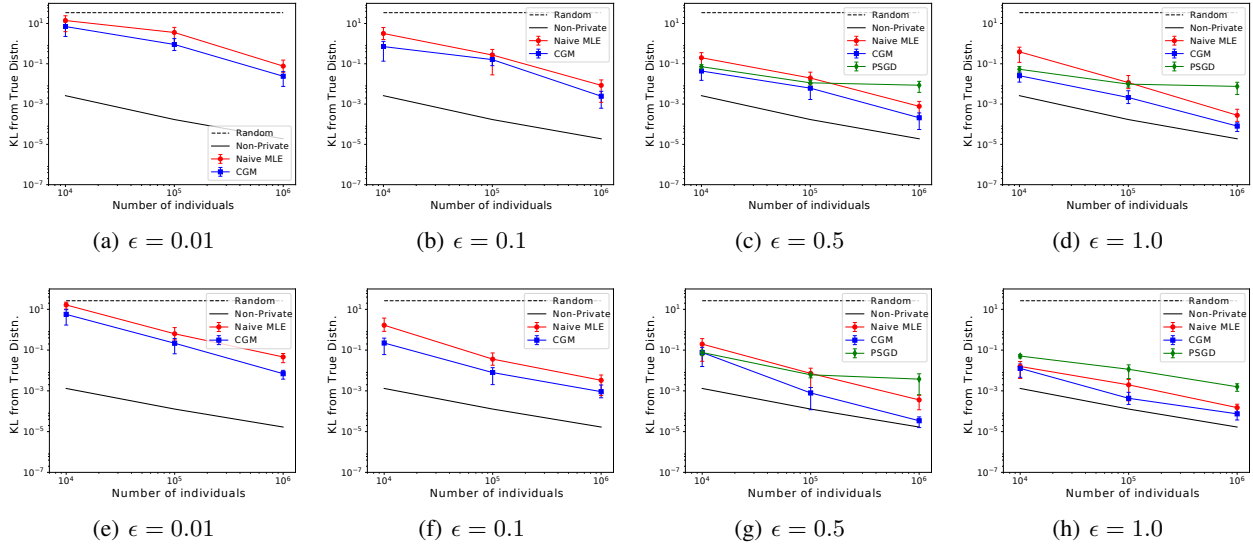


Figure 2. Results on synthetic data generated from third-order chains, (a)–(d), and connected Erdős-Rényi random graphs (e)–(h). Each column represents a different privacy level. Lower  $\epsilon$  signifies stricter privacy guarantees. The x-axis measures population size. The y-axis is KL divergence from the true distribution.

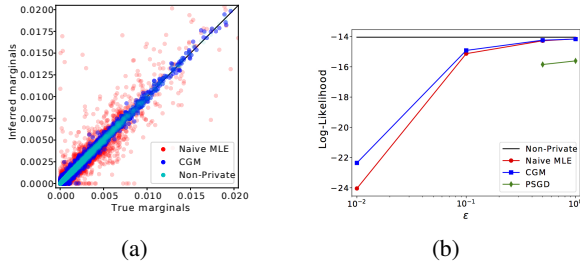


Figure 3. (a) Scatter plots for true vs. inferred values of all edge marginals in an ER graph of 10 nodes with 20 states each. (b) Results for fitting a first-order chain on wifi data. The x-axis is privacy level; lower  $\epsilon$  signifies stronger privacy guarantees. The y-axis is holdout log-likelihood.

pairwise graphical model over hour-long segments. Therefore, we break each individual’s data into 24 one-hour long segments.

An individual now contributes 24 records to each contingency table for the model  $p(\mathbf{x})$ . Therefore, the sensitivity is now 24 times the number of edges (cliques). However, real data is typically sparse—i.e., an individual is typically observed only a small number of times over the observation period. Therefore, to reduce the sensitivity, the data is *normalized* prior to calculating sufficient statistics, in a fashion similar to (He et al., 2015). Each user contributes a value of  $1/K$  to each contingency table, where  $K$  is the number of edges  $(x_s, x_t)$  for which the user’s values are not both null. With this pre-processing in place, the sensitivity equals the number of edges in the model. A trade-off of

this technique is that we bias the model towards individuals with fewer transitions, but we reduce the amount of noise by limiting sensitivity caused by null–null transitions.

We reserve data from 25% of the individuals for testing. To compare different approaches, we apply Naive MLE, CGM, and PSGD to privately learn parameters of a graphical model from the training set (75% of the data), with varying privacy levels. We then calculate holdout log-likelihood of the learned parameters on the test set. We again include a non-private method for reference, but in this case, all methods perform better than the random estimator, so we do not show it.

Figure 3(b) shows the results for fitting a time-homogeneous chain model (edges between adjacent time steps, every potential  $\psi(x_t, x_{t+1})$  is the same, and the model includes a node potential  $\phi(x_1)$  so it can learn a time-stationary model). As in the synthetic data experiments, CGM improves upon naive MLE across all parameter regimes, and performance improves with population size  $N$  and with weakening of privacy (larger  $\epsilon$ ). Both methods outperform PSGD. Naive MLE is the fastest method; CGM is approximately 15x slower, and PSGD is approximately 46x slower.

**Acknowledgments**

This material is based upon work supported by the National Science Foundation under Grant Nos. 1409125, 1409143, 1421325, and 1617533.



