Appendix: Quantifying the Privacy Risks of Learning High-Dimensional Graphical Models

Contents

Symbol	**Description**
m	Number of attributes
n	Pool (training set) size
$\langle G, \hat{\theta} \rangle$	Released Model
$\langle G, \theta \rangle$	Population Model
X_i	Random variable for attribute i
x_i	A particular value for X_i
$V(X_i)$	Set of possible values of attribute i
$Pa^G_{X_i}$	Set of random variables that are parents of node X_i in G
p^v_i	$\Pr(x_i = 1 \vert Pa^G_{X_i} = v; \theta)$
$\hat{p}^v_i$	$\Pr(x_i = 1 \vert Pa^G_{X_i} = v; \hat{\theta})$
n^v_i	Number of samples used to compute $\hat{p}^v_i$
$C(G)$	Complexity of G (number of independent parameters)
η	Maximum number of parents per node in G
$L(x)$	Log-likelihood ratio statistic for data sample x
L_i	Contribution of X_i to the Log-likelihood ratio $L(x)$
F	CDF of $L(x)$ over the general population (under $\mathrm{H}_{\mathrm{OUT}}$)
α	Error (False Positive Rate) of LR tracing attack
β	Power (True Positive Rate) of LR tracing attack
z_s	Quantile at level $1 - s$ of the Standard Normal distribution

Table 1: **Notations**

A Derivation of mean and variance of the likelihood ratio

We compute the mean and variance of $L(x)$ under the two hypotheses. We sketch the proof for the mean $\mathrm{E}(L)$ under the population hypothesis, followed by the variance $\mathrm{Var}(L)$. Similar calculations apply for the pool hypothesis.

Let the target x have the feature vector $(x_1, x_2, \dots, x_m)$, and let us assume, for now, that all attributes are binary: $x_i \in \{0,1\}, i = 1, \dots, m$. In Appendix E we generalize to attributes that can take more than two values. We can take advantage of the Bayesian network decomposition to write the log-likelihood ratio for x as follows:

$$L(x) = \log\left[\frac{\Pr(x; \langle G, \theta\rangle)}{\Pr(x; \langle G, \hat{\theta}\rangle)}\right] = \sum_{i=1}^{m} L_i \tag{1}$$

where L_i is the contribution of X_i to the likelihood ratio, as defined as:

$$\begin{aligned} L_i &= \log\left(\frac{\Pr[X_i|Pa^G_{X_i}; \theta]}{\Pr[X_i|Pa^G_{X_i}; \hat{\theta}]}\right) \\ &= \sum_{v \in V(Pa^G_{X_i})} 1_{\{Pa^G_{X_i}=v\}} \underbrace{\left(x_i \log\frac{p_i^v}{\hat{p}_i^v} + (1-x_i)\log\frac{1-p_i^v}{1-\hat{p}_i^v}\right)}_{L_i^v} \\ &= \sum_{v \in V(Pa^G_{X_i})} 1_{\{Pa^G_{X_i}=v\}} L_i^v \end{aligned} \tag{2}$$

where $p_i^v = \Pr\{X_i = 1|Pa^G_{X_i} = v; \theta\}$, and similarly $\hat{p}_i^v = \Pr\{X_i = 1|Pa^G_{X_i} = v; \hat{\theta}\}$. The notation $1_{\{Pa^G_{X_i}=v\}}$ is an indicator variable for a particular assignment of values to the parent nodes of X_i, i.e. $1_{\{Pa^G_{X_i}=v\}} = 1$ if $Pa^G_{X_i} = v$ and 0 otherwise. The sum ranges over $|V(Pa^G_{X_i})|$ terms L_i^v, one for each element of $V(Pa^G_{X_i})$.

The parameters θ and $\hat{\theta}$ are estimated from data (reference population and pool, respectively). By the central limit theorem, the distribution of such an estimate converges to a Gaussian around the mean value of the estimate as the number of data samples increases. In our derivations of the mean and variance, we use this approximation in (10) and (11). By the Berry-Esseen theorem [1, 3], the rate of convergence to the Gaussian is $O(\frac{1}{\sqrt{n}})$ if the third moment of the random variable being sampled is finite. In our case this condition is true, because each random variable can only take a finite number of possible finite values.

We compute the mean and variance of $L(x)$ as follows:

$$\mathrm{E}_{pop}(L) = \frac{C}{2n} + O(Cn^{-2}) \tag{3a}$$

$$\mathrm{E}_{pool}(L) = -\frac{C}{2n} + O(Cn^{-2}) \tag{3b}$$

$$\mathrm{Var}_{pop}(L) = \frac{C}{n} + O(C^2n^{-2}) \tag{3c}$$

$$\mathrm{Var}_{pool}(L) = \frac{C}{n} + O(C^2n^{-2}). \tag{3d}$$

Proof sketch - Mean under H_{OUT}. The mean $\mathrm{E}_{pop}(L)$ can be computed as follows:

$$\begin{aligned} \mathrm{E}_{pop}(L) &= \sum_{i=1}^{m} \mathrm{E}_{pop}(L_i) \\ &= \sum_{i=1}^{m} \sum_{v} \mathrm{E}_{pop}(1_{\{Pa^G_{X_i}=v\}} L_i^v) \end{aligned} \tag{4}$$

Using approximation (12) in Appendix Section C, we compute $\mathrm{E}_{pop}(1_{\{Pa^G_{X_i}=v\}} L_i^v) \approx \frac{1}{2n} + O(n^{-2})$. Since the total number of L_i^v parameters is $C = \sum_{i=1}^m |V(Pa^G_{X_i})|$, we conclude that

$$\mathrm{E}_{pop}(L) = \frac{C}{2n} + O(Cn^{-2}). \tag{5}$$

□

Proof sketch - Variance under H_{OUT}. By definition,

$$\mathrm{Var}_{pop}(L) = \mathrm{E}_{pop}[L^2] - (\mathrm{E}_{pop}[L])^2. \tag{6}$$

The latter term $(\mathrm{E}_{pop}[L])^2$ is the square of the mean, which we compute in (5). The former term $\mathrm{E}_{pop}[L^2]$ decomposes as follows:

$$\mathrm{E}_{pop}[L^2] = \sum_{i=1}^m \mathrm{E}_{pop}[L_i^2] + 2 \sum_{1\le i<j\le m} \mathrm{E}_{pop}[L_i L_j] \tag{7}$$

We compute $\mathrm{E}_{pop}[L_i^2]$ by expanding $\mathrm{E}_{pop}[(\sum_v 1_{\{Pa^G_{X_i}=v\}} L_i^v)^2]$. Then, approximation (22) in Appendix D gives us that each square term $\mathrm{E}_{pop}[(1_{\{Pa^G_{X_i}=v\}} L_i^v)^2]$ is approximately equal to $\frac{1}{n}$. As for the product terms in the expansion, each term multiplies two different indicator variables $1_{\{Pa^G_{X_i}=v\}}$ and $1_{\{Pa^G_{X_i}=v'\}}$ with $v \neq v'$. Because at most one of the two is equal to 1, all product terms will be zero. Hence $\mathrm{E}_{pop}[L_i^2] = |V(Pa^G_{X_i})| \times \frac{1}{n}$.

