# Data-Dependent Differentially Private Parameter Learning for Directed Graphical Models – Supplementary Material

## 8. Appendix

### 8.1. Background Cntd.

#### 8.1.1. Directed Graphical Models Cntd.

**Variable Elimination Algorithm (VE)**: The complete VE algorithm is given by Algorithm 3. The basic idea of the variable elimination algorithm is that we "*eliminate*" one variable at a time following a predefined order $\prec$ over the nodes of the graph. Let $\Phi$ denote a set of probability factors which is initialized as the set of all CPDs of the DGM and $Z$ denote the variable to be eliminated. For the elimination step, firstly all the probability factors involving the variable to be eliminated, $Z$ are removed from $\Phi$ and multiplied together to generate a new product factor. Next, the variable $Z$ is summed out from this combined factor generating a new factor that is entered into $\Phi$. Thus the VE algorithm essentially involves repeated computation of a sum-product task of the form

$$\boldsymbol{\phi} = \sum_{Z} \prod_{\phi \in \Phi} \phi \tag{16}$$

The complexity of the VE algorithm is defined by the size of the largest factor. Here we state two lemmas regarding the intermediate factors $\boldsymbol{\phi}$ which will be used in Section 8.3.

**Lemma 8.1.** *Every intermediate factor ($\boldsymbol{\phi}$ in (16)) generated as a result of executing the VE algorithm on a DGM $\mathcal{N}$ correspond to a valid conditional probability of some DGM (not necessarily the same DGM, $\mathcal{N}$). (Koller & Friedman, 2009)*

**Lemma 8.2.** *The size of the largest intermediary factor generated as a result of running of the VE algorithm on a DGM is at least equal to the treewidth of the graph (Koller & Friedman, 2009).*

**Corollary.** *The complexity of the VE algorithm with the optimal order of elimination depends on the treewidth of the graph.*

### 8.2. Data-Dependent Differentially Private Parameter Learning for DGMs Cntd.

#### 8.2.1. Consistency between noisy marginal tables

The objective of this step is to input the set of noisy marginal tables $\tilde{M}_i$ and compute perturbed versions of these tables that are mutually consistent (Defn. 2.3. The following procedure has been reproduced from (Hay et al., 2010a; Qardaji et al., 2014) with a few adjustments.

**Algorithm 2** Sum Product Variable Elimination Algorithm

**Notations** : $\Phi$ - Set of factors
$\mathbf{X}$ - Set of variables to be eliminated
$\prec$ - Ordering on $\mathbf{X}$
$X$ - Variable to be eliminated
$Attr(\phi)$ - Attribute set of factor $\phi$

**Procedure** Sum-Product-VE($\Phi, \mathbf{X}, \prec$)

Let $X_1, \cdots, X_k$ be an ordering of $\mathbf{X}$ such that $X_i \prec X_j$ iff $i < j$
  **for** $i = 1, \cdots, k$
    $\Phi \leftarrow$ Sum-Product-Eliminate-Var($\Phi, Z_i$)
    $\phi^* \leftarrow \prod_{\phi \in \Phi} \phi$
  **return** $\phi^*$

**Procedure** Sum-Product-Eliminate-Var($\Phi, X$)

$\Phi' \leftarrow \{\phi \in \Phi : Z \in Attr(\phi)\}$
$\Phi'' \leftarrow \Phi - \Phi'$
$\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
$\boldsymbol{\phi} \leftarrow \sum_{Z} \psi$
  **return** $\Phi'' \cup \{\boldsymbol{\phi}\}$

**Mutual Consistency on a Set of Attributes:**

Assume a set of tables $\{\tilde{M}_i, \cdots, \tilde{M}_j\}$ and let $A = Attr(\tilde{M}_i) \cap \cdots \cap Attr(\tilde{M}_j)$. Mutual consistency, i.e., $\tilde{M}_i[A] \equiv \cdots \equiv \tilde{M}_j[A]$ is achieved as follows:

(1) First compute the best approximation for the marginal table $\tilde{M}_A$ for the attribute set $A$ as follows

$$\tilde{M}_A[A'] = \frac{1}{\sum_{t=1}^{j} \epsilon_t} \sum_{t=i}^{j} \epsilon_t \cdot \tilde{M}_t[A'], A' \in A \tag{17}$$

(2) Update all $\tilde{M}_t$s to be consistent with $\tilde{M}_A$. Any counting query $c$ is now answered as

$$\tilde{M}_t(c) = \tilde{M}_t(c) + \frac{|dom(A)|}{|dom(Attr(\tilde{M})|}(\tilde{M}_A(a) - \tilde{M}_t(a)) \tag{18}$$

where $a$ is the query $c$ restricted to attributes in $A$ and $\tilde{M}_t(c)$ is the response of $c$ on $\tilde{M}_t$ .

**Overall Consistency:**

(1) Take all sets of attributes that are the result of the intersection of some subset of $\bigcup_{i=k+1}^{d}\{X_i \cup X_{pa_i}\}$; these sets form a partial order under the subset relation.

(2) Obtain a topological sort of these sets, starting from the empty set.

(3) For each set $A$, one finds all tables that include $A$, and ensures that these tables are consistent on $A$.

