Data-Dependent Differentially Private Parameter Learning for Directed Graphical Models

Amrita Roy Chowdhury [1] **Theodoros Rekatsinas** [1] **Somesh Jha** [1 2]

Abstract

Directed graphical models (DGMs) are a class of probabilistic models that are widely used for predictive analysis in sensitive domains such as medical diagnostics. In this paper, we present an algorithm for differentially private learning of the parameters of a DGM. Our solution optimizes for the utility of inference queries over the DGM and *adds noise that is customized to the properties of the private input dataset and the graph structure of the DGM*. To the best of our knowledge, this is the first explicit data-dependent privacy budget allocation algorithm in the context of DGMs. We compare our algorithm with a standard data-independent approach over a diverse suite of benchmarks and demonstrate that our solution requires a privacy budget that is roughly $3\times$ smaller to obtain the same or higher utility.

1. Introduction

Directed graphical models (DGMs) are a class of probabilistic models that are widely used in causal reasoning and predictive analytics (Koller & Friedman, 2009). A typical use case for these models is answering "what-if" queries over domains that often work with sensitive information. For example, DGMs are used in medical diagnosis for answering questions, such as what is the most probable disease given a set of symptoms (Pearl, 1998). In learning such models, it is common that the underlying graph structure of the model is publicly known. For instance, in the case of medical data, the dependencies between several physiological symptoms and diseases are well established, standardized, and publicly available. However, the parameters of the model have to be learned from observations. These observations may contain sensitive information as in the case of medical applications. Hence, learning and publicly releasing the parameters of the probabilistic model may lead to privacy violations (Shokri et al., 2017; Zhang et al., 2016a), and thus, the need for privacy-preserving learning mechanisms for DGMs.

In this paper, we focus on the problem of privacy-preserving learning of the parameters of a DGM. For our privacy definition, we use differential privacy (DP) (Dwork & Roth, 2014) – currently the de-facto standard for privacy. We consider the setting when *the structure of the target DGM is publicly known and the parameters of the model are learned from fully observed data*. In this case, all parameters can be estimated via counting queries over the input observations (also referred to as *data set* in the remainder of the paper). The direct way to ensure differential privacy is to add suitable noise to the observations using the standard Laplace mechanism (Dwork & Roth, 2014). Unfortunately, this method is data-independent, i.e., the noise added to the base observations is oblivious of the properties of the input data set and the structure of the DGM, resulting in sub-optimal utility. To address this issue, we turn to *data-dependent methods* which add noise that is customized to the properties of the input data sets (Li et al., 2014; Zhang et al.; Cormode et al., 2012b; Acs et al., 2012; Xu et al., 2012; Xiao et al., 2012; Kotsogiannis et al., 2017b).

We propose a *data-dependent*, ϵ-DP algorithm for learning the parameters of a DGM over fully observed data. Our goal is to minimize errors in arbitrary inference queries that are subsequently answered over the learned DGM. The main contributions are:

(1) **Explicit data-dependent privacy-budget allocation:** Our algorithm computes the parameters of the conditional probability distribution of each random variable in the DGM via separate measurements from the input data set. This lets us optimize the privacy budget allocation across the different variables with the objective of reducing the error in inference queries. We formulate this optimization objective in a data-dependent manner – our optimization objective is informed by both the private input data set and the public graph structure of the DGM. To the best of our knowledge, this is the first work to propose *explicit data-dependent privacy-budget allocation in the context of DGMs*. We evaluate our algorithm on four DGM benchmarks and demon-

[1]Department of Computer Sciences, University of Wisconsin-Madison, USA [2]XaiPient, USA. Correspondence to: Amrita Roy Chowdhury <amrita@cs.wisc.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

strate that our scheme only requires a privacy budget of $\epsilon = 1.0$ to yield the same utility that a data-independent baseline achieves with a much higher $\epsilon = 3.0$. Specifically, our baseline is based on (Zhang et al., 2016b) which is the most recent work that explicitly deals with differentially private parameter estimation for DGMs.

(2) **New theoretical results:** To preserve privacy, we add noise to the parameters of the DGM. To understand how this noise propagates to inference queries, we provide two new theoretical results on the upper and lower bound of the error of inference queries. The upper bound has an exponential dependency on the treewidth of the DGM while the lower bound depends on its maximum degree. We also provide a formulation to compute the sensitivity (Laskey, 1995) of the parameters associated with a node of a DGM targeting the probability distribution of its child nodes only. To the best of our knowledge, these theoretical results are novel.

2. Background

In this section, we review basic background material relevant to this paper.

Directed Graphical Models: A directed graphical model (DGM) or a Bayesian network is a probabilistic model that is represented as a directed acyclic graph, $\mathcal{G}$. The nodes of the graph represent random variables and the edges encode conditional dependencies between the variables. The graphical structure of the DGM represents a factorization of the joint probability distribution of these random variables. Specifically, given a DGM with graph $\mathcal{G}$, let $X_1, \ldots, X_n$ be the random variables corresponding to the nodes of $\mathcal{G}$ and X_{pa_i} denote the set of parents in $\mathcal{G}$ for the node corresponding to variable X_i. The joint probability distribution factorizes as

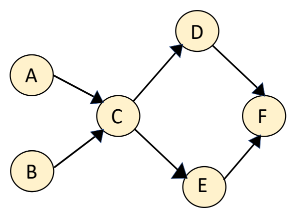


Figure 1: An example directed graphical model

$$P[X_1, \ldots, X_n] = \prod_{i=1}^{n} P[X_i|X_{pa_i}] \quad (1)$$

where each factor $P[X_i|X_{pa_i}]$ corresponds to a conditional probability distribution (CPD). For example, for the DGM depicted by Fig. 1, we have $P[A,B,C,D,E,F] = P[A] \cdot P[B] \cdot P[C|A,B] \cdot P[D|C] \cdot P[E|C] \cdot P[F|D,E]$. For DGMs with discrete random variables, each CPD can be represented as a table of parameters $\Theta_{x_i|x_{pa_i}}$ where each parameter corresponds to a conditional probability and x_i and x_{pa_i} denote variable assignments $X_i = x_i$ and $X_{pa_i} = x_{pa_i}$.

A key task in DGMs is **parameter learning**. Given a DGM with a graph structure $\mathcal{G}$, the goal of parameter learning is to estimate each $\Theta_{x_i|x_{pa_i}}$, a task solved via maximum likelihood estimation (MLE). In the presence of fully observed data $\mathcal{D}$ (i.e., data corresponding to all the nodes of $\mathcal{G}$[1] is available), the maximum likelihood estimates of the CPD parameters take the closed-form (Koller & Friedman, 2009)

$$\Theta_{x_i|x_{pa_i}} = C[x_i, x_{pa_i}]/C[x_{pa_i}] \quad (2)$$

where $C[x_i]$ is the number of records in $\mathcal{D}$ with $X_i = x_i$.

After learning, the DGM is used to answer **inference queries**, i.e., queries that compute the probabilities of certain events (variables) of interest. Inference queries can also include evidence (a subset of the nodes has a fixed assignment). There are three types inference queries in general:

(1) **Marginal inference:** This is used to answer queries of the type "what is the probability of a given variable if all others are marginalized". An example marginal inference query for the DGM in Fig. 1 is $P[F=0] = \sum_A \sum_B \sum_C \sum_D \sum_F P[A,B,C,D,E,F=0]$.

(2) **Conditional Inference:** This type of query answers the probability distribution of some variable conditioned on some evidence e. An example conditional inference query for the DGM in Fig. 1 is $P[A|F=0] = \frac{P[A,F=0]}{P[F=0]}$.

(3) **Maximum a posteriori (MAP) inference:** This type of query asks for the most likely assignment of variables. An example MAP query for the DGM in Fig. 1 is $\max_{A,B,C,D,E}\{P[A,B,C,D,E,F=0]\}$.

For DGMs, inference queries can be answered exactly by the **variable elimination (VE)** algorithm (Koller & Friedman, 2009) which is described in detail in Appx. 8.1.1. The basic idea is that we "*eliminate*" one variable at a time following a predefined order $\prec$ over the graph nodes. Let Φ denote a set of probability factors ϕ (initialized with all the CPDs of the DGM) and Z denote the variable to be eliminated. First, all probability factors involving Z are removed from Φ and multiplied together to generate a new product factor. Next, Z is summed out from this combined factor, generating a new factor $\boldsymbol{\phi}$ that is entered into Φ. Thus, VE corresponds to repeated sum-product computations: $\boldsymbol{\phi} = \sum_Z \prod_{\phi \in \Phi} \phi$.

Additionally, we define a term Markov blanket which is used in Sec. 3.3.

Definition 2.1 (Markov Blanket)**.** The Markov blanket, denoted by $\mathcal{P}(X)$, for a node X in a graphical model is the set of all nodes such that given $\mathcal{P}(X)$, X is conditionally independent of all the other nodes. (Pearl, 1988).