The number of joint terms $\mathrm{E}_{pop}[L_i L_j]$ is $O(C^2)$. From the approximation in Appendix Section D.1 for $\mathrm{E}_{pop}[L_i L_j]$, each of these terms is equal to $\frac{1}{4n^2}$ with error term $O(n^{-2})$. Hence, the value of $\mathrm{E}_{pop}[L^2]$ is

$$E_{pop}[L^2] = \frac{C}{n} + \frac{C^2}{4n^2} + O(C^2 n^{-2}). \tag{8}$$

We conclude that the variance is

$$\begin{aligned}\mathrm{Var}_{pop}(L) =& \mathrm{E}_{pop}[L^2] - (\mathrm{E}_{pop}[L])^2 \\ =& \frac{C}{n} + \frac{C^2}{4n^2} + O(C^2 n^{-2}) - \left(\frac{C}{2n} + O(Cn^{-2})\right)^2 \\ =& \frac{C}{n} + O(C^2 n^{-2}) \end{aligned} \tag{9}$$

□

Although we haven't provided the calculation here, it is possible to calculate the exact value of the $O(C^2n^{-2})$ term from the released model. As a simple example, in appendix F, we calculate the exact value of this $O(C^2n^{-2})$ term, when the released model is a Naive Bayes model.

A.1 Distribution of the log-likelihood ratio

To compute the distribution of $L(x)$, the log-likelihood ratio of the parameter vector estimate with the actual value of the parameter vector in a graphical model, we need to understand what parameters contribute to the likelihood ratio given a data sample. As shown in equation (2), the parameter that contributes to the likelihood ratio for an attribute is determined by the value taken by its parent node. The contribution of attribute X_i to the likelihood function, denoted by L_i, is computed as:

$$L_i = \sum_{v\in V(Pa^G_{X_i})} 1_{\{Pa^G_{X_i}=v\}} L_i^v$$

Hence the distribution of L_i is a mixture of the distributions of L_i^v, where the mixing probabilities are determined by the distribution of the parent nodes (hence the dependence of $L(x)$'s distribution on the probability distribution that generated the data). The distribution of L_i^v, the log-likelihood ratio for the estimate of a single parameter value, is asymptotically a chi-squared distribution with degree of freedom 1 (from Wilks' theorem). Hence, the exact distribution of log-likelihood ratio is a sum of mixture of chi-squared distributions, where the mixing distribution is dependent on the distribution that generated the data. In case of high-dimension models, this log-likelihood ratio distribution is very close to normal distribution (as it is a sum of large number of independent random variables (which are sum of mixtures themselves)). Hence, using the first two moments, that do not depend on the exact distribution of the sensitive data, is sufficient to produce a generic data-independent upper-bound on the privacy risk of learning the graphical model.

B Number of Samples for Estimating Conditional Probabilities

We use $\hat{p}_i^v$ to denote the estimated conditional probability that $X_i = 1$, given that the values of the activator variables are $Pa_{X_i}^G = v$. The number of samples n_i^v used to compute $\hat{p}_i^v$ are approximately Gaussian around np_v (n is the pool size, and p_v is the probability of $Pa_{X_i}^G = v$ in the general population):

$$n_i^v \approx np_v + \sqrt{np_v(1-p_v)}Z_1, \tag{10}$$

where Z_1 is a standard Gaussian random variable. In parallel, the value of $\hat{p}_i^v$ is also approximately Gaussian around the true value p_i^v:

$$\hat{p}_i^v \approx p_i^v + \sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}}Z_2, \tag{11}$$

where Z_2 is a standard Gaussian random variable.

Using these two approximations, we now prove the results required for derivation of LR statistic mean and variance.

C Approximation for mean derivation

As explained in section A, to compute the mean of the likelihood ratio we need the average contribution of each L_i^v i.e. value of $E_{pop}[1_{\{Pa_{X_i}^G=v\}}L_i^v]$. Here we prove that $E_{pop}[1_{\{Pa_{X_i}^G=v\}}L_i^v]$, when the expectation is over population is approximately equal to $\frac{1}{2n}$. When expectation is over pool, the derivation steps are similar and the value is $-\frac{1}{2n}$.

Lemma 1. *We will prove the following result:*

$$E_{pop}[1_{\{Pa_{X_i}^G=v\}}L_i^v] \approx \frac{1}{2n}\left(1 + \frac{1-p_v}{np_v}\right) \tag{12}$$

Proof. We first observe that

$$\begin{aligned} E_{pop}[1_{\{Pa_{X_i}^G=v\}}L_i^v] =& E_{\hat{p}_i^v}\left[E_x\left[1_{\{Pa_{X_i}^G=v\}}L_i^v \mid \hat{p}_i^v\right]\right] \\ =& p_v E_{\hat{p}_i^v}\left[p_i^v \log\frac{p_i^v}{\hat{p}_i^v} + (1-p_i^v)\log\frac{1-p_i^v}{1-\hat{p}_i^v}\right], \end{aligned}$$

and now all we need to show is that

$$E_{Z_1,Z_2}\left[p_i^v \log\frac{p_i^v}{\hat{p}_i^v} + (1-p_i^v)\log\frac{1-p_i^v}{1-\hat{p}_i^v}\right] \approx \frac{1}{2np_v}\left(1 + \frac{1-p_v}{np_v}\right) \tag{13}$$

We approximate $\hat{p}_i^v$ with (11) and we use the Taylor expansion of $\log(1+x) \approx x - \frac{1}{2}x^2$:

$$\begin{aligned} p_i^v \log \frac{p_i^v}{\hat{p}_i^v} \approx & - p_i^v \log \frac{p_i^v + \sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}} Z_2}{p_i^v} \\ = & - p_i^v \log \left(1 + \sqrt{\frac{1-p_i^v}{n_i^v p_i^v}} Z_2\right) \\ \approx & - p_i^v \left(\sqrt{\frac{1-p_i^v}{n_i^v p_i^v}} Z_2 - \frac{1-p_i^v}{2n_i^v p_i^v} Z_2^2 \right) \\ = & - \sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}} Z_2 + \frac{1-p_i^v}{2n_i^v} Z_2^2 \end{aligned} \tag{14}$$

Similarly,

$$(1-p_i^v) \log \frac{1-p_i^v}{1-\hat{p}_i^v} \approx -\sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}} Z_2 + \frac{p_i^v}{2n_i^v} Z_2^2 \tag{15}$$

Adding (14) and (15), we have

$$p_i^v \log \frac{p_i^v}{\hat{p}_i^v} + (1-p_i^v) \log \frac{1-p_i^v}{1-\hat{p}_i^v} \approx -2\sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}} Z_2 + \frac{1}{2n_i^v} Z_2^2 \tag{16}$$

Taking the expectation $E_{Z_2}[.]$, and recalling that $E[Z_2] = 0$ and $E[Z_2^2] = 1$, we have

$$\begin{aligned} E_{Z_1,Z_2}\left[p_i^v \log \frac{p_i^v}{\hat{p}_i^v} + (1-p_i^v) \log \frac{1-p_i^v}{1-\hat{p}_i^v}\right] = & E_{Z_1}[E_{Z_2}[\dots|Z_1]] \\ \approx & E_{Z_1}\left[\frac{1}{2n_i^v}\right] \end{aligned} \tag{17}$$

We now approximate n_i^v with (10) and we use the Taylor expansion of $\frac{1}{1+x} \approx 1 - x + x^2$:

$$\begin{aligned} \frac{1}{2n_i^v} \approx & \frac{1}{2(np_v + \sqrt{np_v(1-p_v)} Z_1)} \\ = & \frac{1}{2np_v} \frac{1}{1 + \sqrt{\frac{1-p_v}{np_v}} Z_1} \\ \approx & \frac{1}{2np_v} \left(1 - \sqrt{\frac{1-p_v}{np_v}} Z_1 + \frac{1-p_v}{np_v} Z_1^2\right) \end{aligned} \tag{18}$$

Taking the expectation $E_{Z_1}[.]$, and recalling that $E[Z_1] = 0$ and $E[Z_1^2] = 1$, we have our final result:

$$E_{Z_1,Z_2}\left[p_i^v \log \frac{p_i^v}{\hat{p}_i^v} + (1-p_i^v) \log \frac{1-p_i^v}{1-\hat{p}_i^v}\right] \approx \frac{1}{2np_v}\left(1 + \frac{1-p_v}{np_v}\right)$$

□

D Approximation for variance derivation

For calculating the variance of likelihood ratio, we need the expected values of L_i^2 and $L_i L_j$. Here we first prove the below approximation and use it to calculate $E(L_i^2)$ and $E(L_i L_j)$. As explained in section A, using these values of $E(L_i^2)$ and $E(L_i L_j)$ in equation 7 we get the variance of LR statistic.