#### 8.2.2. Privacy Analysis

**Theorem 3.1.** *The proposed algorithm (Algorithm 1) for learning the parameters of a fully observed directed graphical model is $\epsilon^B$-differentially private.*

*Proof.* The sensitivity of counting queries is 1. Hence, the computation of the noisy tables $\tilde{T}_i$ (Proc. 1, Line 2-3) is a straightforward application of Laplace mechanism (Sec. 2). This together with Lemma 2.3 makes the computation of $\tilde{T}_i$, $\epsilon^I$-DP. Now the subsequent computation of the optimal privacy budget allocation $\mathcal{E}^*$ is a post-processing operation on $\tilde{T}_i$ and hence by Thm. 2.2 is still $\epsilon^I$-DP. The final parameter computation is clearly ($\epsilon^B$-$\epsilon^I$)-DP. Thus by the theorem of sequential composition (Thm. 2.1), Algorithm 1 is $\epsilon^B$-DP. □

### 8.3. Error Bound Analysis Cntd.

In this section, we present the proofs of Thm. 4.1 and Thm. 4.2.

**Preliminaries and Notations:**
For the proofs, we use the following notations. Let $X$ be the attribute that is being eliminated and let $\mathcal{A} = \bigcup_{\phi_i} Attr(\phi_i)\backslash X$ where $Attr(\phi)$ denotes the set of attributes in $\phi$. For some $a \in dom(\mathcal{A})$, from the variable elimination algorithm (Sec. 8.1.1) for a sum-product term (Eq. (16)) we have

$$\boldsymbol{\phi}_{\mathcal{A}}[a] = \sum_{x}\prod_{i=1}^{t}\phi_i[x,a] \tag{19}$$

Let us assume that factor $\phi[a,x]$ denotes that $Value(Attr(\phi)) \in \{a\}$ and $X = x$. Recall that after computing a sum-product task (given by Eq. (20)), for the variable elimination algorithm (Appx. Algorithm 2), we will be left with a factor term over the attribute set $\mathcal{A}$. For example, if the elimination order for the variable elimination algorithm on our example DGM (Figure 1) is given by $\prec = \{A, B, C, D, E, F\}$ and the attributes are binary valued, then the first sum-product task will be of the following form $\mathcal{A} = \{B, C\}, dom(\mathcal{A}) = \{(0,0),(0,1),(1,0),(1,1)\}$ and the RHS $\phi_i s$ in this case happen to be the true parameters of the DGM,

$$\begin{aligned}
\boldsymbol{\phi}_{B,C}[0,0] &= \Theta[A=0]\cdot\Theta[C=0|A=0,B=0] + \\
&\quad \Theta[A=1]\cdot\Theta[C=0|A=1,B=0] \\
\boldsymbol{\phi}_{B,C}[0,1] &= \Theta[A=0]\cdot\Theta[C=1|A=0,B=0] + \\
&\quad \Theta[A=1]\cdot\Theta[C=1|A=1,B=0] \\
\boldsymbol{\phi}_{B,C}[1,0] &= \Theta[A=0]\cdot\Theta[C=1|A=0,B=0] + \\
&\quad \Theta[A=1]\cdot\Theta[C=1|A=1,B=0] \\
\boldsymbol{\phi}_{B,C}[1,1] &= \Theta[A=0]\cdot\Theta[C=1|A=0,B=1] + \\
&\quad \Theta[A=1]\cdot\Theta[C=1|A=1,B=1] \\
\boldsymbol{\phi}_{B,C} &= [\boldsymbol{\phi}_{B,C}[0,0], \boldsymbol{\phi}_{B,C}[0,1], \boldsymbol{\phi}_{B,C}[1,0], \boldsymbol{\phi}_{B,C}[1,1]]
\end{aligned}$$

#### 8.3.1. Lower Bound

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

$$\delta_{\phi_{\mathcal{A}}} \geq \sqrt{\eta-1}\cdot\delta^{min}_{\phi_i[a,x]}(\phi_i^{min}[a,x])^{\eta-2} \tag{20}$$

*where $X$ is the attribute being eliminated, $\delta_\phi$ denotes the error in factor $\phi$, $Attr(\phi)$ is the set of attributes in $\phi$, $\mathcal{A} = \bigcup_{\phi_i}\{Attr(\phi_i)\}/X$, $x \in dom(X)$, $a \in dom(\mathcal{A})$, $\phi[a,x]$ 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$(X_i)\}+1$.*

*Proof.* **Proof Structure:**

The proof is structured as follows. First, we compute the error for a single term $\boldsymbol{\phi}_{\mathcal{A}}[a], a \in dom(\mathcal{A})$ (Eq. (21),(22),(23)). Next we compute the total error $\delta_{\boldsymbol{\phi}_{\mathcal{A}}}$ by summing over $\forall a \in dom(\mathcal{A})$. This is done by dividing the summands into two types of terms (a) $\Upsilon_{\phi_1[a,x]}$ (b) $\delta_{\prod_{i=1}^{t}\phi_i[a,x]}\prod_{i=1}^{t}\phi_i[a,x]$ (Eq. (25),(26)). We prove that the summation of first type of terms ($\Upsilon_{\phi_1[x]}$) can be lower bounded by 0 non-trivially. Then we compute a lower bound on the terms of the form $\delta_{\prod_{i=2}^{t}\phi_i[a,x]}\prod_{i=2}^{t}\phi_i[a,x]$ (Eq. (31)) which gives our final answer (Eq. (32)).