In a DGM, the Markov blanket of a node consists of its child nodes, parent nodes, and the parents of its child nodes. For example, in Fig. 1, $\mathcal{P}(D) = \{C, E, F\}$.

[1]The attributes of the data set become the nodes of the DGM's graph. For the remainder of the paper we use them interchangeably.

Differential Privacy: We formally define differential privacy (DP) as follows:

Definition 2.2 (Differential Privacy)**.** A randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy (ϵ-DP), where $\epsilon > 0$ is a privacy parameter, iff for any two data sets $\mathcal{D}$ and $\mathcal{D}'$ that differ in a single record, we have

$$\forall t \in Range(\mathcal{A}), P\big[\mathcal{A}(\mathcal{D}) = t\big] \le e^{\epsilon} P\big[\mathcal{A}(\mathcal{D}') = t\big] \quad (3)$$

In our setting, $\mathcal{A}$ (in Eq. (3)) corresponds to an algorithm for learning the parameters of a DGM with a publicly known graph structure from a fully observed data set $\mathcal{D}$.

When applied multiple times, the DP guarantee degrades gracefully as follows.

Theorem 2.1 (Sequential Composition)**.** *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are ϵ_1-DP and ϵ_2-DP algorithms that use independent randomness, then releasing the outputs $(\mathcal{A}_1(\mathcal{D}), \mathcal{A}_2(\mathcal{D}))$ on database $\mathcal{D}$ satisfies $(\epsilon_1 + \epsilon_2)$-DP.*

Any post-processing computation performed on the noisy output of a DP algorithm does not degrade privacy.

Theorem 2.2 (Post-processing)**.** *Let $\mathcal{A} : \mathcal{D} \mapsto R$ be a ϵ-DP algorithm. Let $f : R \mapsto R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{A} : \mathcal{D} \mapsto R'$ is ϵ-DP.*

The privacy guarantee of a DP algorithm can be amplified by a preceding sampling step (Kasiviswanathan et al., 2011; Li et al., 2012). Let $\mathcal{A}$ be an ϵ-DP algorithm and $\mathcal{D}$ be a data set. Let $\mathcal{A}'$ be an algorithm that runs $\mathcal{A}$ on a random subset of $\mathcal{D}$ obtained by sampling it with probability β.

Lemma 2.3 (Privacy Amplification)**.** *Algorithm $\mathcal{A}'$ will satisfy ϵ'-DP where $\epsilon' = ln(1 + \beta(e^{\epsilon} - 1))$*

The **Laplace mechanism** is a standard algorithm to achieve differential privacy (Dwork & Roth, 2014). In this mechanism, in order to output $f(\mathcal{D})$ where $f : \mathcal{D} \mapsto R$, an ϵ-DP algorithm $\mathcal{A}$ publishes $f(\mathcal{D}) + Lap\Big(\frac{\Delta f}{\epsilon}\Big)$ where $\Delta f = \max_{\mathcal{D},\mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||_1$ is known as the sensitivity of the function. The probability density function of $Lap(b)$ is given by $\mathbf{f}(x) = \frac{1}{2b} e^{\left(-\frac{|x-\mu|}{b}\right)}$. The sensitivity of the function f is the maximum magnitude by which an individual's data can change f. The sensitivity of counting queries is 1.

Next, we define two terms, namely marginal table and mutually consistent marginal tables, that are used in Sec. 3.3.

Let $\mathcal{D}$ be a data set defined over attributes $\mathcal{X}$ and $\mathbf{X}$ be an attribute set such that $\mathbf{X} = \{X_1, \cdots, X_k\}, \mathbf{X} \subseteq \mathcal{X}$. Let $s = \prod_{i=1}^{k} |dom(X_i)|$ and $dom(\mathbf{X}) = \{v_1, \cdots, v_s\}$ represent the domain of $\mathbf{X}$. The **marginal table** for the attribute set $\mathbf{X}$ denoted by $M_{\mathbf{X}}$, is computed as follows:

(1) Populate the entries of table $T_{\mathbf{X}}$ of size s from $\mathcal{D}$ such that each entry $j \in [s], T_{\mathbf{X}}[j] = \#$ records in $\mathcal{D}$ with $\mathbf{X} = v_j$. This step is also called *materialization*.

(2) Compute $M_{\mathbf{X}}$ from $T_{\mathbf{X}}$ such that $M_{\mathbf{X}}[j] = T_{\mathbf{X}}[j] / \sum_{i=1}^{s} T_{\mathbf{X}}[i], j \in [s]$.

Thus the entries of $M_{\mathbf{X}}$ corresponds to the values of the joint probability distribution over the attributes in $\mathbf{X}$.

Let $Attr(M)$ denote the set of attributes on which a marginal table M is defined and $M_1 \equiv M_2$ denote that the two marginal tables have the same values for every entry.

Definition 2.3 (Mutually Consistent Marginal Tables)**.** Two noisy marginal tables $\tilde{M}_i$ and $\tilde{M}_j$ are defined to be mutually consistent iff the marginal table over the attributes in $Attr(\tilde{M}_i) \cap Attr(\tilde{M}_j)$ reconstructed from $\tilde{M}_i$ is exactly the same as the one reconstructed from $\tilde{M}_j$, i.e.,

$$\tilde{M}_i[Attr(\tilde{M}_i) \cap Attr(\tilde{M}_j)] \equiv \tilde{M}_j[Attr(\tilde{M}_i) \cap Attr(\tilde{M}_j)] \quad (4)$$

3. Data-Dependent Differentially Private Parameter Learning for DGMs

In this section, we describe our proposed solution for differentially private learning of the parameters of a fully observed DGM by adding data and structure dependent noise.

3.1. Problem Setting

Let $\mathcal{D}$ be a sensitive data set of size m with attributes $\mathcal{X} = \langle X_1, \cdots, X_n \rangle$ and let $\mathcal{N} = \langle \mathcal{G}, \Theta \rangle$ be the DGM of interest. The graph structure $\mathcal{G}$ of $\mathcal{N}$ defined over the attribute set $\mathcal{X}$ is publicly known. Our goal is to *learn the parameters Θ, i.e., the CPDs of $\mathcal{N}$, in a data-dependent differentially private manner from $\mathcal{D}$ such that the error in inference queries over the ϵ-DP DGM is minimized*.

3.2. Key Ideas

Our solution is based on the following two key observations:

(1) The parameters $\Theta[X_i|X_{pa_i}]$ of the DGM $\mathcal{N}$ can be estimated separately via *counting queries over the empirical marginal table* of the attribute set $X_i \cup X_{pa_i}$.

(2) The *factorization over $\mathcal{N}$ decomposes the overall ϵ-DP learning problem into a set of separate ϵ-DP learning subproblems (one for each CPD)*. For example, for the DGM in Fig. 1, the following six CPDs have to be learned separately $\{P[A], P[B], P[C|A,B], P[D|C], P[E|C], P[F|D,E]\}$. Thus the total privacy budget has to be divided among these subproblems. However, due to the structure of the graph and the data set, some nodes will have more impact on inference queries than others. Hence, allocating more budget (and thus, getting better accuracy) to these nodes will result in reduced overall error for the inference queries.

Our method is outlined in Alg. 1 and proceeds in two stages. In the first stage, we obtain preliminary noisy measurements of the parameters of $\mathcal{N}$ which are used along with some graph specific properties (the height and out-degree of each node) to formulate a data-dependent optimization objective for privacy budget allocation. The solution of this objective is then used in the second stage to compute the final parameters. For instance, for the DGM in Fig. 1, node A (root node) would typically have a higher privacy budget than node F (leaf node). In summary, if ϵ^B is the total privacy budget available, we spend ϵ^I to obtain preliminary parameter measurements in Stage I and the remaining $\epsilon^B - \epsilon^I$ is used for the final parameter computation in Stage II, after optimal allocation across the marginal tables. As a result, our scheme only requires a privacy budget of $\epsilon = 1.0$ to yield the same utility that a standard data-independent method achieves with $\epsilon = 3.0$ (Sec. 5).

Next, we describe our algorithm in detail and highlight how we address the two core technical challenges in our solution:

(1) how to reduce the privacy budget cost for the first stage ϵ^I (equivalently increase ϵ^B-ϵ^I) (Alg. 1, Lines 1-3), and

(2) what properties of the data set and the graph should the optimization objective be based on (Alg. 1, Lines 5-11).

3.3. Algorithm Description

We now describe the two stages of our technique:

Stage I – Formulation of optimization objective: First, we handle the trade-off between the two parts of the total privacy budget ϵ^I and $\epsilon^B - \epsilon^I$. While we want to maximize $\epsilon^B - \epsilon^I$ to reduce the amount of noise in the final parameters, sufficient budget ϵ^I is required to obtain good estimates of the statistics of the data set to form the data-dependent budget allocation objective. To handle this trade-off, we use the sampling strategy from Lemma 2.3 to improve the accuracy of the optimization objective computation (Alg. 1, Lines 1-2). Specifically, for sufficiently large values of m and sampling rate β, we have $\sqrt{(\beta(1-\beta)/n)} + 2\sqrt{(2)}/\epsilon < 2\sqrt{(2)}/ln(1+\beta(e^{\epsilon}-1))$ where L.H.S and R.H.S is the expected error with and without sampling, respectively. This allows us to assign a relatively low value to ϵ^I increasing our budget for the final parameter computation.