Lemma 2. *We will prove the following approximation:*

$$\mathrm{E}_{\hat{p}_i^v}\left[p_i^v \left(\log \frac{p_i^v}{\hat{p}_i^v}\right)^2 + (1-p_i^v)\left(\log \frac{1-p_i^v}{1-\hat{p}_i^v}\right)^2\right] \approx \frac{1}{np_v}\left(1 + \frac{1-p_v}{np_v}\right) \tag{19}$$

Proof. Using approximation (14)

$$\begin{aligned}
\mathrm{E}_{\hat{p}_i^v}\left[p_i^v\left(\log\frac{p_i^v}{\hat{p}_i^v}\right)^2\right] \approx & \mathrm{E}_{Z_1,Z_2}\left[\frac{1}{p_i^v}\left(-\sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}}Z_2+\frac{1-p_i^v}{2n_i^v}Z_2^2\right)^2\right] \\
=&\frac{1}{p_i^v}\,\mathrm{E}_{Z_1,Z_2}\left[\frac{p_i^v(1-p_i^v)}{n_i^v}Z_2^2+\left(\frac{1-p_i^v}{2n_i^v}\right)^2Z_2^4-2\sqrt{\frac{p_i^v(1-p_i^v)}{n_i^v}}\frac{1-p_i^v}{2n_i^v}Z_2^3\right] \\
=&\frac{1}{p_i^v}\,\mathrm{E}_{Z_1}\left[\frac{p_i^v(1-p_i^v)}{n_i^v}+3\left(\frac{1-p_i^v}{2n_i^v}\right)^2\right] \\
\approx&(1-p_i^v)\,\mathrm{E}_{Z_1}\left[\frac{1}{n_i^v}\right] \\
\approx&(1-p_i^v)\,\mathrm{E}_{Z_1}\left[\frac{1}{np_v}\left(1-\sqrt{\frac{1-p_v}{np_v}}Z_1+\frac{1-p_v}{np_v}Z_1^2\right)\right] \\
=&\frac{1-p_i^v}{np_v}\left(1+\frac{1-p_v}{np_v}\right)
\end{aligned} \tag{20}$$

Similar to (20), we have:

$$\mathrm{E}_{\hat{p}_i^v}\left[(1-p_i^v)\left(\log\frac{1-p_i^v}{1-\hat{p}_i^v}\right)^2\right]\approx\frac{p_i^v}{np_v}\left(1+\frac{1-p_v}{np_v}\right) \tag{21}$$

The desired result follows. □

D.1 Approximation of $\mathrm{E}_{pop}[L_i^2]$

We approximate $\mathrm{E}_{pop}[L_i^2]$ as:

$$\begin{aligned}
\mathrm{E}_{pop}[L_i^2] =& \mathrm{E}_{pop}\left[\left(\sum_v 1_{\{Pa_{X_i}^G=v\}}L_i^v\right)^2\right] \\
=&\mathrm{E}_{\hat{p}_i^v}\left[\mathrm{E}\left[\left(\sum_v 1_{\{Pa_{X_i}^G=v\}}L_i^v\right)^2\middle|\hat{p}_i^v\right]\right] \\
=&\sum_v p_v\,\mathrm{E}_{\hat{p}_i^v}[(L_i^v)^2] \\
=&\sum_v p_v\,\mathrm{E}_{\hat{p}_i^v}\left[p_i^v\left(\log\frac{p_i^v}{\hat{p}_i^v}\right)^2+(1-p_i^v)\left(\log\frac{1-p_i^v}{1-\hat{p}_i^v}\right)^2\right] \\
\approx&\sum_v\frac{1}{n}\left(1+\frac{1-p_v}{np_v}\right)\text{(from approximation (19))} \\
=&\frac{1}{n}|V(Pa_{X_i}^G)|+\frac{1}{n^2}\sum_v\frac{1-p_v}{p_v}
\end{aligned} \tag{22}$$

Combining the definition of complexity with equation (22), we have:

$$\sum_{i=1}^m \mathrm{E}_{pop}[L_i^2]\approx\frac{C}{n}+\frac{1}{n^2}\sum_{i=1}^m\sum_v\frac{1-p_v}{p_v} \tag{23}$$

D.2 Approximation of $\mathrm{E}_{pop}[L_iL_j]$

There are three possible cases while finding the value of $\mathrm{E}[L_iL_j]$. The random variables X_i and X_j might not have any common parents, might have some common parents or one is the parent of other. We start with the case in which X_i and X_j have no common parents. Let $p(v_i, v_j)$ represent the joint probability of $Pa^G_{X_i} = v_i$ and $Pa^G_{X_j} = v_j$.

$$\begin{aligned}
\mathrm{E}_{pop}[L_i L_j] =& \mathrm{E}_{pop}\left[\left(\sum_{v} 1_{\{Pa^G_{X_i}=v_i\}} L_i^v\right)\left(\sum_{v_j} 1_{\{Pa^G_{X_j}=v_j\}} L_j^v\right)\right] \\
&= \mathrm{E}_{pop}\left[\sum_{v_i, v_j} 1_{\{Pa^G_{X_i}=v_i\}} 1_{\{Pa^G_{X_j}=v_j\}} L_i^v L_j^v\right] \\
&= \sum_{v_i, v_j} \mathrm{E}_{pop}\left[1_{\{Pa^G_{X_i}=v_i\}} 1_{\{Pa^G_{X_j}=v_j\}} L_i^v L_j^v\right] \\
&= \sum_{v_i, v_j} p(v_i, v_j)\, \mathrm{E}_{pop}\left[L_i^v L_j^v\right] \\
&\approx \sum_{v_i, v_j} p(v_i, v_j) \times \frac{1}{2np_{v_i}} \times \frac{1}{2np_{v_j}} (\text{from } (13)) \\
&= \sum_{v_i, v_j} \frac{1}{4n^2} \times \frac{p(v_i, v_j)}{p_{v_i} p_{v_j}} \qquad (24)
\end{aligned}$$

In the case where X_i and X_j have common parents S_{ij}, let S_i represent the parents exclusive to X_i and S_j represent parents exclusive to X_j. Let $p(v_i, v_j, v_{ij})$ represent the joint probability of $Pa^G_{X_i} = v_i$ and $Pa^G_{X_j} = v_j$ and common parent of X_i and X_j, $Pa^G_{X_{i,j}} = v_{ij}$.

$$\begin{aligned}
\mathrm{E}_{pop}[L_i L_j] =& \mathrm{E}_{pop}\left[\left(\sum_{v_i, v_{ij}} 1_{\{Pa^G_{X_i}=v_i\}} 1_{\{Pa^G_{X_{ij}}=v_{ij}\}} L_i^v\right)\left(\sum_{v_j, v_{ij}} 1_{\{Pa^G_{X_j}=v_j\}} 1_{\{Pa^G_{X_{ij}}=v_{ij}\}} L_j^v\right)\right] \\
&= \mathrm{E}_{pop}\left[\sum_{v_i, v_j, v_{ij}} \left(1_{\{Pa^G_{X_i}=v_i\}} 1_{\{Pa^G_{X_j}=v_j\}} 1_{\{Pa^G_{X_{ij}}=v_{ij}\}} L_i^v L_j^v\right)\right] \\
&= \sum_{v_i, v_j, v_{ij}} p(v_i, v_j, v_{ij})\, \mathrm{E}_{pop}\left[L_i^v L_j^v\right] \\
&\approx \sum_{v_i, v_j, v_{ij}} p(v_i, v_j, v_{ij}) \times \frac{1}{2np(v_i, v_{ij})} \times \frac{1}{2np(v_j, v_{ij})} \qquad (\text{from } (13)) \\
&= \sum_{v_i, v_j, v_{ij}} \frac{1}{4n^2} \times \frac{p(v_i, v_j, v_{ij})}{p(v_i, v_{ij}) p(v_j, v_{ij})} \qquad (25)
\end{aligned}$$