**Step 1: Computing error in a single term $\boldsymbol{\phi}_{\mathcal{A}}[a]$, $\delta_{\boldsymbol{\phi}_{\mathcal{A}}[a]}$**
The error in $\boldsymbol{\phi}_{\mathcal{A}}[a]$, due to noise injection is given by ,

$$\begin{aligned}
\delta_{\boldsymbol{\phi}_{\mathcal{A}}[a]} &= \left|\sum_x\prod_{i=1}^{t}\phi_i[x,a] - \sum_x\prod_{i=1}^{t}\tilde{\phi}_i[a,x]\right| \\
&= \left|\sum_x\left(\phi_1[x,a]\prod_{i=2}^{t}\phi_i[x,a] - \tilde{\phi}_1[x,a]\prod_{i=2}^{t}\tilde{\phi}_i[x,a]\right)\right| \\
&= \left|\textstyle\sum_x\left(\phi_1[x,a]\prod_{i=2}^{t}\phi_i[x,a] - \tilde{\phi}_1[x,a]\prod_{i=2}^{t}(\phi_i[x,a]\pm\delta_{\phi_i[x,a]})\right)\right|
\end{aligned} \tag{21}$$

Using the rule of standard error propagation, we have

$$\delta_{\prod_{i=2}^{t}\phi_i[x,a]} = \prod_{i=2}^{t}\tilde{\phi}_i[x,a]\sqrt{\sum_{i=2}^{t}\Big(\frac{\delta_{\phi_i[x,a]}}{\phi_i[x,a]}\Big)^2} \tag{22}$$

Thus from the above equation (Eq. (22)) we can rewrite Eq. (21) as follows,

$$\begin{aligned}
&= \Big|\textstyle\sum_x\Big(\phi_1[x,a]\prod_{i=2}^{t}\phi_i[x,a] - \phi_1[\tilde{x},a]\prod_{i=2}^{t}\phi_i[x,a](1\pm\delta_{\prod_{i=2}^{t}\phi_i[x,a]})\Big)\Big| \\
&= \Big|\textstyle\sum_x\Big((\phi_1[a,x]-\phi_1[\tilde{a},x])\prod_{i=2}^{t}\phi_i[a,x] \pm \delta_{\prod_{i=1}^{t}\phi_i[a,x]}\prod_{i=1}^{t}\phi_i[a,x]\Big)\Big|
\end{aligned} \tag{23}$$

**Step 2: Compute total error $\delta_{\boldsymbol{\phi}_{\mathcal{A}}}$**
Now, total error in $\phi_{\mathcal{A}}$ is

$$\delta_{\phi_{\mathcal{A}}} = \sum_a \delta_{\boldsymbol{\phi}_{\mathcal{A}}[a]} \tag{24}$$

Collecting all the product terms from the above equation (24) with $\phi_1[a,x]-\tilde{\phi}_1[a,x]$ as a multiplicand, we get

$$\Upsilon_{\phi_1[a,x]} = (\phi_1[a,x]-\tilde{\phi}_1[a,x])\sum_a\prod_{i=2}^{t}\phi_i[a,x] \tag{25}$$

Thus $\delta_{\boldsymbol{\phi}_{\mathcal{A}}}$ can be rewritten as

$$\delta_{\boldsymbol{\phi}_{\mathcal{A}}} = \sum_{a,x}\Upsilon_{\phi_1[a,x]} \pm \sum_{a,x}\prod_{i=1}^{t}\phi_i[a,x]\delta_{\prod_{i=2}^{t}\phi_i[a,x]} \tag{26}$$

First we show that for a specific DGM we have $\sum_{a,x}\Upsilon_{\phi_1[a,x]} = 0$ as follows. Let us assume that the DGM has $Attr(\phi_1) = X$. Thus $\phi_1[a,x]$ reduces to just $\phi_1[x]$.

$$\begin{aligned}
\Upsilon_{\phi_1[x]} &= (\phi_1[x]-\tilde{\phi}_1[x])\sum_a\prod_{i=2}^{t}\phi_i[x] \\
&= (\phi_1[x]-\tilde{\phi}_1[x])\Big(\sum_{a_k}\cdots\sum_{a_1}\prod_{i=2}^{t}\phi_i[a_1,\cdots,a_k,x]\Big) \\
&\quad [\mathcal{A} = \langle\mathcal{A}_1,\cdots\mathcal{A}_k\rangle, a_j \in dom(\mathcal{A}_j), j\in[k]] \\
&= (\phi_1[x]-\tilde{\phi}_1[x])\Big(\textstyle\sum_{a_k}\cdots\sum_{a_2}\prod_{i=3}^{t}\phi_i[a_2,\cdots,a_k,x]\sum_{a_1}\phi_2[a_1,\cdots,a_k,x]\Big) \\
&\quad [\text{Assuming that } \phi_2 \text{ is the only factor with attribute } \mathcal{A}_1]
\end{aligned}$$

Now each factor $\phi_i$ is either a true parameter (CPD) of the DGM $\mathcal{N}$ or a CPD over some other DGM (lemma 8.1). Thus, let us assume that $\phi_2$ represents a conditional of the form $P[\mathcal{A}_1|\mathbf{A},X], \mathbf{A} = \mathcal{A}/\mathcal{A}_1$. Thus we have $\sum_{a_1}\phi_2[a_1,\cdots,a_k,x] = \sum_{a_1}P[\mathcal{A}_1 = a_1|\mathcal{A}_2 = a_2,\cdots,\mathcal{A}_k = a_k, X = x] = 1$. Now repeating the above process over all $i\in\{3,\cdots,t\}\phi_i$s , we get

$$\Upsilon_{\phi_1[x]} = \phi_1[x]-\tilde{\phi}_1[x] \tag{27}$$

For the ease of understanding, we illustrate the above result on our example DGM (Figure 1). Let us assume that the order of elimination is given by $\prec = \langle A,B,C,D,E,F\rangle$. For simplicity, again we assume binary attributes. Let $\phi_C$ be the factor that is obtained after eliminating $A$ and $B$. Thus the sum-product task for eliminating $C$ is given by