Next, we estimate the parameters $\widehat{\Theta}$ on the sampled data set $\mathcal{D}'$ via the procedure $ComputeParameters$ (described below and outlined in Procedure 1) using budget allocation $\mathcal{E}$ (Alg. 1, Lines 3-4). Note that $\widehat{\Theta}$ is only required for the optimization objective formulation and is different from the final parameters $\widetilde{\Theta}$ (Alg. 1, Line 18). Hence, for $\widehat{\Theta}$ we use a naive allocation policy of equal privacy budget for all tables.

Finally, we compute the privacy budget optimization objective $\mathcal{F}_{\mathcal{D},\mathcal{G}}$ that depends on the data set $\mathcal{D}$ and graph structure $\mathcal{G}$ (Alg. 1, Line 5-12). The details are discussed in Sec. 3.4.

Stage II – Final parameter computation: We solve for the optimal privacy budget allocation $\mathcal{E}^*$ from $\mathcal{F}_{\mathcal{D},\mathcal{G}}$ and use it to compute a copy of the parameters $\overline{\Theta}$ (Alg. 1, Lines 13-14). We obtain the final parameters $\widetilde{\Theta}$ by computing the weighted average of the corresponding values in $\overline{\Theta}$ and the preliminary estimate $\widehat{\Theta}$ (Alg. 1, Lines 15-20). Note that $\mathcal{E}^*[i] + \epsilon^I/n$ is the total privacy budget spent on the CPD of node X_i in the two rounds.

Algorithm 1 Differentially private learning of the parameters of a directed graphical model

Input: $\mathcal{D}$ - Input data set of size m with attributes $\mathcal{X} = \langle X_1, \cdots, X_n \rangle$;
$\mathcal{G}$ - Graph Structure of DGM $\mathcal{N}$;
ϵ^B - Total privacy budget;
β - Sampling rate for Stage I;
ϵ^I - Privacy budget for Stage I

Output: $\widetilde{\Theta}$ - Noisy parameters of $\mathcal{N}$

Stage I: Optimization objective formulation for privacy budget allocation

1: $\epsilon = \ln\left(\frac{e^{\epsilon^I}-1}{\beta} + 1\right)$ ▷ Computing privacy parameter
2: Construct a new dataset $\mathcal{D}'$ by sampling $\mathcal{D}$ with probability β
3: $\mathcal{E} = [\frac{\epsilon}{n}, \cdots, \frac{\epsilon}{n}]$
4: $\widehat{\Theta}, \widehat{T}, \widehat{T}_{pa} = ComputeParameters(\mathcal{D}', \mathcal{E})$
5: **for** $i = 1$ to n
6: $\widetilde{\delta}_i = ComputeError(i, \widehat{T}_i, \widehat{T}_{pa_i})$ ▷ Estimating error of parameters for X_i using Eq. (12)
7: h_i = Height of node X_i
8: o_i = Out-degree of node X_i
9: $\widetilde{\Delta}_i^{\mathcal{N}} = ComputeSensitivity(i, \widehat{\Theta})$ ▷ Computing sensitivity of the parameters using Eq. (7)
10: $W[i] = (h_i + 1) \cdot (o_i + 1) \cdot (\widetilde{\Delta}_i^{\mathcal{N}} + 1)$
11: **end for**
12: $\mathcal{F}_{\mathcal{G},\mathcal{D}} = \sum_{i=1}^{n-1} W[i] \cdot \widetilde{\delta}_i/\epsilon_i + W[n] \cdot \widetilde{\delta}_n/(\epsilon^B - \epsilon^I - \sum_{i=1}^{n-1}\epsilon_i)$ ▷ Optimization Objective

Stage II: Final computation of the parameter Θ

13: Solve for $\mathcal{E}^* = \{\epsilon_i^*\}$ from minimizing $\mathcal{F}_{\mathcal{G},\mathcal{D}}$ using Eq. (13)
14: $\overline{\Theta}, \overline{T}, \overline{T}_{pa} = ComputeParameters(\mathcal{D}, \mathcal{E}^*)$
15: $\widetilde{\Theta} = \varnothing$
16: **for** $X_i \in \mathcal{X}$
▷ Assuming $\widehat{\Theta} = \bigcup_{X_i \in \mathcal{X}} \widehat{\Theta}[X_i|X_{pa_i}]$ and $\overline{\Theta} = \bigcup_{X_i \in \mathcal{X}} \overline{\Theta}[X_i|X_{pa_i}]$
17: $\widehat{\epsilon_i} = (\epsilon^I/n)/(\mathcal{E}^*[i] + \epsilon^I/n), \overline{\epsilon_i} = \mathcal{E}^*[i]/(\mathcal{E}^*[i] + \epsilon^I/n)$
18: $\widetilde{\Theta}[X_i|X_{pa_i}] = \widehat{\epsilon_i} \cdot \widehat{\Theta}[X_i|X_{pa_i}] + \overline{\epsilon_i} \cdot \overline{\Theta}[X_i|X_{pa_i}]$ ▷Weighted Mean
19: $\widetilde{\Theta} = \widetilde{\Theta} \bigcup \widetilde{\Theta}[X_i|X_{pa_i}]$
20: **end for**
21: Return $\widetilde{\Theta}$

Procedure 1 $ComputeParameters$**:** The goal of this procedure is, *given a privacy budget allocation $\mathcal{E}$, to derive the parameters of $\mathcal{N}$ under DP*. First, we materialize the tables for the attribute sets $X_i \cup X_{pa_i}$ and $X_{pa_i} i \in [n]$ (Proc. 1, Line 2), and then inject noise drawn from $Lap(\frac{2}{\mathcal{E}[i]})$ (using half of the privacy budget $\frac{\mathcal{E}[i]}{2}$ for each table) into each of their cells (Proc. 1, Line 3) to generate $\widetilde{T}_i$ and $\widetilde{T}_{pa_i}$ respectively. Next, we convert $\widetilde{T}_i$ and $\widetilde{T}_{pa_i}$ to a marginal table $\widetilde{M}_i$, i.e.,

Procedure 1 $ComputeParameters$

Input: $\mathcal{D}$ - Data set with attributes $\mathcal{X} = \langle X_1, \cdots, X_n \rangle$;
$\mathcal{E}[1, \cdots, n]$ - Array of privacy budget allocation

1: **for** $i = 1$ to n
2: Materialize tables T_i and T_{pa_i} for the attribute sets $X_i \cup X_{pa_i}$ and X_{pa_i} respectively for $\mathcal{D}$
3: Add noise $\sim Lap(\frac{2}{\mathcal{E}[i]})$ to each entry of T_i and T_{pa_i} to generate noisy tables $\widetilde{T}_i$ and $\widetilde{T}_{pa_i}$ respectively
4: Convert $\widetilde{T}_i$ into noisy marginal table $\widetilde{M}_i$ using $\widetilde{T}_{pa_i}$
5: **end for**
6: $MutualConsistency(\bigcup_{X_i \in \mathcal{X}} \widetilde{M}_i)$
▷Ensures mutual consistency (Def. 2.3) among the noisy marginal tables sharing subsets of attributes
7: **for** $i = 1$ to n
8: Construct $\widetilde{\Theta}[X_i|X_{pa_i}]$ from $\widetilde{M}_i$ ▷ Using Eq. 2
9: **end for**
10: Return $\widetilde{\Theta} = \bigcup_{X_i \in \mathcal{X}} \widetilde{\Theta}[X_i|X_{pa_i}], \widetilde{T} = \bigcup_{X_i \in \mathcal{X}} \widetilde{T}_i, \widetilde{T}_{pa} = \bigcup_{X_i \in \mathcal{X}} \widetilde{T}_{pa_i}$

joint distribution $P_{\mathcal{N}}[X_i, X_{pa_i}]$ (Proc. 1, Line 4) as

$$\widetilde{M}_i[x_i, x_{pa_i}] = \frac{T[x_i, x_{pa_i}]/T_{pa_i}[x_{pa_i}]}{\sum\limits_{v \in dom(X_i)} T[v, x_{pa_i}]/T_{pa_i}[x_{pa_i}]}$$

The denominator in the above equation is for normalization. This is followed by ensuring that all $\widetilde{M}_i$s are mutually consistent (Def. 2.3) on all the attribute subsets (Proc. 1, Line 6). For this, we follow the techniques outlined in (Hay et al., 2010a; Qardaji et al., 2014) and further described in Appx. 8.2.1. Finally, we derive $\widetilde{\Theta}[X_i|X_{pa_i}]$ (the noisy estimate of $P_{\mathcal{N}}[X_i|X_{pa_i}]$) from $\widetilde{M}_i$ (Proc. 1, Lines 7-10). Note that although $\widetilde{M}_i$ could have been derived from $\widetilde{T}_i$ alone, we also use $\widetilde{T}_{pa_i}$ for its computation to ensure independence of the added noise in Eq. (5).