In the case where X_j is a parent of X_i,

$$\begin{aligned}
\mathrm{E}_{pop}[L_iL_j] =& \mathrm{E}_{pop}\left[\left(\sum_{v_i} 1_{\{Pa^G_{X_i}=v_i\}} x_j L_i^v\right)\left(\sum_{v_j} 1_{\{Pa^G_{X_j}=v_j\}} L_j^v\right)\right] \\
=& \mathrm{E}_{pop}\left[\sum_{v_i,v_j}\left(1_{\{Pa^G_{X_i}=v_i\}} 1_{\{Pa^G_{X_j}=v_j\}} x_j L_i^v \left(x_j \log \frac{p_j^v}{\hat{p}_j^v}\right)\right)\right] \\
=& \sum_{v_i,v_j} p(v_i,v_j,x_j)\, \mathrm{E}_{pop}\left[L_i^v \log \frac{p_j^v}{\hat{p}_j^v}\right] \\
\approx& \sum_{v_i,v_j} p(v_i,v_j,x_j) \times \frac{1}{2np(v_i,x_j)} \times \frac{1-p_j^v}{2np_j^v} (\text{from (13)}) \\
=& \sum_{v_i,v_j} \frac{1-p_j^v}{4n^2} \times \frac{p(v_i,v_j,x_j)}{p(v_i,x_j)p_j^v}
\end{aligned} \tag{26}$$

E Generic Categorical Variable

In this section, we generalize our results to any categorical variables (not just binary). The extension from binary to categorical is straightforward. We will have a similar expression for the likelihood ratio statistic:

$$\begin{aligned}
L(x) =& \log\left(\frac{\Pr(x;\langle G,\theta\rangle)}{\Pr(x;\langle G,\hat{\theta}\rangle)}\right) \\
=& \sum_{i=1}^m L_i,
\end{aligned}$$

where L_i is the contribution of X_i to $L(x)$:

$$L_i = \sum_v 1_{\{Pa^G_{X_i}=v\}} L_i^v$$

Instead of writing L_i^v as

$$L_i^v = x_i \log \frac{p_i^v}{\hat{p}_i^v} + (1-x_i)\log\frac{1-p_i^v}{1-\hat{p}_i^v}$$

we write

$$L_i^v = \sum_{o\in V(X_i)} 1_{\{x_i=o\}} \log \frac{p_{io}^v}{\hat{p}_{io}^v},$$

$$p_{io}^v = \Pr(x_i = o | Pa^G_{X_i} = v)$$

Now,

$$\begin{aligned}
\mathrm{E}_{pop}[L_i^v] =& \sum_{o\in V(X_i)} E\left[p_{io}^v \log \frac{p_{io}^v}{\hat{p}_{io}^v}\right] \\
=& \sum_{o\in V(X_i)} \frac{1-p_{io}^v}{2n_i^v} \quad (\text{from (14)}) \\
=& \frac{|V(X_i)|-1}{2n_i^v}
\end{aligned} \tag{27}$$

$$\begin{aligned}
\mathrm{E}_{pop}[L_i] =& \sum_v E[1_{\{Pa^G_{X_i}=v\}} L^v_i | Pa^G_{X_i} = v] \\
=& \sum_v E_{\hat{p}^v_i} \left[E_x \left[1_{\{Pa^G_{X_i}=v\}} L^v_i \mid \hat{p}^v_i \right] \right] \\
=& \sum_v p_v \frac{|V(X_i)| - 1}{2n^v_i} \quad \text{(from (27))} \\
=& \sum_v \frac{|V(X_i)| - 1}{2n} + O(n^{-2}) \quad \text{(from (18))} \\
=& |V(Pa_{X_i})| \times \frac{|V(X_i)| - 1}{2n} + O(n^{-2})
\end{aligned}$$

Now we can calculate $\mathrm{E}_{pop}[L(x)]$ as:

$$\begin{aligned}
\mathrm{E}_{pop}[L(x)] =& \sum_{i=1}^{m} E_{pop}[L_i] \\
\approx& \sum_{i=1}^{m} |V(Pa_{X_i})| \times \frac{|V(X_i)| - 1}{2n} + O(n^{-2}) \\
=& \frac{C}{2n} + O(Cn^{-2})
\end{aligned}$$

Hence,

$$\mathrm{E}_{pop}[L(x)] = \frac{C}{2n} + O(Cn^{-2}) \tag{28}$$

Similarly for deriving variance we have,

$$\begin{aligned}
\mathrm{E}_{pop}[(L^v_i)^2] =& \sum_{o \in V(X_i)} \mathrm{E} \left[p^v_{io} \left(\log \frac{p^v_{io}}{\hat{p}^v_{io}} \right)^2 \right] \\
=& \sum_{o \in V(X_i)} \frac{1 - p^v_{io}}{n^v_i} \quad \text{(from (20))} \\
=& \frac{|V(X_i)| - 1}{n^v_i}
\end{aligned} \tag{29}$$

Using equation (29), we can calculate $\mathrm{E}_{pop}[L^2_i]$ as:

$$\begin{aligned}
\mathrm{E}_{pop}[L^2_i] =& \sum_v \mathrm{E}[1_{\{Pa^G_{X_i}=v\}} (L^v_i)^2 | Pa^G_{X_i} = v] \\
=& \sum_v \mathrm{E}_{\hat{p}^v_i} \left[E_x \left[1_{\{Pa^G_{X_i}=v\}} (L^v_i)^2 \mid \hat{p}^v_i \right] \right] \\
=& \sum_v p_v \frac{|V(X_i)| - 1}{n^v_i} \quad \text{(from (29))} \\
=& \sum_v \frac{|V(X_i)| - 1}{n} + O(n^{-2}) \quad \text{(from (18))} \\
=& |V(Pa_{X_i})| \times \frac{|V(X_i)| - 1}{n} + O(n^{-2})
\end{aligned}$$

Hence,

$$\begin{aligned}\sum_{i=1}^{m} \mathrm{E}_{pop}[L_i^2] =& \sum_{i=1}^{m} |V(Pa_{X_i})| \times \frac{|V(X_i)|-1}{n} + O(n^{-2}) \\ =& \frac{C}{n} + O(Cn^{-2})\end{aligned}$$

$$\begin{aligned}\mathrm{Var}_{pop}(L) =& \mathrm{E}_{pop}[L^2] - (\mathrm{E}_{pop}[L])^2 \\ \mathrm{E}_{pop}[L^2] =& \sum_{i=1}^{m} \mathrm{E}[L_i^2] + 2 \sum_{1 \le i < j \le m} \mathrm{E}[L_i L_j]\end{aligned}$$