$$\begin{aligned}
\boldsymbol{\phi}_{D,E}[0,0] &= \phi_C[C=0]\cdot\Theta[D=0|C=0]\Theta[E=0|C=0] \\
&\quad +\phi_C[C=1]\cdot\Theta[D=0|C=1]\Theta[E=0|C=1] \\
\boldsymbol{\phi}_{D,E}[0,1] &= \phi_C[C=0]\cdot\Theta[D=0|C=0]\Theta[E=1|C=0] \\
&\quad +\phi_C[C=1]\cdot\Theta[D=0|C=1]\Theta[E=1|C=1] \\
\boldsymbol{\phi}_{D,E}[1,0] &= \phi_C[C=0]\cdot\Theta[D=1|C=0]\Theta[E=0|C=0] \\
&\quad +\phi_C[C=1]\cdot\Theta[D=1|C=1]\Theta[E=0|C=1] \\
\boldsymbol{\phi}_{D,E}[1,1] &= \phi_C[C=0]\cdot\Theta[D=1|C=0]\Theta[E=1|C=0] \\
&\quad +\phi_C[C=1]\cdot\Theta[D=1|C=1]\Theta[E=1|C=1]
\end{aligned}$$

Hence considering noisy $\tilde{\boldsymbol{\phi}}_{D,E}$ we have,

$$\begin{aligned}
&\Upsilon_{\phi_C[0]} = (\phi_C[C=0]-\tilde{\phi}_C[C=0])\cdot(\Theta[D=0|C=0]\Theta[E=0|C=0] \\
&+\Theta[D=0|C=0]\Theta[E=1|C=0]+\Theta[D=1|C=0]\Theta[E=0|C=0] \\
&\quad +\Theta[D=1|C=0]\Theta[E=1|C=0]) \\
&= (\phi_C[C=0]-\tilde{\phi}_C[C=0])\cdot\Big(\Theta[D=0|C=0]\big(\Theta[E=0|C=0]+\Theta[E=1|C=0]\big) \\
&\quad +\Big(\Theta[D=1|C=0]\big(\Theta[E=0|C=0]+\Theta[E=1|C=0]\big)\Big) \\
&= (\phi_C[C=0]-\tilde{\phi}_C[C=0])\cdot\big(\Theta[D=0|C=0]+\Theta[D=1|C=0]\big) \\
&\quad \Big[\because \Theta[E=0|C=0]+\Theta[E=1|C=0]=1\Big] \\
&\quad = \phi_C[C=0]-\tilde{\phi}_C[C=0] \\
&\quad \Big[\because \Theta[D=0|C=0]+\Theta[D=1|C=0]=1\Big]
\end{aligned} \tag{28}$$

Similarly

$$\Upsilon_{\phi_C[1]} = \phi_C[C=1]-\tilde{\phi}_C[C=1] \tag{29}$$

Now using Eq. (27) and summing over $\forall x\in dom(X)$

$$\begin{aligned}
\sum_x\Upsilon_{\phi_1[x]} &= \sum_x(\phi_1[x]-\tilde{\phi}_1[x]) \\
&= 0\Big[\because\sum_x\phi_1[x] = \sum_x\tilde{\phi}_1[x] = 1\Big]
\end{aligned} \tag{30}$$

Referring back to our example above, since $\phi_C[1]+\phi_C[0] = \tilde{\phi}_C[C=0]+\tilde{\phi}_C[C=1]$, quite trivially

$$\begin{aligned}
&\phi_C[C=0]+\phi_C[C=1] = \tilde{\phi}_C[C=0]+\tilde{\phi}_C[C=1] \\
&\Rightarrow (\phi_C[C=0]-\tilde{\phi}_C[C=0])+(\phi_C[C=1]-\tilde{\phi}_C[C=1]) = 0
\end{aligned}$$

Thus, from Eq. (26)

$$\delta_{\phi_{\mathcal{A}}} = \sum_x \Upsilon_{\phi_1[x]} \pm \sum_{a,x} \delta_{\prod_{i=2}^t \phi_i[a,x]} \prod_{i=1}^t \phi_i[a,x]$$

$$= \sum_{a,x} \delta_{\prod_{i=2}^t \phi_i[a,x]} \prod_{i=1}^t \phi_i[a,x]$$

$$\Big[\text{From Eq. (30) and dropping } \pm \text{ as we are dealing with errors}\Big]$$

$$\geq \delta^{min}_{\prod_{i=2}^t \phi_i[a,x]} \sum_{a,x} \prod_{i=1}^t \phi_i[a,x]$$

$$\Big[\delta^{min}_{\prod_{i=2}^t \phi_i[a,x]} = \min_{a,x} \Big\{\delta_{\prod_{i=2}^t \phi_i[a,x]}\Big\}\Big]$$

$$\geq \delta^{min}_{\prod_{i=2}^t \phi_i[a,x]}$$

$$\Big[\because \text{By Lemma 8.1 } \phi_{\mathcal{A}} \text{ is a CPD, thus } \sum_{a,x} \prod_{i=1}^t \phi_i[a,x] \geq 1\Big]$$

$$= min_{a,x}\Bigg\{\prod_{i=2}^t \phi_i[x,a] \sqrt{\sum_{i=2}^t \Big(\frac{\delta_{\phi_i[a,x]}}{\phi_i[a,x]}\Big)^2}\Bigg\}$$

$$\geq min_{a,x}\Bigg\{\prod_{i=2}^t \phi_i[x,a] \sqrt{(t-1)\Big(\frac{\delta^{min}_{\phi_i[a,x]}}{\phi_i[x,a]}\Big)^2}\Bigg\}$$