3.4. Optimal Privacy Budget Allocation

Our goal is to find the optimal privacy budget allocation over the marginal tables, $\widetilde{M}_i, i \in [n]$ for $\mathcal{N}$ such that the error in the subsequent inference queries on $\mathcal{N}$ is minimized.

Observation I: *A more accurate estimate of the parameters of $\mathcal{N}$ will result in better accuracy for the subsequent inference queries.* Hence, we focus on reducing the total error of the parameters of $\mathcal{N}$. From Eq. (2) and our Laplace noise injection (Proc. 1, Line 3), for a privacy budget of ϵ, the value of a parameter of the DGM computed from the noisy marginal tables $\widetilde{M}_i$ is expected to be

$$\widetilde{\Theta}[x_i|x_{pa_i}] = \Big(\underbrace{C[x_i, x_{pa_i}]}_{\substack{\text{True count for records} \\ \text{with } X_i = x_i \\ \text{and } X_{pa_i} = x_{pa_i}}} \pm \underbrace{\frac{2\sqrt{2}}{\epsilon}}_{\substack{\text{Noise due to} \\ \text{Laplace} \\ \text{mechanism}}}\Big) \Big/ \Big(C[x_{pa_i}] \pm \frac{2\sqrt{2}}{\epsilon}\Big) \quad (5)$$

Thus, from the rules of standard error propagation (err), the error in $\Theta[x_i, x_{pa_i}]$ is

$$\delta_{\Theta[x_i, x_{pa_i}]} = \Theta[x_i, x_{pa_i}]\sqrt{8/(\epsilon \cdot C[x_{pa_i}])^2 + 8/(\epsilon \cdot C[x_i, x_{pa_i}])^2} \quad (6)$$

where $C[x_i]$ denotes the number of records in $\mathcal{D}$ with $X_i = x_i$. Hence, the mean error for the parameters of X_i is

$$\delta_i = \frac{1}{|dom(X_i \cup X_{pa_i})|} \sum_{x_i, x_{pa_i}} \delta_{\Theta[x_i|x_{pa_i}]}$$

where $dom(S)$ is the domain of the attribute set S. Since using the true counts, $C[x_i]$, would violate privacy, Alg. 1 uses the noisy estimates from $\widetilde{T}_i$ and $\widetilde{T}_{pa_i}$ (Alg. 1, Line 6).

Observation II: *Depending on the data set and the graph structure, different nodes will have different impact on inferencing.* This information can be captured by a corresponding weighting coefficient $W[i], i \in [n]$ for each node.

Computation of weighting coefficient $W[i]$: For a given node X_i, the weighting coefficient $W[i]$ is computed from the following three node features:

(1) **Height of the node h_i:** The height of a node X_i is the length of the longest path between X_i and a leaf node. Due to the factorization of the joint distribution over a DGM, the marginal probability distribution of a node depends only on the set of its ancestor nodes (as is explained in the following discussion on the computation of sensitivity). Thus, a node with large height will affect the inference queries on more nodes (all its successors) than say a leaf node.

(2) **Out-degree of the node o_i:** A node causally affects all its children nodes. Thus the impact of a node with high out-degree on inferencing will be more than say a leaf node.

(3) **Sensitivity $\Delta_i^{\mathcal{N}}$:** Sensitivity of a parameter in a DGM measures the impact of small changes in the parameter value on a target probability. Laskey (Laskey, 1995) proposed a method of computing sensitivity by using the partial derivative of output probabilities with respect to the parameter being varied. However, previous works have mostly focused on the target probability to be a joint distribution of all the variables. In this paper, we present a method to compute sensitivity by targeting the probability distribution of child nodes only. Let $\Delta_i^{\mathcal{N}}$ denote the mean sensitivity of the parameters of X_i on target probabilities of all the nodes in $Child(X_i)$= {set of all the child nodes of X_i}. Formally,

$$\Delta_i^{\mathcal{N}} = \frac{1}{|dom(X_i \cup X_{pa_i})|}\Big(\sum_{x_i, x_{pa_i}} \frac{1}{|Child(X_i)|}\Big(\underbrace{\sum_{Y \in Child(X_i)} \frac{1}{|dom(Y)|}\Big(\sum_y \frac{\partial P_{\mathcal{N}}[Y=y]}{\partial \Theta[x_i|x_{pa_i}]}\Big)\Big)\Big)}_{\substack{\text{computing the partial derivatives} \\ \text{of the parameters of the child nodes only}}} \quad (7)$$

A node X_i can affect another node Y only iff it is in its Markov blanket (Defn. 2.1), i.e., $Y \in \mathcal{P}(X_i)$. However due to the factorization of the joint distribution over a DGM, $\forall Y \in \mathcal{P}(X_i), Y \notin Child(X_i), P_{\mathcal{N}}[Y]$ can be expressed without $\Theta[x_i|x_{pa_i}]$. Thus just computing the mean sensitivity of the parameters over the set of child nodes $\Delta_i^{\mathcal{N}}$ turns out to be a good weighting metric for our setting. $\Delta_i^{\mathcal{N}}$ for leaf nodes is thus 0. Note that $\Delta_i^{\mathcal{N}}$ is distinct from the notion of sensitivity of a function in the Laplace mechanism (Sec. 2).

Computing Sensitivity $\Delta_i^{\mathcal{N}}$: Let $Y \in Child(X_i)$ and $\Gamma(Y) = \{\mathbf{Y}_1, \cdots, \mathbf{Y}_t\}, t < n$ denote the set of all nodes such that there is a directed path from $\mathbf{Y}_i$ to Y. In other words $\Gamma(Y)$ denotes the set of ancestors of Y in $\mathcal{G}$. From the factorization of the joint distribution over $\mathcal{N}$, we have

$$P_{\mathcal{N}}[\mathbf{Y}_1, \cdots, \mathbf{Y}_t, Y] = P_{\mathcal{N}}[Y|Y_{pa}] \cdot \prod_{\mathbf{Y}_i \in \Gamma(Y)} P_{\mathcal{N}}[\mathbf{Y}_i|\mathbf{Y}_{pa_i}]$$
$$P_{\mathcal{N}}[Y] = \sum_{\mathbf{Y}_1} \dots \sum_{\mathbf{Y}_t} P_{\mathcal{N}}[\mathbf{Y}_1, \cdots, \mathbf{Y}_t, Y]$$

Therefore, using our noisy preliminary parameter estimates (Alg. 1, Line 4), we compute

$$\frac{\partial \widetilde{P}_{\mathcal{N}}[Y=y]}{\partial \Theta[x_i|x_{pa_i}]} = \sum_{\substack{\mathbf{y}_i \in dom(\mathbf{Y}_i), \\ \mathbf{y}_{pa_i} \in dom(\mathbf{Y}_{pa_i}), \\ \mathbf{Y}_i \in \Gamma(Y)}} \Big(\prod_{\mathbf{Y}_i \in \Gamma(Y)} \widehat{\Theta}[\mathbf{Y}_i = \mathbf{y}_i | \mathbf{Y}_{pa_i} \mathbf{y}_{pa_i}] \cdot \underset{x_i, x_{pa_i}}{\zeta} (\mathbf{Y}_i = \mathbf{y}_i, \mathbf{Y}_{pa_i} = \mathbf{y}_{pa_i}) \Big) \cdot \widehat{\Theta}[Y = y|Y_{pa} = y_{pa}] \cdot \underset{x_i, x_{pa_i}}{\zeta} (Y_{pa} = y_{pa})$$

$$\underbrace{\underset{x_i, x_{pa_i}}{\zeta} (Z_1 = z_1, \cdots, Z_t = z_t)}_{\substack{\text{indicator variable to ensure} \\ \text{only relevant terms are retained}}} = \begin{cases} 1 & \text{if } \bigcup_{i=1}^{t} Z_i \bigcap \{X_i \cup X_{pa_i}\} = \varnothing \\ 1 & \text{if } \forall Z_i, Z_i \in \{X \cup X_{pa_i}\} \Rightarrow \\ & \quad z_i \in \{x, x_{pa_i}\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\zeta_{x_i, x_{pa_i}}$ is an indicator variable which ensures that only the product terms $\prod_{\mathbf{Y}_i \in \Gamma(Y)} \widehat{\Theta}[y_i|y_{pa_i}]$ involving parameters with attributes in $\{X_i, X_{pa_i}, Y\}$ that match up with the corresponding values in $\{x_i, x_{pa_i}, y\}$ are retained in the computation (as all others terms have partial derivative 0). Thus, the noisy mean sensitivity estimate for the parameters of node X_i, $\widetilde{\Delta}_i^{\mathcal{N}}$, can be computed from Eq. (7) and (8).