Similar to the derivations of $\sum \mathrm{E}_{pop}[L_i]$ and $\sum \mathrm{E}_{pop}[L_i^2]$, we will have

$$\sum_{i,j} \mathrm{E}_{pop}[L_i L_j] = \frac{C^2}{4n^2} + O(C^2 n^{-2})$$

Hence, for categorical variables:

$$Var_{pop}[L(x)] = \frac{C}{n} + O(C^2 n^{-2}) \tag{30}$$

F Naive Bayes

In section A, while deriving the variance, we haven't calculated the exact value of the $O(C^2n^{-2})$ term. From the released model, it is possible to calculate the exact value of this term. Here we derive the exact value of the $O(C^2n^{-2})$ term, when the released model is a Naive Bayes model. Let the number of attributes in the model be equal to m. Hence, the complexity of the model is $C = 2m - 1$. Let X_1 be the class variable and $p_i^1 = Pr(X_i = 1|X_1 = 1)$. Then, using equation (13) we have:

$$\begin{aligned}\mathrm{E}_{pop}(L) = \mathrm{E}_{pop} &\Bigg[x_1 \log \frac{p_1}{\hat{p}_1} + (1-x_1) \log \frac{1-p_1}{1-\hat{p}_1} + x_1 \sum_{i=2}^{m} \left(x_i \log \frac{p_i^1}{\hat{p}_i^1} + (1-x_i) \log \frac{1-p_i^1}{1-\hat{p}_i^1} \right) \\ &+ (1-x_1) \sum_{i=2}^{m} \left(x_i \log \frac{p_i^0}{\hat{p}_i^0} + (1-x_i) \log \frac{1-p_i^0}{1-\hat{p}_i^0} \right) \Bigg] \\ = \frac{1}{2n} &+ \sum_{i=2}^{m} \left[p_1 \times \frac{1}{2np_1} + \frac{1}{2n^2} \left[\frac{1-p_1}{p_1} \right] \right] + \sum_{i=2}^{m} \left[(1-p_1) \times \frac{1}{2n(1-p_1)} + \frac{1}{2n^2} \left[\frac{p_1}{1-p_1} \right] \right] \\ = \frac{2m-1}{2n} &+ O(mn^{-2}) \\ = \frac{C}{2n} &+ O(Cn^{-2})\end{aligned} \tag{31}$$

We can calculate the exact value of $\mathrm{E}_{pop}(L^2)$ using the equations (19), (25) and (26) as below :

$$\begin{aligned}
\mathrm{E}_{pop}(L^2) = \mathrm{E}_{pop} \Bigg[\Bigg[& x_1 \log \frac{p_1}{\hat{p}_1} + (1 - x_1) \log \frac{1 - p_1}{1 - \hat{p}_1} + x_1 \sum_{i=2}^{m} \left(x_i \log \frac{p_i^1}{\hat{p}_i^1} + (1 - x_i) \log \frac{1 - p_i^1}{1 - \hat{p}_i^1} \right) \\
& + (1 - x_1) \sum_{i=2}^{m} \left(x_i \log \frac{p_i^0}{\hat{p}_i^0} + (1 - x_i) \log \frac{1 - p_i^0}{1 - \hat{p}_i^0} \right) \Bigg]^2 \Bigg] \\
= & \frac{1}{n} + \sum_{i=2}^{m} \left[p_1 \times \frac{1}{np_1} + \frac{1}{n^2} \left[\frac{1 - p_1}{p_1} \right] \right] + \sum_{i=2}^{m} \left[(1 - p_1) \times \frac{1}{n(1 - p_1)} + \frac{1}{n^2} \left[\frac{p_1}{1 - p_1} \right] \right] \\
& + 2 \left[\binom{m-1}{2} \times \frac{p_1}{4n^2 \hat{p}_1^2} + \binom{m-1}{2} \times \frac{1 - p_1}{4n^2 (1 - \hat{p}_1)^2} + \frac{(m-1)(1-p_1)}{4n^2} + \frac{(m-1)(p_1)}{4n^2} \right] \\
= & \frac{2m - 1}{n} + \frac{(m-1)(m-2)}{4n^2} \left[\frac{p_1}{\hat{p}_1^2} + \frac{1 - p_1}{(1 - \hat{p}_1)^2} \right] + O(mn^{-2}) \\
\approx & \frac{C}{n} + \frac{m^2}{4n^2} \left[\frac{1}{p_1(1 - p_1)} \right] + O(mn^{-2})
\end{aligned} \tag{32}$$

Combining equations (31) and (32), we have the variance for Naive Bayes as:

$$\begin{aligned}
\mathrm{Var}_{pop}(L) &= \mathrm{E}_{pop}(L^2) - (\mathrm{E}_{pop}(L))^2 \\
&= \frac{C}{n} + \frac{m^2}{4n^2} \left[\frac{1}{p_1(1 - p_1)} - 4 \right] + O(mn^{-2}) \\
&\approx \frac{C}{n} + O(C^2 n^{-2})
\end{aligned} \tag{33}$$

G Understanding the complexity metric - Parameter estimation errors

The complexity of a Bayesian network $\langle G, \theta \rangle$ with discrete random variables is the number of independent parameters used to define its probability distribution.

$$C(\langle G, \theta \rangle) = \sum_{i=1}^{m} |V(Pa_{X_i}^{G})| (|V(X_i)| - 1)$$

The parameters θ are estimated from the pool data. To understand the privacy risk of this learning to members of the pool, we need to study the influence a member can have on the value of the parameters. Fisher information quantifies the amount of information a random variable carries about the parameter(s) θ of the probability distribution from which it is generated.

$$I(\theta) = -\mathrm{E}_{\theta}(\nabla^2 l(\theta)),$$

where $I(\theta)$ is Fisher information, and $l(\theta)$ is the log-likelihood function for θ.

If $\hat{\theta}$ is a Maximum Likelihood Estimate of θ, then it is known that

$$\hat{\theta} = Normal(\theta, I(\hat{\theta})^{-1}).$$

The log-likelihood functions of parameter θ from a PGM $\langle G, \theta \rangle$, given a sample x are typically of the form:

$$\begin{aligned}
l(\theta) &= \log \left[\Pr(x; \langle G, \theta \rangle) \right] \\
&= \sum_{i=1}^{m} l_i
\end{aligned}$$

where l_i is contribution of X_i to the likelihood function:

$$l_i = \sum_{v \in V(Pa^G_{X_i})} 1_{\{Pa^G_{X_i}=v\}} l_i^v$$
$$l_i^v = \sum_{o \in V(X_i)} 1_{\{x_i=o\}} \log p^v_{io}$$

$$l = \sum_{i,v,o} f_{i,v,o}(x_1, x_2, \ldots, x_m) \log(p^v_{io}),$$

where $f_{i,v,o}$ are activator functions (some combination of x_i's) for the parameter p^v_{io}.

$$I(p) = -E_p(\nabla^2 l(p))$$

All the non-diagonal elements of the information matrix are zero, because:

$$\frac{\partial}{\partial p^v_{io}} \frac{\partial}{\partial p^v_{jo}} [\sum_{i,v,o} f_{i,v,o}(x_1, x_2, \ldots, x_m) \log(p^v_{io})] = 0, \forall i \neq j$$

This implies that all the standard normal variables used to represent frequencies in pool are pair-wise independent i.e. all the estimation errors are independent across parameters. The difference between an estimated parameter value (calculated from the pool) and the actual parameter value (calculated from the general population) is the estimation error that leaks information about the pool. Since these estimation errors are independent across parameters of a Bayesian network, each parameter makes a separate contribution to the power of the attacker. Hence, the complexity measure defined as the number of independent parameters, captures the potential privacy risk of the model.

H What about models trained with differential privacy?

The bound provided in Theorem 1 is computed assuming that the parameters are learned without any privacy defense. The parameters can also be learned with a privacy defense (like differential privacy) in place. The effect of a differentially private learning mechanism on our bound can be better reasoned under the recently introduced notion of "f-differential privacy" (f-DP) [2]. f-DP is a new relaxation of differential privacy based on a framework of hypothesis testing. It characterizes the trade-off between type I and type II errors in distinguishing any two neighboring datasets using a function f. When the function f is from a specific family that characterizes the trade-off between type I and type II errors in distinguishing the two normal distributions $\mathcal{N}(0, 1)$ and $\mathcal{N}(\mu, 1)$ based on one draw, it is said to be μ-GDP. If the learning mechanism satisfies μ-GDP, then the bound on power of membership inference in Theorem 1 will become:

$$z_\alpha + z_{1-\beta} \leq \mu \tag{34}$$

Corollary 2.13 in the paper [2] provides the relationship between μ-GDP and the standard $(\epsilon, \delta) - DP$.