$$[\delta^{min}_{\phi_i[x,a]} = min_{i,a,x}\Big\{\delta_{\phi_i[a,x]}\Big\}]$$

$$\geq min_{a,x}\Bigg\{\sqrt{(t-1)}\frac{\delta^{min}_{\phi_i[a,x]}}{\phi_i^{max}} \prod_{i=2}^t \phi_i[a,x]\Bigg\}$$

$$\Big[\phi_i^{max} = \max_i\{\phi_i[a,x]\}\Big]$$

$$\geq \delta^{min}_{\phi_i[a,x]}\sqrt{t-1}(\phi_i^{min}[a,x])^{t-2} \tag{31}$$

$$\Big[\text{Assuming } \phi_i^{min}[a,x] = min_{i,a,x}\{\phi_i[a,x]\}\Big]$$

Now, recall from the variable elimination algorithm that during each elimination step, if $Z$ is the variable being eliminated then we the product term contains all the factors that include $Z$. For a DGM with graph $\mathcal{G}$, the maximum number of such factors is clearly $\eta = \max_{X_i}\{\text{out-degree}(X_i) + \text{in-degree}(X_i)\} + 1$ of $\mathcal{G}$, i.e., $t \leq \eta$. Additionally we have $\phi^{min}[a,x] \leq \frac{1}{d_{min}} \leq \frac{1}{2}$ where $d_{min}$ is the minimum size of $dom(Attr(\phi))$ and clearly $d_{min} \geq 2$. Since $2^t \geq \sqrt{t}, t \geq 2$, under the constraint that $t$ is an integer and $\phi^{min}[a,x] \leq \frac{1}{2}$, we have

$$\delta_{\phi_{\mathcal{A}}} \geq \sqrt{\eta - 1}\delta^{min}_{\phi_i[a,x]}(\phi^{min}[a,x])^{\eta-2} \tag{32}$$

□

### 8.3.2. Upper Bound

**Theorem 4.2.** *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,*

$$\delta_{\phi_{\mathcal{A}}} \leq 2 \cdot \eta \cdot d^{\kappa} \delta^{max}_{\phi_i[a,x]} \tag{33}$$

*where $X$ is the attribute being eliminated, $\delta_\phi$ denotes the error in factor $\phi$, $\kappa$ is the treewidth of $\mathcal{G}$, $d$ is the maximum attribute domain size, $Attr(\phi)$ is the set of attributes in $\phi$, $\mathcal{A} = \bigcup_i^t \{Attr(\phi_i)\}/X$, $a \in dom(\mathcal{A})$, $x \in dom(X)$, $\phi[a,x]$ 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$(X_i)\} + 1$.*

*Proof.* **Proof Structure:**

The proof is structured as follows. First we compute an upper bound for a product of $t > 0$ noisy factors $\tilde{\phi}_i[a,x], i \in [t]$ (Lemma 8.3). Next we use this lemma, to bound the error, $\delta_{\phi_{\mathcal{A}}}[a]$, for the factor, $\phi_{\mathcal{A}}[a], a \in dom(\mathcal{A})$ (Eq. (34)). Finally we use this result to bound the total error, $\delta_{\phi_{\mathcal{A}}}$, by summing over $\forall a \in dom(\mathcal{A})$ (Eq. (35)).

**Step 1: Computing the upper bound of the error of a single term $\tilde{\phi}_{\mathcal{A}}[a]$, $\delta_{\phi_{\mathcal{A}}[a]}$**

**Lemma 8.3.** *For $a \in dom(\mathcal{A}), x \in dom(X)$*

$$\prod_{i=1}^t \tilde{\phi}_i[a,x] \leq \prod_{i=1}^t \phi_i[a,x] + \sum_i \delta_{\phi_i[a,x]}$$

*Proof.* First we consider the base case when $t = 2$.

**Base Case**:

$$\tilde{\phi_1}[a,x]\tilde{\phi_2}[a,x] = (\phi_1[a,x] \pm \delta_{\phi_1[a,x]})(\phi_2[a,x] \pm \delta_{\phi_2[a,x]})$$

$$\leq (\phi_1[a,x] + \delta_{\phi_1[a,x]})(\phi_2[a,x] + \delta_{\phi_2[a,x]})$$

$$= (\phi_1[a,x] \cdot \phi_2[a,x] + \delta_{\phi_1[a,x]}(\phi_2[a,x] + \delta_{\phi_2[a,x]}) + \delta_{\phi_2[a,x]} \cdot \phi_1[a,x])$$

$$\leq (\phi_1[a,x] \cdot \phi_2[a,x] + \delta_{\phi_1[a,x]} + \delta_{\phi_2[a,x]}\phi_1[a,x])$$

$$\Big[\because (\phi_2[a,x] + \delta_{\phi_2[a,x]}) \leq 1 \text{ as } \tilde{\phi}_i[a,x] \text{ is still a valid probability distribution}\Big]$$

$$\leq \phi_1[a,x] \cdot \phi_2[a,x] + \delta_{\phi_1[a,x]} + \delta_{\phi_2[a,x]}$$