Optimization Objective: Let ϵ_i denote the privacy budget for node X_i. Thus from the above discussion, the optimization objective $\mathcal{F}_{\mathcal{D},\mathcal{G}}$ is formulated as a weighted sum of the parameter error and the optimization problem is given by

$$\begin{aligned} &\underset{\epsilon_i}{\textbf{minimize}} \quad \mathcal{F}_{\mathcal{D},\mathcal{G}} \\ &\textbf{subject to} \quad \epsilon_i > 0, \forall i \in [n] \end{aligned} \tag{9}$$

$$\mathcal{F}_{\mathcal{D},\mathcal{G}} = \Big(\sum_{i=1}^{n-1} \underbrace{W[i]}_{\text{weight}} \cdot \underbrace{\frac{\widetilde{\delta}_i}{\epsilon_i}}_{\text{mean error}} + W[n] \cdot \frac{\widetilde{\delta}_n}{\epsilon^B - \epsilon^I - \sum_{i=1}^{n-1} \epsilon_i} \Big) \tag{10}$$

$$W[i] = (\underbrace{h_i}_{\text{height}} + 1) \cdot (\underbrace{o_i}_{\text{out-degree}} + 1) \cdot (\underbrace{\widetilde{\Delta}_i^{\mathcal{N}}}_{\text{sensitivity}} + 1) \tag{11}$$

$$\widetilde{\delta}_i = \frac{1}{|dom(X_i \cup X_{pa_i})|} \sum_{x_i, x_{pa_i}} \Big(\widehat{\Theta}[x_i|x_{pa_i}] \sqrt{1/\widehat{T}[x_{pa_i}]^2 + 1/\widehat{T}[x_i, x_{pa_i}]^2} \Big) \tag{12}$$
$$\widehat{T}[x_i, x_{pa_i}], \widehat{T}[x_{pa_i}] \in \widehat{T}_i$$

where h_i and o_i are the height and out-degree of the node X_i respectively, $\Delta_i^{\mathcal{N}}$ is the sensitivity of the parameters of X_i, $\frac{\widetilde{\delta}_i}{\epsilon_i}$ gives the measure for estimated mean error for the parameters (CPD) of X_i and the denominator of the last term of Eq. (10) captures the linear constraint $\sum_{i=1}^{n} \epsilon_i = \epsilon^B - \epsilon^I$. As stated by Eq. (11), the weighting coefficient $W[i]$ is defined as the product of the aforementioned three features. The extra additive term 1 is used to handle leaf nodes to ensure non-zero weighting coefficients. Let ϵ_i^* denote the optimal privacy budget for node X_i. The objective $\mathcal{F}_{\mathcal{D},\mathcal{G}}$ has a closed form solution as follows

$$c_j = 1/(W[j] \cdot \widetilde{\delta}_j), j \in [n], \epsilon^{II} = \epsilon^B - \epsilon^I$$
$$\epsilon_i^* = \frac{\epsilon^{II} \prod_{j=1, j \neq i}^{n} \sqrt{c_j}}{\sum_{j \in [n]} \prod_{l=1, l \neq j}^{n} \sqrt{c_l}}, i \in [n-1], \epsilon_n^* = \epsilon^{II} - \sum_{i=1}^{n-1} \epsilon_i^* \tag{13}$$

Discussion: There are two sources of information to be considered for a DGM - (1) graph structure $\mathcal{G}$ (2) data set $\mathcal{D}$. h_i and o_i are purely graph characteristics that summarise the graphical properties of the node X_i. $\widetilde{\Delta}_i^{\mathcal{N}}$ captures the interactions of the graph structure with the actual parameter values thereby encoding the data set dependent information. Hence, we theorize that the aforementioned three features are sufficient for constructing the weighting coefficients.

Also note that it is trivial to modify our proposed algorithm to allow estimation with Dirichlet priors (the most popular choice for a prior (Koller & Friedman, 2009)). Specifically, R.H.S of Eq. (2) changes to $(C[x_i, x_{pai}] + \alpha_k)/(C[x_{pai}] + \sum_k(\alpha_k)$ where α_k are the parameters of the publicly known prior.

Illustration of Algorithm 1: Here we illustrate Alg. 1 on the example DGM of Fig. 1. The parameters Θ of this DGM are the CPDs $\{P[A], P[B], P[C|A, B], P[D|C], P[E|C], P[F|D, E]\}$. Thus we need to construct 6 marginal tables over the attribute sets $\langle\{A\}, \{B\}, \{C, A, B\}, \{D, C\}, \{E, C\}, \{F, D, E\}\rangle$. First, we compute a preliminary estimate of the above parameters from a sampled dataset $\mathcal{D}'$ (Alg. 1, Line 1-4). For this, we need to ensure mutual consistency between $\widetilde{M}_A$ and $\widetilde{M}_C$ on attribute A, $\widetilde{M}_B$ and $\widetilde{M}_C$ on attribute B and so on (Proc. 1, Line 6). This is followed by the formulation of $\mathcal{F}_{\mathcal{D},\mathcal{G}}$. (Alg. 1, Line 5-12). Here we show the computation of $W[i]$ for node A. For simplicity, we assume binary attributes. $h_A = 3$ and $o_A = 1$ trivially. For $\Delta_A^{\mathcal{N}}$, we need to compute the sensitivity of the parameters of A on the target probability of C, i.e., $\frac{\partial P[C=0]}{\partial \Theta[A=0]}$, $\frac{\partial P[C=0]}{\partial \Theta[A=1]}$, $\frac{\partial P[C=1]}{\partial \Theta[A=1]}$, and $\frac{\partial P[C=1]}{\partial \Theta[A=0]}$ which is computed as $\frac{\partial \widetilde{P}[C=0]}{\partial \Theta[A=0]} = \widehat{\Theta}[B=0]\widehat{\Theta}[C=0|A=0, B=0] + \widehat{\Theta}[B=1]\widehat{\Theta}[C=0|A=0, B=1]$. The rest of the partial derivatives are computed in a similar manner to give us

$$\Delta_A^{\mathcal{N}} = \frac{1}{4} \Big(\frac{\partial P[C=0]}{\partial \Theta[A=0]} + \frac{\partial P[C=0]}{\partial \Theta[A=1]} + \frac{\partial P[C=1]}{\partial \Theta[A=1]} + \frac{\partial P[C=1]}{\partial \Theta[A=0]} \Big)$$

Finally, we use the solution of $\mathcal{F}_{\mathcal{D},\mathcal{G}}$ to compute the final parameters (Alg. 1, Line 13-21).

3.5. Privacy Analysis

Theorem 3.1. *The proposed algorithm (Alg. 1) for learning the parameters of a DGM with a publicly known graph structure over fully observed data is ϵ^B-DP.*

The proof of the above theorem follows from Thms. 2.1 and 2.2 and is presented in Appx. 8.2.2. The DGM learned via our algorithm can be released publicly and any inference query run on it will still be ϵ^B-DP (Thm. 2.2).

4. Error Analysis for Inference Queries

As discussed in Sec. 3.4, our optimization objective minimizes a weighted sum of the parameter errors. To understand how the error propagates from the parameters to the inference queries, we present two general results bounding the error of a sum-product term of the VE algorithm, given the errors in the factors.

Theorem 4.1. ***[Lower Bound]*** *For a DGM $\mathcal{N}$, for any sum-product term of the form $\phi_{\mathcal{A}} = \sum_x \prod_{i=1}^t \phi_i, t \in \{2, \cdots, \eta\}$ in the VE algorithm,*

$$\delta_{\phi_{\mathcal{A}}} \geq \sqrt{\eta - 1} \cdot \delta^{min}_{\phi_i[a,x]} (\phi_i^{min}[a, x])^{\eta-2} \quad (14)$$

where X is the attribute being eliminated, δ_ϕ denotes the error in the factor ϕ, $Attr(\phi)$ is the set of attributes in ϕ, $\mathcal{A} = \bigcup_{\phi_i}\{Attr(\phi_i)\}/X$, $x \in dom(X)$, $a \in dom(\mathcal{A})$, $\phi[a, x]$ denotes that $Value(Attr(\phi)) \in \{a\} \wedge X = x$, $\delta^{min}_{\phi_i[a,x]} = min_{i,a,x}\{\delta_{\phi_i[a,x]}\}$, $\phi_i^{min}[a, x] = min_{i,a,x}\{\phi_i[a, x]\}$ and $\eta = \max_{X_i}\{$in-degree$(X_i)+$ out-degree$(X_i)\} + 1$.