Corollary 1 [2]: A mechanism is μ-GDP if and only if it is $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$, where

$$\delta(\epsilon) = \Phi\Big(-\frac{\epsilon}{\mu} + \frac{\mu}{2}\Big) - \mathrm{e}^\epsilon \Phi\Big(-\frac{\epsilon}{\mu} - \frac{\mu}{2}\Big)$$

, and Φ is the CDF of standard normal distribution.

Using Corollary 1 and equation 34, we can calculate how our bound in Theorem 1 changes when the parameters are learned with differential privacy guarantees.

I Evaluation details: Bayesian network learning and Data synthesis

In this section, we describe the methods used in the evaluation for learning structure and parameters of a Bayesian network and generating synthetic data. See [5] for a comprehensive overview of the methods to learn Bayesian networks.

I.1 Structure Learning

The objective is to learn the significant dependencies between random variables, and represent them as a graph. We used an existing algorithm based on maximizing a score function that measures how correlated different attributes are, according to the training data [4]. For each attribute we find a set of attributes which are highly correlated with it, yet are not significantly correlated among themselves.

$$score(Pa^G_{X_i}) = \frac{\sum_{X_j \in Pa^G_{X_i}} corr(X_i, X_j)}{\sqrt{|Pa^G_{X_i}| + \sum_{x_j, x_k \in Pa^G_{X_i}} corr(X_j, X_k)}}, \tag{35}$$

$$corr(X_i, X_j) = 2 - 2\frac{H(X_i, X_j)}{H(X_i) + H(X_j)},$$

where H is the entropy function.

While optimizing this score for each attribute, we need to make sure that the graph remains acyclic. Also, to control the complexity of the graph, we impose a condition on η, the maximum number of parents for each node. We use an iterative and greedy algorithm that adds parents to each node while maximizing the score for all nodes at each iteration, subject to the constraints.

I.2 Parameter Learning

We assume a prior distribution on all possible values of the parameters θ, and use the training data set to update this distribution, using a Bayesian approach.

Let X_i be the random variable for a categorical attribute. Let $\vec{\theta}_i$ be the parameters of the conditional probability $\Pr[X_i | Pa^G_{X_i}; \theta]$. For each assignment of values to $Pa^G_{X_i}; \theta$, we assume a prior distribution on all the possible k-dimensional multinomial distributions. The prior distribution for each assignment v comes from a Dirichlet family, i.e., $\vec{\theta}^v_i \sim Dirichlet(\vec{\alpha}^v_i)$, where $\vec{\alpha}^v_i$ is the hyper parameters of the distribution.

Let $\vec{c}^v_i = [c^v_{i1}, c^v_{i2}, \cdots, c^v_{ik}]$ include the frequency of the events $[X_i = j | Pa^G_{X_i}; \theta = v]$ in training data. We compute the posterior distribution for $\vec{\theta}^v_i$ as $Dirichlet(\vec{\alpha}^v_i + \vec{c}^v_i)$. Thus, the most likely estimation for set of parameters $\vec{\theta}^v_i$ is:

$$\theta^v_{ij} = \frac{\alpha^v_{ij} + c^v_{ij}}{\sum_{j=1}^k (\alpha^v_{ij} + c^v_{ij})}. \tag{36}$$

In all our experiments, we use a uniform prior i.e., we set $\vec{\alpha}^v_i$ to 1 in all dimensions.

I.3 Data Synthesis

Given a data set D, we want to synthesize datasets that are close in distribution to D. Graphical models could be used for inference and prediction, as well as generating synthetic data (from the underlying distribution that they encode). We use the below process for generating synthetic datasets:

1. Learn a Bayesian network $\langle G, \theta \rangle$ from the data set D.
2. Create a Bayesian network $\langle G', \theta' \rangle$ with $G' = G$, and θ' drawn from the posterior Dirichlet distribution for θ, which was computed during parameter learning.
3. Draw independent samples from $\langle G', \theta' \rangle$.

In our experiments, while generating the synthetic data, we use $\eta = 3$ for learning the structure G of the Bayesian network $\langle G, \theta \rangle$ from the data set D.

J Additional evaluation

J.1 Effect of releasing statistically insignificant edges

We analyze the effect on the power of attack of releasing edges (conditional probabilities) that are statistically insignificant. To perform this evaluation, we consider the case where the structure of released model is not learned from data but generated in a random way.

Figure 1 compares the power of the attack when two different models of almost equal complexity are released. The structure of first model is generated by randomly adding edges and the structure of second model is learned from data. Adversary uses the released model as the population model to calculate the LR statistic and perform tracing attack. We can observe that the attack power is similar in both the cases.

The edges that are generated randomly might not be statistically significant, but they leak about membership. In the LR Test for tracing attack, we rely on the difference between probability distributions for pool and population. Adding a statistically insignificant edge gives similar probability distributions for all configurations of the parent. Although the conditional probabilities are similar, their values will be different for pool and population and hence they will leak about membership. **Statistically insignificant edges leak as much information about membership as significant edges.**

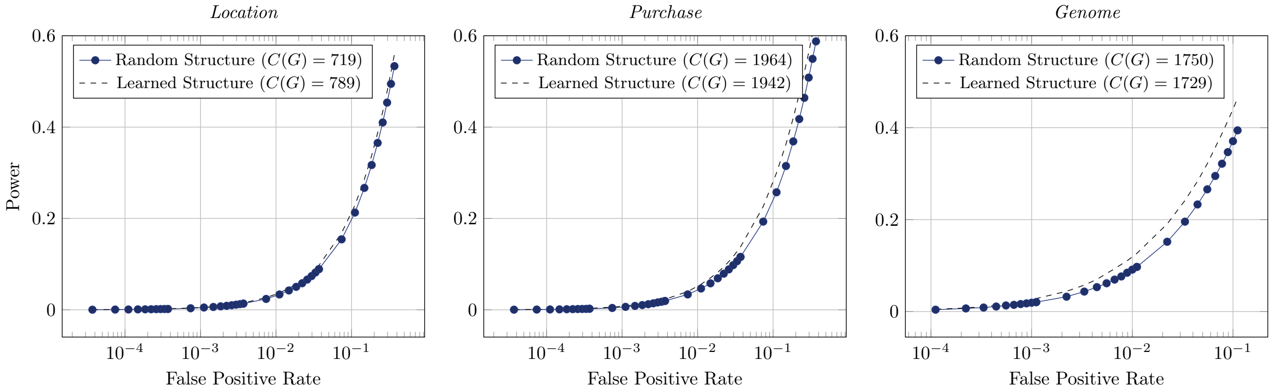


Figure 1: **Effect of Releasing Graphical Models with Random Edges:** Here we compare the power of attack, when two models of similar complexity learned on the dataset are released but structure of one model is learned from data and the structure of other is generated randomly. We can see that for close values of C, the power of attack is almost same for both the models. This shows that statistically insignificant edges leak as much information about membership, as that of significant edges.

J.2 Optimality of the theoretical threshold

In Figure 2, we compare theoretical thresholds for certain false positive rates with their corresponding values estimated using the reference population. The adversary has access to some reference population. For a given false positive rate, the adversary chooses the threshold based on the likelihood ratio on the reference population data. The attacker then runs the LR test tracing attack. When $\eta = 0$ (row 1), we observe that the theoretical threshold values are much higher than the estimated values. When $\eta = 3$ (row 2), the observed thresholds are closer to the estimated values.