## References

- Abadi, Martín, Chu, Andy, Goodfellow, Ian, McMahan, H. Brendan, Mironov, Ilya, Talwar, Kunal, and Zhang, Li. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.
- Barak, Boaz, Chaudhuri, Kamalika, Dwork, Cynthia, Kale, Satyen, McSherry, Frank, and Talwar, Kunal. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 273–282. ACM, 2007.
- Bassily, Raef, Smith, Adam, and Thakurta, Abhradeep. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 464–473. IEEE, 2014.
- Blum, Avrim, Dwork, Cynthia, McSherry, Frank, and Nissim, Kobbi. Practical privacy: the SuLQ framework. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pp. 128–138. ACM, 2005.
- Chaudhuri, Kamalika and Monteleoni, Claire. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pp. 289–296, 2009.
- Chaudhuri, Kamalika, Monteleoni, Claire, and Sarwate, Anand D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar): 1069–1109, 2011.
- Dimitrakakis, Christos, Nelson, Blaine, Mitrokotsa, Aikaterini, and Rubinstein, Benjamin I.P. Robust and private Bayesian inference. In *International Conference on Algorithmic Learning Theory*, pp. 291–305. Springer, 2014.
- Duchi, John, Shalev-Shwartz, Shai, Singer, Yoram, and Chandra, Tushar. Efficient projections onto the  $l_1$ -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279. ACM, 2008.
- Dwork, Cynthia and Roth, Aaron. *The Algorithmic Foundations of Differential Privacy*. Found. and Trends in Theoretical Computer Science, 2014.
- Dwork, Cynthia, McSherry, Frank, Nissim, Kobbi, and Smith, Adam. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pp. 265–284. Springer, 2006.
- Fienberg, Stephen E. and Rinaldo, Alessandro. Maximum likelihood estimation in log-linear models. *The Annals of Statistics*, pp. 996–1023, 2012.
- Fisher, Ronald Aylmer. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922.
- Foulds, James, Geumlek, Joseph, Welling, Max, and Chaudhuri, Kamalika. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI'16*, pp. 192–201, 2016.
- Haberman, Shelby J. Log-linear models for frequency data: Sufficient statistics and likelihood equations. *The Annals of Statistics*, 1(4):617–632, 1973. ISSN 00905364. URL <http://www.jstor.org/stable/2958307>.
- Hardt, Moritz, Ligett, Katrina, and McSherry, Frank. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pp. 2339–2347, 2012.
- He, Xi, Cormode, Graham, Machanavajjhala, Ashwin, Procopiuc, Cecilia M., and Srivastava, Divesh. DPT: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment*, 8(11):1154–1165, 2015.
- Jain, Prateek and Thakurta, Abhradeep. Differentially private learning with kernels. *ICML (3)*, 28:118–126, 2013.
- Karwa, Vishesh, Slavković, Aleksandra B., and Krivitsky, Pavel. Differentially private exponential random graphs. In *International Conference on Privacy in Statistical Databases*, pp. 143–155. Springer, 2014.
- Karwa, Vishesh, Slavković, Aleksandra, et al. Inference using noisy degrees: Differentially private  $\beta$ -model and synthetic graphs. *The Annals of Statistics*, 44(1): 87–112, 2016.
- Kasiviswanathan, Shiva Prasad, Lee, Homin K., Nissim, Kobbi, Raskhodnikova, Sofya, and Smith, Adam. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.
- Kifer, Daniel, Smith, Adam, and Thakurta, Abhradeep. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1(41):3–1, 2012.
- Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

- Kullback, Solomon and Leibler, Richard A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- Liu, Li-Ping, Sheldon, Daniel R., and Dietterich, Thomas G. Gaussian approximation of collective graphical models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- Nguyen, Duc Thien, Kumar, Akshat, Lau, Hoong Chuin, and Sheldon, Daniel. Approximate inference using DC programming for collective graphical models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 685–693, 2016.
- Rubinstein, Benjamin I.P., Bartlett, Peter L., Huang, Ling, and Taft, Nina. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *arXiv preprint arXiv:0911.5708*, 2009.
- Sheldon, Daniel R. and Dietterich, Thomas G. Collective graphical models. *Neural Information Processing Systems (NIPS)*, 2011.
- Sheldon, Daniel R., Sun, Tao, Kumar, Akshat, and Dietterich, Thomas G. Approximate inference in collective graphical models. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.
- Smith, Adam. Efficient, differentially private point estimators. *arXiv preprint arXiv:0809.4794*, 2008.
- Smith, Adam. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (STOC)*, pp. 813–822. ACM, 2011.
- Sun, Tao, Sheldon, Daniel R., and Kumar, Akshat. Message passing for collective graphical models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.
- Vilnis, Luke, Belanger, David, Sheldon, Daniel, and McCallum, Andrew. Bethe projections for non-local inference. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 892–901. AUAI Press, 2015.
- Wainwright, Martin J. and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.
- Wang, Yu-Xiang, Fienberg, Stephen, and Smola, Alex. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2493–2502, 2015.
- Williams, Oliver and McSherry, Frank. Probabilistic inference and differential privacy. In *Advances in Neural Information Processing Systems*, pp. 2451–2459, 2010.
- Wu, Xi, Kumar, Arun, Chaudhuri, Kamalika, Jha, Somesh, and Naughton, Jeffrey F. Differentially private stochastic gradient descent for in-RDBMS analytics. *CoRR*, abs/1606.04722, 2016. URL <http://arxiv.org/abs/1606.04722>.
- Yang, Xiaolin, Fienberg, Stephen E., and Rinaldo, Alessandro. Differential privacy for protecting multidimensional contingency table data: Extensions and applications. *Journal of Privacy and Confidentiality*, 4(1): 5, 2012.
- Zhang, Jun, Cormode, Graham, Procopiuc, Cecilia M., Srivastava, Divesh, and Xiao, Xiaokui. Privbayes: Private data release via Bayesian networks. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1423–1434, 2014.
- Zhang, Zuhe, Rubinstein, Benjamin I.P., and Dimitrakakis, Christos. On the differential privacy of Bayesian inference. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.