Theorem 4.2. ***[Upper Bound]*** *For a DGM $\mathcal{N}$, for any sum-product term of the form $\phi_{\mathcal{A}} = \sum_x \prod_{i=1}^t \phi_i$, $t \in \{2, \cdots, n\}$ in the VE algorithm with the optimal elimination order,*

$$\delta_{\phi_{\mathcal{A}}} \leq 2 \cdot \eta \cdot d^\kappa \delta^{max}_{\phi_i[a,x]} \quad (15)$$

where X is the attribute being eliminated, δ_ϕ denotes the error in the factor ϕ, κ is the treewidth of $\mathcal{G}$, d is the maximum domain size of an attribute, $Attr(\phi)$ is the set of attributes in ϕ, $\mathcal{A} = \bigcup_i^t\{Attr(\phi_i)\}/X$, $a \in dom(\mathcal{A})$, $x \in dom(X)$, $\phi[a, x]$ denotes that $Value(Attr(\phi)) \in \{a\} \wedge X = x$, $\delta^{max}_{\phi_i[a,x]} = \max_{i,a,x}\{\delta_{\phi_i[a,x]}\}$ and $\eta = \max_{X_i}\{$in-degree$(X_i) +$ out-degree$(X_i)\} + 1$.

For proving the lower bound, we introduce a specific instance of the DGM based on Lemma 8.1 (Appx. 8.1.1). For the upper bound, with the optimal elimination order of the VE algorithm, the maximum error has an exponential dependency on the treewidth κ. For example, for the DGM in Fig. 1, the maximum error is bounded by $2 \cdot \eta \cdot d^2 \cdot \delta^{max}_{\phi_i[a,x]}$. This is very intuitive as even the complexity of the VE algorithm has the same dependency on κ. The answer of a marginal inference query is the factor generated from the last sum-product term. Also, since the initial set of ϕ_is for the first sum-product term computation are the actual parameters of the DGM, all the errors in the subsequent intermediate factors and hence the inference query itself can be bounded by functions of parameter errors using the above theorems.

5. Evaluation

We evaluate the utility of the DGM learned via our algorithm by studying the following three questions:

(1) Does our scheme lead to low error estimation of the DGM parameters?

(2) Does our scheme result in low error inference query responses?

(3) How does our scheme fare against data-independent approaches?

Evaluation Highlights: First, focusing on the parameters of the DGM, we find that our scheme achieves low L1 error (at most 0.2 for $\epsilon = 1$) and low KL divergence (at most 0.13 for $\epsilon = 1$) across all test data sets. Second, we find that for marginal and conditional inferences, our scheme provides low L1 error and KL divergence (both around 0.05 at max for $\epsilon = 1$) for all test data sets. Our scheme also provides high accuracy for MAP queries (93.5% accuracy for $\epsilon = 1$ averaged over all test data sets). Finally, our scheme achieves strictly better utility than the data-independent baseline; our scheme only requires a privacy budget of $\epsilon = 1.0$ to yield the same utility that the data-independent baseline achieves with $\epsilon = 3.0$.

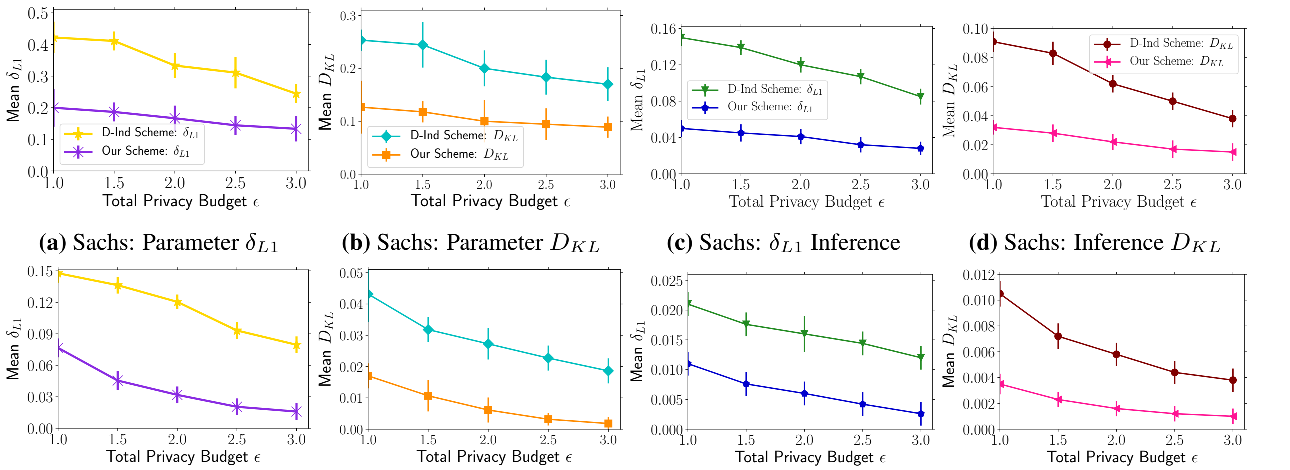
5.1. Experimental Setup

Data sets: We evaluate our proposed scheme on four benchmark DGMs (BN) namely *Asia, Sachs, Child* and *Alarm*. For all four DGMs, the evaluation is carried out on corresponding synthetic data sets (D1, 2014; D2) with 10K records each. These data sets are standard benchmarks for evaluating DGM inferencing and are derived from real-world use cases (BN). Due to space constraints, we present the results for only two of the data sets (Sachs and Child) here; the rest are presented in Appx. 9.1. The details of Sachs and Child are as follows:

Sachs: Number of nodes – 11; Number of arcs – 17; Number of parameters – 178

Child: Number of nodes – 20; Number of arcs – 25; Number of parameters – 230

Baseline: We compare our results with a data-independent baseline (denoted by *D-Ind*) which corresponds to executing Proc. 1 on the entire input data set $\mathcal{D}$ with privacy budget



(a) Sachs: Parameter δ_{L1} **(b)** Sachs: Parameter D_{KL} **(c)** Sachs: δ_{L1} Inference **(d)** Sachs: Inference D_{KL}

(e) Child: Parameter δ_{L1} **(f)** Child: Parameter D_{KL} **(g)** Child: δ_{L1} Inference **(h)** Child: Inference D_{KL}

Figure 2: Parameter and Inference (Marginal and Conditional) Error Analysis: We observe that our scheme only requires a privacy budget of $\epsilon = 1.0$ to yield the same utility that *D-Ind* achieves with $\epsilon = 3.0$.

Table 1: MAP Inference Error Analysis

ϵ	ρ							
	Asia		**Sachs**		**Child**		**Alarm**	
	D-Ind Scheme	**Our Scheme**	**D-Ind Scheme**	**Our Scheme**	**D-Ind Scheme**	**Our Scheme**	**D-Ind Scheme**	**Our Scheme**
1	0.88	1	0.81	0.86	0.79	0.93	0.89	0.95
1.5	0.93	1	0.87	0.93	0.83	0.95	0.92	0.98
2	1	1	0.92	0.98	0.89	0.97	0.95	1
2.5	1	1	0.96	1	1	1	1	1
3	1	1	1	1	1	1	1	1

Table 2: Error Bound Analysis

	μ			
	Asia	**Sachs**	**Child**	**Alarm**
D-Ind Scheme	0.008	0.065	0.0035	0.0046
Our Scheme	0.0035	0.04	0.0014	0.0012

array $\mathcal{E} = [\frac{\epsilon^B}{n}, \cdots, \frac{\epsilon^B}{n}]$. *D-Ind* is in fact based on (Zhang et al., 2016b) (see Sec. 6 for details) which is the most recent work that explicitly deals with parameter estimation for DGMs. *D-Ind* is also identical (it has an additional consistency step) to an algorithm used in PrivBayes (Zhang et al., 2017) which uses DGMs to generate high-dimensional data. Other candidate baselines from the differential privacy literature include MWEM (Hardt et al., 2012), HDMM (McKenna et al., 2018), Ding et al's work (Ding et al., 2011a) and PriView (Qardaji et al., 2014). However, even with binary attributes, MWEM, (Ding et al., 2011a) and HDMM have run time $O(2^n), O(2^n)$ and $O(4^n)$ respectively for marginal queries in our setting. Our scheme uses a sub protocol ($MutualConsistency()$) from PriView. However, PriView works best for answering all $\binom{n}{k}$ k-way marginals (for $k = 2$ or 3). In contrast, our setting computes only n marginals of varying size (often greater than 3). Hence PriView gives lower accuracy than our baseline as the privacy budget is wasted in computing irrelevant marginals. For example, we have empirically verified that for dataset Sachs, the mean error of the parameters for our baseline is an order smaller than that of PriView.

Metrics: For conditional and marginal inference queries we compute the following two metrics: L1-error, $\delta_{L1} = \sum_{x,y} |P[x|y] - \tilde{P}[x|y]|$and KL divergence, $D_{KL} = \sum_{x,y} \tilde{P}[x|y] \ln\left(\frac{\tilde{P}[x|y]}{P[x|y]}\right)$ where $P[x|y]$ is either a true CPD of the DGM (parameter) or a true marginal/conditional inference query response and $\tilde{P}[x|y]$ is the corresponding noisy estimate obtained from our scheme. For answering MAP inferences, we compute $\rho = \frac{\text{\# Correct answers}}{\text{\#Total runs}}$.