When $\eta = 0$, the parameter estimation errors are correlated, which reduces the amount of information leakage. The adversary, when using the theoretical threshold, overestimates the amount of leakage (power) and hence chooses a higher threshold. When $\eta = 3$, the released model captures most of the dependencies among attributes in the data. Hence the observed threshold will be closer to the theoretical threshold values. **From the adversary's perspective, the theoretical threshold value is sub-optimal when the released model is underfitted (loss of utility).**

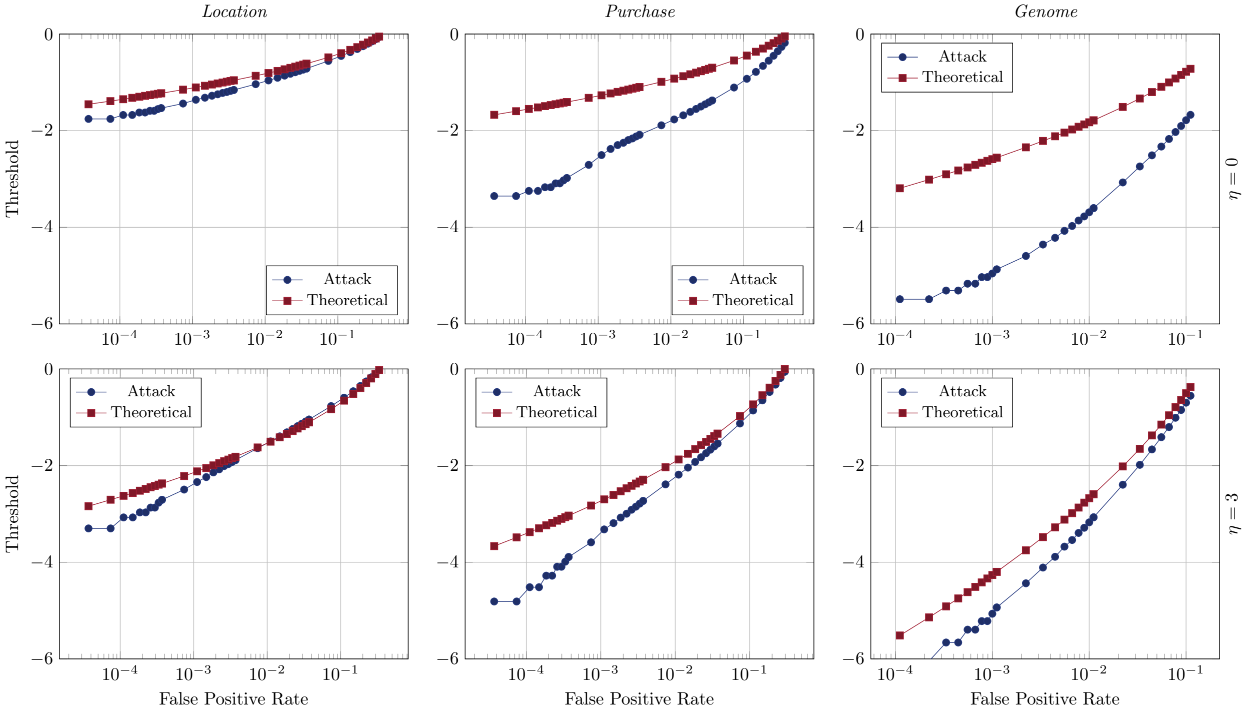


Figure 2: **Effect of releasing underfitted models on threshold selection:** This plot compares the threshold values estimated by the adversary using reference population at different false positive rates with their corresponding theoretical values. The label Attack indicates that the threshold is estimated by the adversary using reference population. We observe that for underfit models ($\eta = 0$) (first row), the threshold value estimated from the reference population is way less than the corresponding theoretical value. As the model gets closer to the generator distribution ($\eta = 3$) (second row), the estimated threshold values get closer to the theoretical values.

J.3 Effect of using a complex model for estimating likelihood of Null hypothesis

In this subsection, we present the effect of population model choice on the behavior of the likelihood ratio test and on the power of the tracing attack. Specifically, we study the effect of using models that are more complex than the released model as population model. The parameters of a graphical model $\langle G, \hat{\theta} \rangle$ with $\eta = 1$ are learned on the pool data and released. The adversary has access to a complex and better representative model $\langle G_{pop}, \theta \rangle$ that was learned with $\eta = 3$. The adversary can choose to use either the released model structure G or a complex model structure G_{pop} as population model structure.

Figure 3 compares the empirical distribution of test statistic (likelihood ratio) values computed on members of the pool and on non-members for both choices of population model on the genome dataset. On the right, we observe that the member distribution is indistinguishable from the non-member distribution when G_{pop} (learned with $\eta = 3$) is used as structure of population model. We also observe that the values of the likelihood ratio are much higher – from 20 to 70 – compared to the values we observe on the left (narrowly concentrated around 0) when the structure of population model is same as that of released model (learned with $\eta = 1$).

When a complex model is used as population model, the likelihood value of the null hypothesis increases for both members and non-members. Hence it cannot help in distinguishing members from non-members. *Also it changes the meaning of the hypothesis test. When a complex model is used to compute the likelihood of null hypothesis, the computed likelihood is no longer the likelihood of the target being a random sample from the population. The meaning of this new hypothesis test would be the following: which of the* ***models is more likely*** *given the target. Since complex models are more likely compared to simpler models, the test statistic (likelihood ratio) values will be very high and positive.* Figure 4 compares the power of the tracing attack for both choices of population model. The power of the attack is higher when the released model structure is used as the population model structure. **Knowledge of additional statistics about population other than the released statistics doesn't increase the power of adversary.**

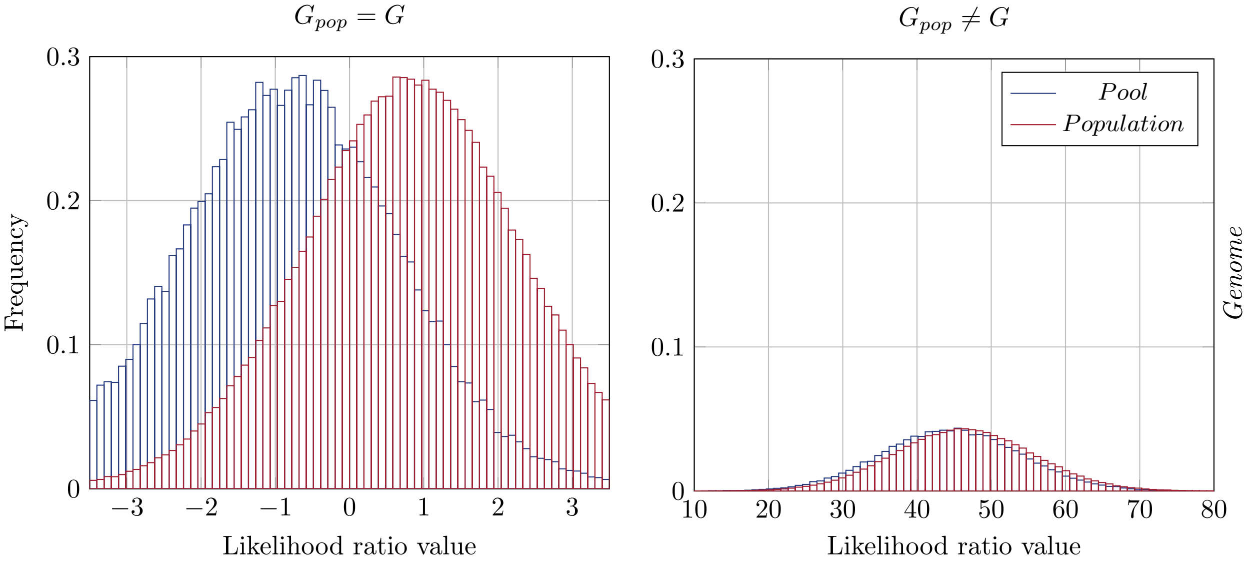
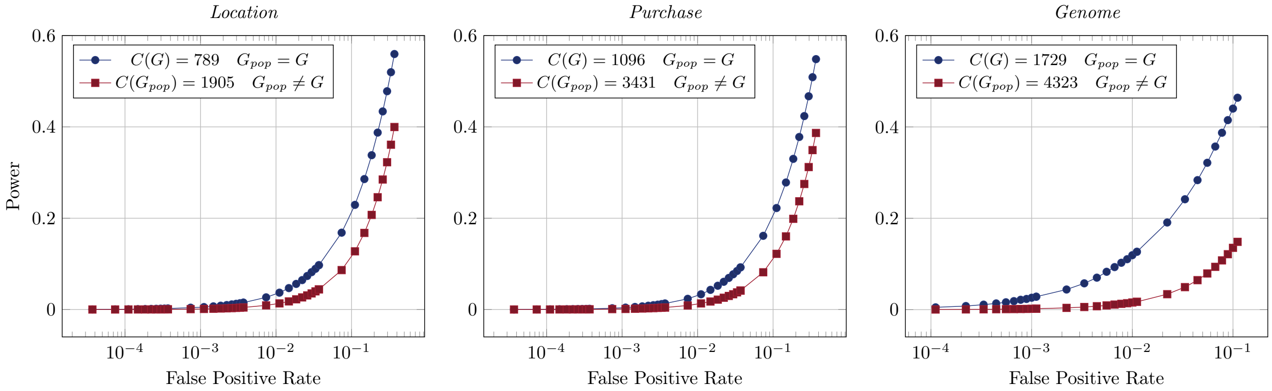