$$\Big[\because \phi_1[a,x] < 1\Big]$$

**Inductive Case:**

Let us assume that the lemma holds for $t = k$. Thus we

have

$$\prod_{i=1}^{k+1}\tilde{\phi}_i[a,x] = \prod_{i=1}^{k}\tilde{\phi}_i[a,x]\cdot\tilde{\phi}_{k+1}[a,x]$$
$$\leq (\prod_{i=1}^{k}\phi_i[a,x] + \sum_i \delta_{\phi_i[a,x]})\cdot(\phi_{k+1}[a,x]+\delta_{\phi_{k+1}}[a,x])$$
$$\leq \prod_{i=1}^{k+1}\phi_i[a,x] + \sum_i \delta_{\phi_i[a,x]}\cdot(\phi_{k+1}[a,x]+\delta_{\phi_{k+1}}[a,x])$$
$$+\delta_{\phi_{k+1}}[a,x]\prod_{i=1}^{k}\phi_i[a,x]$$
$$\leq \prod_{i=1}^{k+1}\phi_i[a,x] + \sum_{i=1}^{k+1}\delta_{\phi_i[a,x]} + \delta_{\phi_{k+1}}[a,x]\prod_{i=1}^{k}\phi_i[a,x]$$
$[\because (\phi_{k+1}[a,x]+\delta_{\phi_{k+1}[a,x]}) \leq 1$ as $\tilde{\phi}_{k+1}[a,x]$ is still a valid probability distribution$]$
$$\leq \prod_{i=1}^{k+1}\phi_i[a,x] + \sum_{i=1}^{k+1}\delta_{\phi_i[a,x]} [\because \forall i, \phi_i[a,x]\leq 1]$$

Hence, we have

$$\prod_{i=1}^{t}\tilde{\phi}_i[a,x] \leq \prod_{i=1}^{t}\phi_i[a,x] + \sum_i \delta_{\phi_i[a,x]}$$

□

Next, we compute the error for the factor, $\boldsymbol{\phi}_{\mathcal{A}}[a], a \in dom(\mathcal{A})$ as follows

$$\delta_{\boldsymbol{\phi}_A[a]} = \Big|\sum_x \prod_{i=1}^{t}\phi_i[a,x] - \sum_x\prod_{i=1}^{t}\tilde{\phi}_i[a,x]\Big|$$
$$= \Big|\sum_x \prod_{i=1}^{t}\phi_i[a,x] - \phi_1[a,x]\prod_{i=2}^{t}\tilde{\phi_i}[a,x]\Big|$$
$$= \Big|\sum_x \prod_{i=1}^{t}\phi_i[a,x] - \tilde{\phi}_1[a,x]\prod_{i=2}^{t}(\phi_i[a,x]\pm\delta_{\phi_i[a,x]})\Big|$$
$$\leq \Big|\sum_x \Big(\prod_{i=1}^{t}\phi_i[a,x] - \tilde{\phi}_1[a,x](\prod_{i=2}^{t}\phi_i[a,x] + \sum_{i=2}^{t}\delta_{\phi_i[a,x]})\Big)\Big|$$
$$\Big[\text{Using Lemma 8.3}\Big]$$
$$\leq \Big|\sum_x \Big((\phi_1[a,x]-\tilde{\phi}_1[a,x])\prod_{i=2}^{t}\phi_i[a,x] + \tilde{\phi}_1[a,x]\sum_{i=2}^{t}\delta_{\phi_i[a,x]}\Big)\Big|$$
$$\leq \Big|\sum_x \Big((\phi_1[a,x]-\tilde{\phi}_1[a,x])\prod_{i=2}^{t}\phi_i[a,x] + \eta\tilde{\phi}_1[a,x]\delta^*_{\phi_i[a,x]}\Big)\Big|$$
$$\Big[\because t\leq\eta \text{ and assuming } \delta^*_{\phi_i[a,x]} = \max_{i,x}\{\delta_{\phi_i[a,x]}\}\Big]$$
$$= \Big|\sum_x(\phi_1[a,x]-\tilde{\phi}_1[a,x])\prod_{i=2}^{t}\phi_i[a,x] + \eta\delta^*_{\phi_i[a,x]}\sum_x\tilde{\phi}_1[a,x]\Big|$$
$$\leq \sum_x\Big|\phi_1[a,x]-\tilde{\phi}_1[a,x]\Big| + \eta\delta^*_{\phi_i[a,x]}\sum_x\tilde{\phi}_1[a,x] \quad (34)$$
$$\Big[\because \phi_i[a,x]\leq 1\Big]$$

**Step 2: Computing the upper bound of the total error $\delta_{\boldsymbol{\phi}_{\mathcal{A}}}$**

Now summing over $\forall a \in dom(\mathcal{A})$,

$$\delta_{\boldsymbol{\phi}_{\mathcal{A}}} = \sum_a \delta_{\boldsymbol{\phi}_{\mathcal{A}}[a]}$$
$$\leq \sum_a\Big(\sum_x|\phi_1[a,x]-\tilde{\phi}_1[a,x]| + \eta\delta^*_{\phi_i[a,x]}\sum_x\tilde{\phi}_1[a,x]\Big)$$
[From Eq. (34)]
$$= \delta_{\phi_1} + \eta\delta^{max}_{\phi_i[a,x]}\sum_a\sum_x\tilde{\phi}_1[a,x]\Big[\delta^{max}_{\phi_i[a,x]} = \max_a\{\delta^*_{\phi_i[a,x]}\}\Big]$$

Now by Lemma 8.2, maximum size of $\mathcal{A}\cup X$ is given by the treewidth of the DGM, $\kappa$. Thus from the fact that $\phi_1$ is a CPD (Lemma 8.1), we observe that $\sum_a\sum_x\tilde{\phi}_1[a,x]$ is maximized when $\phi_1[a,x]$ is of the form $P[A'|\mathbf{A}], A'\in\mathcal{A}\cup X, |A'|=1, \mathbf{A}=(\mathcal{A}\cup X)/A'$ and is upper bounded by $d^\kappa$ where $d$ is the maximum domain size of an attribute.