Setup: We evaluate each data set on 20 random inference queries (10 marginal inference, 10 conditional inference) and report mean error over 10 runs. For MAP queries, we run 20 random queries and report the mean result over 10 runs. The queries are of the form $P[X|Y]$ where attribute subsets X and Y are varied from being singletons up to the full attribute set. We compare our results with a standard data-independent baseline (denoted by *D-Ind*) (Zhang et al., 2016b; 2017) which corresponds to executing Procedure 1 on the entire input data set $\mathcal{D}$ and the privacy budget array $\mathcal{E} = [\frac{\epsilon^B}{n}, \cdots, \frac{\epsilon^B}{n}]$. All the experiments have been implemented in Python and we set $e^I = 0.1 \cdot e^B, \beta = 0.1$.

5.2. Experimental Results

Fig. 2 shows the mean δ_{L1} and D_{KL} for the noisy parameters, and the marginal and conditional inferences for the data sets Sachs and Child. The main observation is that our scheme achieves strictly lower error than that of *D-Ind*. Specifically, our scheme only requires a privacy budget of $\epsilon = 1.0$ to yield the same utility that *D-Ind* achieves with $\epsilon = 3.0$. In most practical scenarios, the value of ϵ typically does not exceed 3 (Hsu et al., 2014). In Table 1, we present our experimental results for MAP queries. We see that our scheme achieves higher accuracy than *D-Ind*. For example, our scheme provides an accuracy of at least 86% while *D-Ind* achieves 81% accuracy for $\epsilon = 1$. Finally, given a marginal inference query $\mathcal{Q}$, we compute the scale normalized error in $\mathcal{Q}$ as $\mu = \frac{\delta_{L1}[\mathcal{Q}]-LB}{UB-LB}$ where UB and LB are the upper and lower bound computed using Thm. 4.2 and Thm. 4.1[2] respectively. Clearly, the lower the value of μ, the closer it is to the lower bound and vice versa. We report the mean value of μ for 20 random inference queries (marginal and conditional) for $\epsilon = 1$ in Table 2. We observe that the errors are closer to their respective lower bounds. This is more prominent for the errors obtained from our data-dependent scheme than those of *D-Ind*.

Thus, we conclude that the non-uniform budget allocation in our data-dependent scheme gives better utility than uniform budget allocation. For example, for DGM Asia (Fig. 3), our scheme allocates the privacy budget in the following order: "asia">"smoke">"bronc" >"lung">"tub">"xray"> "either">"dysp". As expected, nodes with greater height and out-degree are assigned higher budget than leaf nodes.

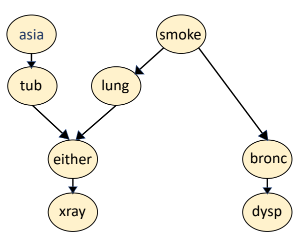


Figure 3: Graph Structure of DGM Asia

6. Related Work

Here we briefly discuss the related literature (see Appx. 9 for a detailed review). There has been a fair amount of work in differentially private Bayesian inferencing (Dimitrakakis et al., 2014; Wang et al., 2015b; Foulds et al., 2016; Zhang et al., 2016b; Geumlek et al., 2017; Bernstein & Sheldon, 2018; Heikkilä et al., 2017; Wang et al., 2015a; Dziugaite & Roy, 2018; Zhang & Li, 2019; Schein et al., 2018; Park et al., 2016b; Jälkö et al., 2016; Barthe et al., 2016; Bernstein et al., 2017; Zhang et al., 2020). All of the above works have different setting/goal from our paper. Specifically, in (Zhang et al., 2016b) the authors propose algorithms for private Bayesian inference on graphical models. However, their proposed solution does not add data-dependent noise. In fact, their proposed algorithms (Alg. 1 and Alg. 2 in (Zhang et al., 2016b)) are essentially the same in spirit as our baseline solution, *D-Ind*. Moreover, some proposals from (Zhang et al., 2016b) can be combined with *D-Ind*. For example, to ensure mutual consistency, (Zhang et al., 2016b) adds Laplace noise in the Fourier domain while *D-Ind* uses techniques of (Hay et al., 2010a). *D-Ind* is also identical (it has an additional consistency step) to an algorithm used in (Zhang et al., 2017) which uses DGMs to generate high-dimensional data. Data-dependent noise addition is a popular technique in differential privacy (Acs et al., 2012; Xu et al., 2012; Zhang et al.; Xiao et al., 2012; Hardt et al., 2012; Li et al., 2014; Kotsogiannis et al., 2017a).

[2] UB and LB are computed separately for each run of the experiment from their respective empirical parameter errors.

7. Conclusion

In this paper, we have proposed an algorithm for differentially private learning of the parameters of a DGM with a publicly known graph structure over fully observed data. The noise added is customized to the private input data set as well as the public graph structure of the DGM. To the best of our knowledge, we propose the first explicit data-dependent privacy budget allocation mechanism in the context of DGMs. Our solution achieves strictly higher utility than that of a standard data-independent approach; our solution requires roughly 3× smaller privacy budget to achieve the same or higher utility.

References

Bn repository. URL http://www.bnlearn.com/bnrepository/.

Data repository. URL https://www.ccd.pitt.edu/wiki/index.php/Data_Repository.

Propagation of errors. URL https://www.lockhaven.edu/~dsimanek/scenario/errorman/propagat.htm.

Data. 2014. URL https://github.com/albertofranzin/data-thesis.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978318. URL http://doi.acm.org/10.1145/2976749.2978318.

Acs, G., Castelluccia, C., and Chen, R. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*, pp. 1–10, Dec 2012. doi: 10.1109/ICDM.2012.80.

Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., and McMahan, B. cpsgd: Communication-efficient and differentially-private distributed sgd. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7564–7575. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7984-cpsgd-communication-efficient-and-differentially-private-distributed-sgd.pdf.

Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., and Talwar, K. 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*, PODS '07, pp. 273–282, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-685-1. doi: 10.1145/1265530.1265569. URL http://doi.acm.org/10.1145/1265530.1265569.

Barthe, G., Farina, G. P., Gaboardi, M., Arias, E. J. G., Gordon, A., Hsu, J., and Strub, P.-Y. Differentially private bayesian programming. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 68–79, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978371. URL http://doi.acm.org/10.1145/2976749.2978371.

Bernstein, G. and Sheldon, D. R. Differentially private bayesian inference for exponential families. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 2919–2929. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7556-differentially-private-bayesian-inference-for-exponential-families.pdf.

Bernstein, G., McKenna, R., Sun, T., Sheldon, D., Hay, M., and Miklau, G. Differentially private learning of undirected graphical models using collective graphical models. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 478–487. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305431.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, July 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2021036.

Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., and Yu, T. Differentially private spatial decompositions. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pp. 20–31, Washington, DC, USA, 2012a. IEEE Computer Society. ISBN 978-0-7695-4747-3. doi: 10.1109/ICDE.2012.16. URL http://dx.doi.org/10.1109/ICDE.2012.16.

Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., and Yu, T. Differentially private spatial decompositions. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pp. 20–31, Washington, DC, USA, 2012b. IEEE Computer Society. ISBN 978-0-7695-4747-3. doi: 10.1109/ICDE.2012.16. URL http://dx.doi.org/10.1109/ICDE.2012.16.

Diakonikolas, I., Hardt, M., and Schmidt, L. Differentially private learning of structured discrete distributions. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2566–2574. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5713-differentially-private-learning-of-structured-discrete-distributions.pdf.

Dimitrakakis, C., Nelson, B., Mitrokotsa, A., and Rubinstein, B. I. P. Robust and private bayesian inference. In Auer, P., Clark, A., Zeugmann, T., and Zilles, S. (eds.), *Algorithmic Learning Theory*, pp. 291–305, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11662-4.

Ding, B., Winslett, M., Han, J., and Li, Z. Differentially private data cubes: Optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pp. 217–228, New York, NY, USA, 2011a. Association for Computing Machinery. ISBN 9781450306614. doi: 10.1145/1989323.1989347. URL https://doi.org/10.1145/1989323.1989347.

Ding, B., Winslett, M., Han, J., and Li, Z. Differentially private data cubes: Optimizing noise sources and consistency. pp. 217–228, 01 2011b. doi: 10.1145/1989323.1989347.

Dwork, C. and Roth, A. *The Algorithmic Foundations of Differential Privacy*, volume 9. August 2014.

Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pp. 51–60, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4244-7. doi: 10.1109/FOCS.2010.12. URL http://dx.doi.org/10.1109/FOCS.2010.12.

Dziugaite, G. K. and Roy, D. M. Data-dependent pac-bayes priors via differential privacy. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8430–8441. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8063-data-dependent-pac-bayes-priors-via-differential-privacy.pdf.

Foulds, J., Geumlek, J., Welling, M., and Chaudhuri, K. 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, Arlington, Virginia, United States, 2016. AUAI Press. ISBN 978-0-9966431-1-5. URL http://dl.acm.org/citation.cfm?id=3020948.3020969.