Figure 3: **Comparison of likelihood ratio distributions computed on members of the pool (blue histogram) and on non-members (red histogram): Left:** To calculate the likelihood of the null hypothesis H_{OUT}, we use the population model $\langle G, \theta \rangle$, whose structure is the same as that of the released model $\langle G, \hat{\theta} \rangle$ ($\eta = 1$). We observe that the member distribution is clearly distinguishable from the non-member distribution. **Right:** To calculate the likelihood of the null hypothesis H_{OUT}, we use the population model $\langle G_{pop}, \theta \rangle$, whose structure is different and more complex ($\eta = 3$) than the released model $\langle G, \hat{\theta} \rangle$ ($\eta = 1$). We observe that the member/non-member distributions are indistinguishable. Also, the values of the likelihood ratio are much higher compared to the left part of the figure. Using a complex population model might increase the likelihood of null hypothesis H_{OUT}, but it increases the value for both members and non-members (as a complex model can explain both members and non-members better than a simple model can), making them indistinguishable. Hence the optimal choice of population model for the adversary is the released model estimated over the reference population.

Figure 4: **Effect of using a complex model as population model:** The parameters of a graphical model $\langle G, \hat{\theta} \rangle$ with $\eta = 1$ are learned on the pool data and the model is released. The adversary has access to a better (more complex) generative model $\langle G_{pop}, \theta \rangle$ with ($\eta = 3$). We observe how the power of the attack changes when calculating the likelihood of null hypothesis using this complex generative model structure instead of the released model structure. We can see that the power of the attack reduces when the population model structure is not the same as the released model structure. As shown in Figure 3, using a complex population model increases the likelihood for both members and non-members and hence cannot help in distinguishing them. This shows that it is not possible to increase the power of adversary using knowledge about additional statistics on the data that are not present in the released graphical model.



J.4 Effect of Biased Sampling

In this section, we empirically study the effect of sampling bias on the power of tracing attack. We model a case of sampling bias, where we discriminate against individuals with some attribute value (say 1). We add a bias in the sampling mechanism for pool, by making the probability of selecting an attribute with value 1 as $1 - bias$.

$$Pr(select|X_i = 0) = 1 \tag{37}$$

$$Pr(select|X_i = 1) = 1 - bias \tag{38}$$

Synthetic data for this experiment was generated from graphs learned on Genome data. Pool is sampled in a biased way as described above. The parameter *bias* can be used alter the amount of sampling bias. We generate a total of 10000 samples, of which we randomly select 2000 as pool and 4000 as reference population.

Figure 5 shows the effect of bias on power of attack. We can clearly observe that power of attack increases with increase in bias. When the pool is drawn from same distribution as population, we leveraged on finite sample estimation error for membership inference. If pool is drawn from distribution that is even slightly different from population distribution, power of attack increases and can be greater than provided bounds. **Biased sampling increases the power of tracing attack**

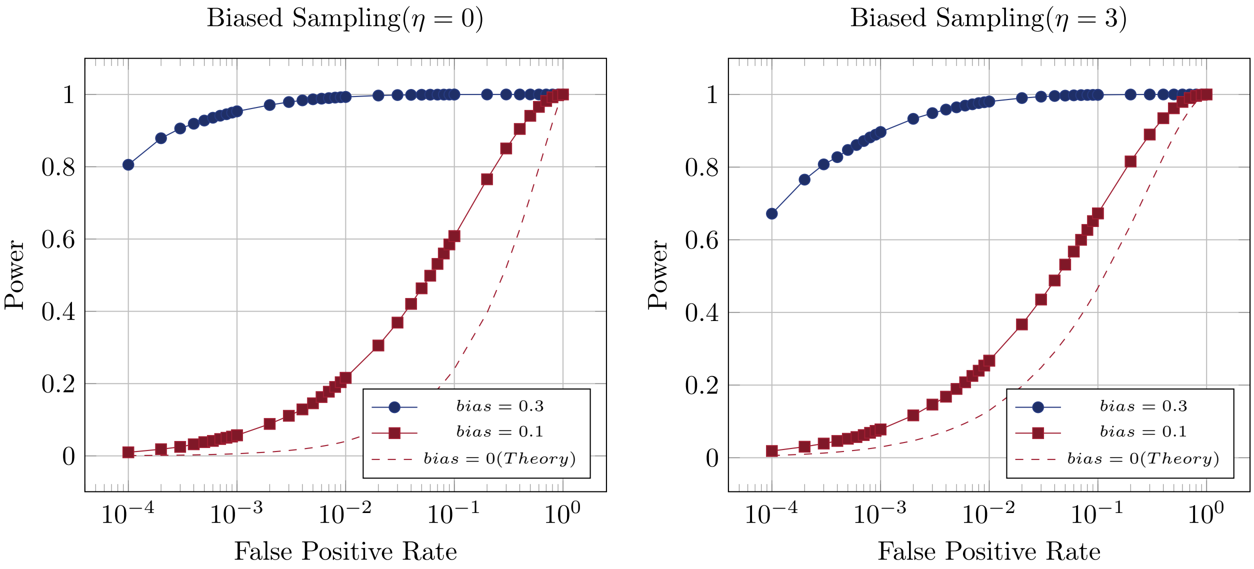


Figure 5: **Comparison of Power values in case of biased sampling:** Figure shows the effect of sampling bias on the power of tracing attack. We generate synthetic data using graph structures of different complexity that are learned on Genome data. The conditional probability values are generated from a Dirichlet distribution fitted to the conditional probabilities in corresponding graph of Genome data. The parameter *bias* is used to tune the bias in sampling of the pool. We can observe that power of attack in case of biased sampling is greater than the theoretical bound (with out considering bias). With increasing value of *bias*, the pool distribution deviates more from the population distribution, which increases the power of attack.

References

[1] A. C. Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

[2] J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

[3] C.-G. Esseen. On the liapunoff limit of error in the theory of probability. *Arkiv för matematik, astronomi och fysik*, 1942.

[4] M. A. Hall. Correlation-based feature selection for machine learning. 1999.

[5] D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.