$$\delta_{\boldsymbol{\phi}_{\mathcal{A}}} \leq \delta_{\phi_1} + \eta d^\kappa\delta^{max}_{\phi_i[a,x]}$$
[By lemma 8.2 and that $\phi_1$ is CPD from lemma 8.1]
where $\kappa$ is the treewidth of $\mathcal{G}$ an
$d$ is the maximum domain size of an attribute
$$\leq 2\cdot\eta\cdot d^\kappa\delta^{max}_{\phi_i[a,x]}\Big[\because \delta_{\phi_1}\leq\eta\cdot d^\kappa\delta^{max}_{\phi_i[a,x]}\Big] \quad (35)$$

□

## 9. Related Work

In this section, we review related literature. There has been a steadily growing amount work in differentially private machine learning models for the last couple of years. We list some of the most recent work in this line (not exhaustive list). (Abadi et al., 2016; Wu et al., 2017; Agarwal et al., 2018) address the problem of differentially private SGD. The authors of (Park et al., 2016a) present an algorithm for differentially private expectation maximization. In (Lei, 2011) the problem of differentially private M-estimators is addressed. Algorithms for performing expected risk minimization under differential privacy has been proposed in (Wang et al., 2017; Chaudhuri et al., 2011). In (Wang et al.,

2015a) two differentially private subspace clustering algorithms are proposed.

There has been a fair amount of work in differentially private Bayesian inferencing and related notions (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; Zhang & Li, 2019; Schein et al., 2018; Park et al., 2016b; Jälkö et al., 2016; Barthe et al., 2016; Bernstein et al., 2017; Dziugaite & Roy, 2018; Zhang et al., 2017; 2020). In (Heikkilä et al., 2017) the authors present a solution for DP Bayesian learning in a distributed setting, where each party only holds a subset of the data a single sample or a few samples of the data. In (Dziugaite & Roy, 2018) the authors show that a data-dependent prior learnt under $\epsilon$-DP yields a valid PAC-Bayes bound. The authors in (Williams & Mcsherry, 2010) show that probabilistic inference over differentially private measurements to derive posterior distributions over the data sets and model parameters can potentially improve accuracy. An algorithm to learn an unknown probability distribution over a discrete population from random samples under $\epsilon$-DP is presented in (Diakonikolas et al., 2015). In (Bernstein & Sheldon, 2018) the authors present a method for private Bayesian inference in exponential families that learns from sufficient statistics. The authors of (Wang et al., 2015b) and (Dimitrakakis et al., 2014) show that posterior sampling gives differential privacy "for free" under certain assumptions. In (Foulds et al., 2016) the authors show that Laplace mechanism based alternative for "One Posterior Sample" is as asymptotically efficient as non-private posterior inference, under general assumptions. A Rényi differentially private posterior sampling algorithm is presented in (Geumlek et al., 2017). (Zhang & Li, 2019) proposes a differential private Naive Bayes classification algorithm for data streams. (Zhang et al., 2016b) presents algorithms for private Bayesian inference on probabilistic graphical models. In (Park et al., 2016b), the authors introduce a general privacy-preserving framework for Variational Bayes. An expressive framework for writing and verifying differentially private Bayesian machine learning algorithms is presented in (Barthe et al., 2016). The problem of learning discrete, undirected graphical models in a differentially private way is studied in (Bernstein et al., 2017). (Schein et al., 2018) presents a general method for privacy-preserving Bayesian inference in Poisson factorization. In (Zhang et al., 2020), the authors consider the problem of learning Markov Random Fields under differential privacy. 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 (Algorithm 1 and Algorithm 2 as 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 (*D-Ind* has an additional consistency step) to an algorithm used in (Zhang et al., 2017) which uses DGMs to generate high-dimensional data.

A number of data-dependent differentially private algorithms have been proposed in the past few years. (Acs et al., 2012; Xu et al., 2012; Zhang et al.; Xiao et al., 2012) outline data-dependent mechanisms for publishing histograms. In (Cormode et al., 2012b) the authors construct an estimate of the dataset by building differentially private kd-trees. MWEM (Hardt et al., 2012) derives estimate of the dataset iteratively via multiplicative weight updates. In (Li et al., 2014) differential privacy is achieved by adding data and workload dependent noise. (Kotsogiannis et al., 2017a) presents a data-dependent differentially private algorithm selection technique. (Gupta et al., 2012; Dwork et al., 2010) present two general data-dependent differentially private mechanisms. Certain data-independent mechanisms attempt to find a better set of measurements in support of a given workload. One of the most prominent technique is the matrix mechanism framework (Yuan et al., 2012; Li et al., 2010) which formalizes the measurement selection problem as a rank-constrained SDP. Another popular approach is to employ a hierarchical strategy (Hay et al., 2010b; Cormode et al., 2012a; Xiao et al., 2011). (Yaroslavtsev et al., 2013; Barak et al., 2007; Ding et al., 2011b; Gupta et al., 2011; Thaler et al., 2012; Hay et al., 2010a) propose techniques for marginal table release.

### 9.1. Evaluation Cntd.

#### 9.1.1. Data sets

As mentioned in Section 5, we evaluate our algorithm on the following four DGMs.