Geumlek, J., Song, S., and Chaudhuri, K. Rényi differential privacy mechanisms for posterior sampling. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5295–5304, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Gupta, A., Hardt, M., Roth, A., and Ullman, J. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pp. 803–812, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993742. URL http://doi.acm.org/10.1145/1993636.1993742.

Gupta, A., Roth, A., and Ullman, J. Iterative constructions and private data release. In *Proceedings of the 9th International Conference on Theory of Cryptography*, TCC'12, pp. 339–356, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-28913-2. doi: 10.1007/978-3-642-28914-9_19. URL http://dx.doi.org/10.1007/978-3-642-28914-9_19.

Hardt, M., Ligett, K., and McSherry, F. A simple and practical algorithm for differentially private data release. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pp. 2339–2347, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999325.2999396.

Hay, M., Rastogi, V., Miklau, G., and Suciu, D. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3 (1-2):1021–1032, Sep 2010a. ISSN 2150-8097. doi: 10.14778/1920841.1920970. URL http://dx.doi.org/10.14778/1920841.1920970.

Hay, M., Rastogi, V., Miklau, G., and Suciu, D. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1-2):1021–1032, September 2010b. ISSN 2150-8097. doi: 10.14778/1920841.1920970. URL http://dx.doi.org/10.14778/1920841.1920970.

Heikkilä, M., Lagerspetz, E., Kaski, S., Shimizu, K., Tarkoma, S., and Honkela, A. Differentially private bayesian learning on distributed data. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3226–3235. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6915-differentially-private-bayesian-learning-on-distributed-data.pdf.

Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., and Roth, A. Differential privacy: An economic method for choosing epsilon. *2014 IEEE 27th Computer Security Foundations Symposium*, Jul 2014. doi: 10.1109/csf.2014.35. URL http://dx.doi.org/10.1109/CSF.2014.35.

Jälkö, J., Honkela, A., and Dikmen, O. Differentially private variational inference for non-conjugate models. *CoRR*, abs/1610.08749, 2016.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, June 2011. ISSN 0097-5397. doi: 10.1137/090756090. URL http://dx.doi.org/10.1137/090756090.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193, 9780262013192.

Kotsogiannis, I., Machanavajjhala, A., Hay, M., and Miklau, G. Pythia: Data dependent differentially private algorithm selection. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pp. 1323–1337, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-4197-4. doi: 10.1145/3035918.3035945. URL http://doi.acm.org/10.1145/3035918.3035945.

Kotsogiannis, I., Machanavajjhala, A., Hay, M., and Miklau, G. Pythia: Data dependent differentially private algorithm selection. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pp. 1323–1337, New York, NY, USA, 2017b. ACM. ISBN 978-1-4503-4197-4. doi: 10.1145/3035918.3035945. URL http://doi.acm.org/10.1145/3035918.3035945.

Laskey, K. B. Sensitivity analysis for probability assessments in bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):901–909, June 1995. ISSN 0018-9472. doi: 10.1109/21.384252.

Lei, J. Differentially private m-estimators. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 361–369. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4376-differentially-private-m-estimators.pdf.

Li, C., Hay, M., Rastogi, V., Miklau, G., and McGregor, A. Optimizing linear counting queries under differential privacy. In *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '10, pp. 123–134, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0033-9. doi: 10.1145/1807085.1807104. URL http://doi.acm.org/10.1145/1807085.1807104.

Li, C., Hay, M., Miklau, G., and Wang, Y. A data- and workload-aware algorithm for range queries under differential privacy. *Proceedings of the VLDB Endowment*, 7(5):341–352, Jan 2014. ISSN 2150-8097. doi: 10.14778/2732269.2732271. URL http://dx.doi.org/10.14778/2732269.2732271.

Li, N., Qardaji, W. H., and Su, D. On sampling, anonymization, and differential privacy or, *k*-anonymization meets differential privacy. In *7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012*, pp. 32–33, 2012. doi: 10.1145/2414456.2414474. URL https://doi.org/10.1145/2414456.2414474.

McKenna, R., Miklau, G., Hay, M., and Machanavajjhala, A. Optimizing error of high-dimensional statistical queries under differential privacy. *Proc. VLDB Endow.*, 11(10):1206–1219, June 2018. ISSN 2150-8097. doi: 10.14778/3231751.3231769. URL https://doi.org/10.14778/3231751.3231769.

Park, M., Foulds, J., Chaudhuri, K., and Welling, M. Dp-em: Differentially private expectation maximization, 2016a.

Park, M., Foulds, J. R., Chaudhuri, K., and Welling, M. Variational bayes in private settings (vips). *CoRR*, abs/1611.00340, 2016b.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0-934613-73-7.

Pearl, J. *Graphical Models for Probabilistic and Causal Reasoning*, pp. 367–389. Springer Netherlands, Dordrecht, 1998. ISBN 978-94-017-1735-9. doi: 10.1007/978-94-017-1735-9_12. URL https://doi.org/10.1007/978-94-017-1735-9_12.

Qardaji, W., Yang, W., and Li, N. Priview: Practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pp. 1435–1446, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2376-5. doi: 10.1145/2588555.2588575. URL http://doi.acm.org/10.1145/2588555.2588575.

Schein, A., Wu, Z. S., Zhou, M., and Wallach, H. M. Locally private bayesian inference for count models. *CoRR*, abs/1803.08471, 2018.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, May 2017. doi: 10.1109/SP.2017.41.

Thaler, J., Ullman, J., and Vadhan, S. Faster algorithms for privately releasing marginals. *Lecture Notes in Computer Science*, pp. 810–821, 2012. ISSN 1611-3349. doi: 10.1007/978-3-642-31594-7_68. URL http://dx.doi.org/10.1007/978-3-642-31594-7_68.

Wang, D., Ye, M., and Xu, J. Differentially private empirical risk minimization revisited: Faster and more general. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 2722–2731. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6865-differentially-private-empirical-risk-minimization-revisited-faster-and-more-general.pdf.

Wang, Y., Wang, Y.-X., and Singh, A. Differentially private subspace clustering. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1000–1008, Cambridge, MA, USA, 2015a. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969351.

Wang, Y.-X., Fienberg, S. E., and Smola, A. J. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Bach, F. R. and Blei, D. M. (eds.), *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2493–2502. JMLR.org, 2015b. URL http://dblp.uni-trier.de/db/conf/icml/icml2015.html#WangFS15.

Williams, O. and Mcsherry, F. Probabilistic inference and differential privacy. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23*, pp. 2451–2459. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/3897-probabilistic-inference-and-differential-privacy.pdf.

Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. *Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*, 2017. doi: 10.1145/3035918.3064047. URL http://dx.doi.org/10.1145/3035918.3064047.

Xiao, X., Wang, G., and Gehrke, J. Differential privacy via wavelet transforms. *IEEE Trans. on Knowl. and Data Eng.*, 23(8):1200–1214, August 2011. ISSN 1041-4347. doi: 10.1109/TKDE.2010.247. URL http://dx.doi.org/10.1109/TKDE.2010.247.

Xiao, Y., Gardner, J., and Xiong, L. Dpcube: Releasing differentially private data cubes for health information. In *2012 IEEE 28th International Conference on Data Engineering*, pp. 1305–1308, April 2012. doi: 10.1109/ICDE.2012.135.

Xu, J., Zhang, Z., Xiao, X., Yang, Y., and Yu, G. Differentially private histogram publication. In *2012 IEEE 28th International Conference on Data Engineering*, pp. 32–43, April 2012. doi: 10.1109/ICDE.2012.48.

Yaroslavtsev, G., Procopiuc, C. M., Cormode, G., and Srivastava, D. Accurate and efficient private release of datacubes and contingency tables. In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, pp. 745–756, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4673-4909-3. doi: 10.1109/ICDE.2013.6544871. URL http://dx.doi.org/10.1109/ICDE.2013.6544871.

Yuan, G., Zhang, Z., Winslett, M., Xiao, X., Yang, Y., and Hao, Z. Low-rank mechanism:optimizing batch queries under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1352–1363, Jul 2012. ISSN 2150-8097. doi: 10.14778/2350229.2350252. URL http://dx.doi.org/10.14778/2350229.2350252.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization, 2016a.

Zhang, G. and Li, S. Research on differentially private bayesian classification algorithm for data streams. In *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, pp. 14–20, March 2019. doi: 10.1109/ICBDA.2019.8713253.

Zhang, H., Kamath, G., Kulkarni, J., and Wu, Z. S. Privately learning markov random fields, 2020.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42 (4):25:1–25:41, October 2017. ISSN 0362-5915. doi: 10.1145/3134428. URL http://doi.acm.org/10.1145/3134428.

Zhang, X., Chen, R., Xu, J., Meng, X., and Xie, Y. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 587–595. doi: 10.1137/1.9781611973440.68. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.68.

Zhang, Z., Rubinstein, B. I. P., and Dimitrakakis, C. On the differential privacy of bayesian inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2365–2371. AAAI Press, 2016b. URL http://dl.acm.org/citation.cfm?id=3016100.3016229.