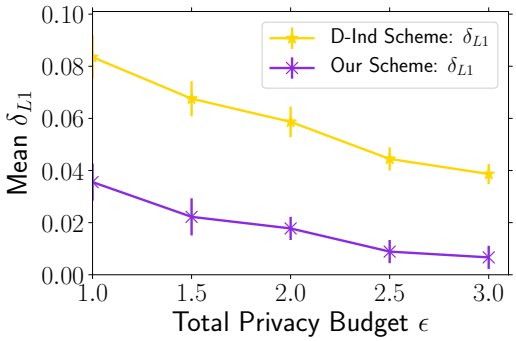
(1) **Asia**: Number of nodes – 8; Number of arcs – 8; Number of parameters – 18

(2) **Sachs**: Number of nodes – 11; Number of arcs – 17; Number of parameters – 178

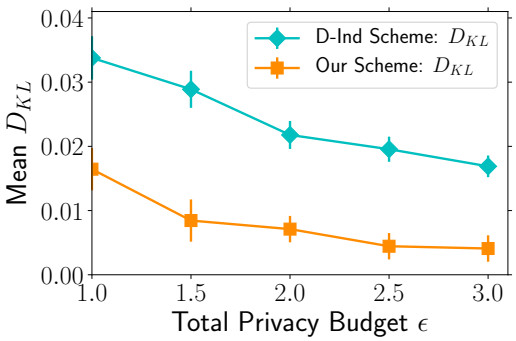
(3) **Child**: Number of nodes – 20; Number of arcs – 25; Number of parameters – 230

(4) **Alarm**: Number of nodes – 37; Number of arcs – 46; Number of parameters – 509

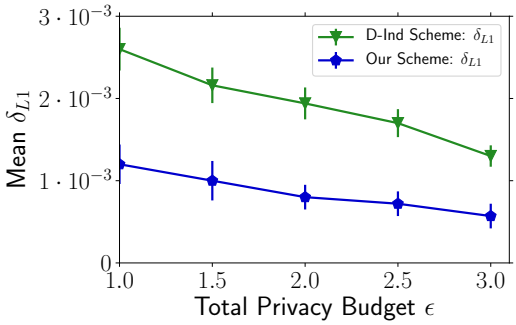
The error analysis for the data sets Asia and Alarm are presented in Fig. 4.



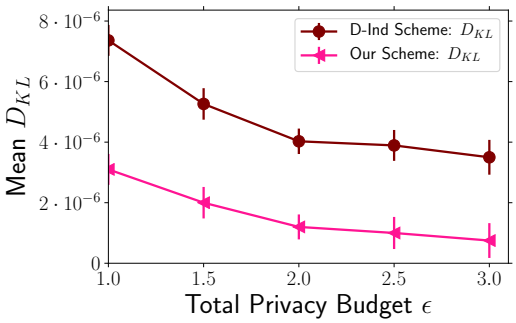

**(a)** Asia: Parameters $\delta_{L1}$



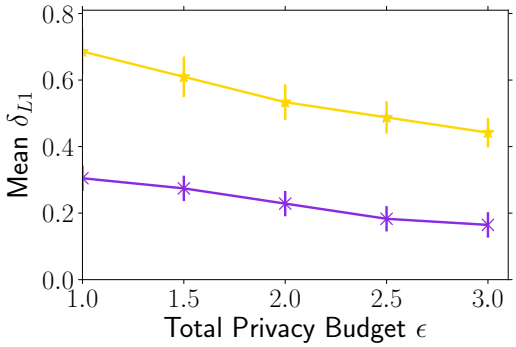

**(b)** Asia: Parameters $D_{KL}$



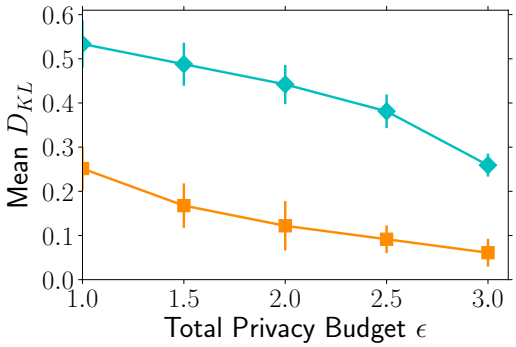

**(c)** Asia: Inference $\delta_{L1}$



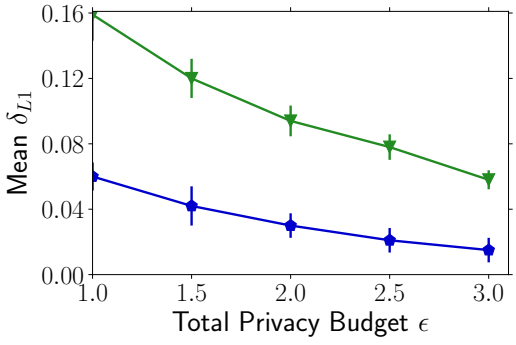

**(d)** Asia: Inference $D_{KL}$



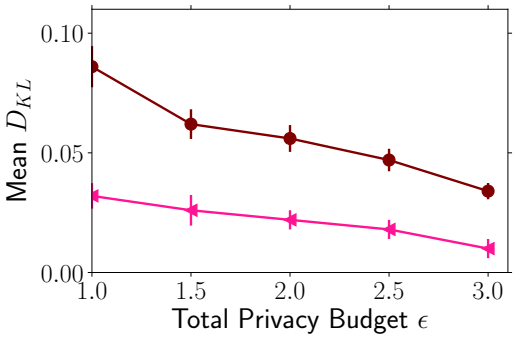

**(e)** Alarm: Parameters $\delta_{L1}$




**(f)** Alarm: Parameters $D_{KL}$




**(g)** Alarm: Inference $\delta_{L1}$




**(h)** Alarm: Inference $D_{KL}$

**Figure 4:** Parameter and Inference (Marginal and Conditional) Error Analysis Cntd.