Quantifying the Privacy Risks of Learning High-Dimensional Graphical Models

Sasi Kumar Murakonda
National University of Singapore
murakond@comp.nus.edu.sg

Reza Shokri
National University of Singapore
reza@comp.nus.edu.sg

George Theodorakopoulos
Cardiff University
TheodorakopoulosG@cardiff.ac.uk

Abstract

Models leak information about their training data. This enables attackers to infer sensitive information about their training sets, notably determine if a data sample was part of the model's training set. The existing works *empirically* show the *possibility* of these membership inference (tracing) attacks against complex deep learning models. However, the attack results are dependent on the specific training data, can be obtained *only after* the tedious process of training the model and performing the attack, and are missing any measure of the confidence and unused potential power of the attack.

In this paper, we *theoretically* analyze the maximum power of tracing attacks against high-dimensional graphical models, with the focus on Bayesian networks. We provide a tight upper bound on the power (true positive rate) of these attacks, with respect to their error (false positive rate), for a given model structure *even before* learning its parameters. As it should be, the bound is independent of the knowledge and algorithm of any specific attack. It can help in identifying which model structures leak more information, how adding new parameters to the model increases its privacy risk, and what can be gained by adding new data points to decrease the overall information leakage. It provides a measure of the potential leakage of a model given its structure, as a function of the model complexity and the size of the training set.

1 Introduction

How much is the privacy risk of releasing high-dimensional models which are trained on sensitive data? We focus on measuring information leakage of models about their training data, using tracing (membership inference) attacks. In a tracing attack, given the released model and a target data sample, the adversary aims at inferring whether or not the target sample was a member of the training set. We use the term tracing attack and membership inference attack interchangeably (Dwork et al., 2017; Shokri et al., 2017). The attack is evaluated based on its power (true positive rate), and its error (false positive rate), in its binary decisional task.

Tracing attacks have been extensively studied for summary statistics, where independent statistics (e.g., mean) of attributes of high-dimensional data are released. Homer et al. (2008) showed the existence of powerful tracing attacks; more recent work provided theoretical frameworks to analyze the upper bound on the power of these inference attacks (Sankararaman et al., 2009), and their robustness to noisy statistics (Dwork et al., 2015). The theoretical analysis helps explain the major causes of information leakage and assess the privacy risk even before computing the statistics. However, these analyses are limited to simple models such as product distributions.

Advanced machine learning models, such as deep neural networks, have recently been tested against tracing attacks. In the black-box setting, the attacker can only observe predictions of the model. The attack involves training inference models that can distinguish between members and non-members from the predictions that the target model produces (Shokri et al., 2017). The attacks are tested against deep neural networks as well as machine-learning-as-a-service platforms, and their accuracy is shown to be related to their generalization error (Shokri et al., 2017; Yeom et al., 2018). In the white-box setting, the attacker obtains the parameters of the model, and decides that the target sample is

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

a member if the gradients of the loss with respect to the model's parameters computed on the target data sample are aligned with the model's parameters (Nasr et al., 2019). Large models are empirically shown to be more vulnerable to the attack, even if they have better generalization performance. These attacks highlight the susceptibility of high-dimensional neural networks to tracing attacks. However, their analysis is limited to empirical measurements of the attack success, after training the models on particular data sets.

Contributions Using the above-mentioned existing methods, it is possible to reason theoretically about tracing attacks, yet only for simple models (independent statistics). In parallel, it is possible to perform empirical tracing attacks against complex models (deep neural networks), yet without much theoretical analysis on the maximum power of inference attacks. In this paper, we aim at addressing this gap by providing a theoretical analysis of tracing attacks against high dimensional graphical models, i.e. models with many parameters. We provide a bound on the performance of tracing attacks to quantify the privacy risk that learning a model, with max likelihood estimation (MLE), implies for the training set. Using the bound, one can determine the elements of a released model that contribute to the power of the attacker. Our focus is on *probabilistic graphical models*, which are very general statistical models that capture the correlations among data attributes, as they are among fundamental models for machine learning, and the basis of deep learning models via e.g., restricted Boltzmann machines.

We use the likelihood ratio test (LR test) as the foundation of our tracing attack (Sankararaman et al., 2009). This enables us to **design the most powerful attack** against any probabilistic model. Thus, for any given error, there exists no other attack strategy that can achieve a higher power. Our objective is not to empirically evaluate the performance of attacks (even the theoretically strongest one) on trained models. We instead **compute the maximum achievable power of tracing attacks**. This upper bound can be used as a measure to evaluate the effectiveness of different attack algorithms by comparing their achieved power to the bound for any false positive error.

Our objective is to identify the elements of a model that cause membership information leakage and measure their influence. We prove that, for a given model structure, the potential leakage of the model (the leakage that corresponds to the most powerful attack for any given error) is proportional to the square root of model's complexity (defined as the number of its independent parameters), and is inversely proportional to the square root of the size of the training set. Thus, the theoretical bound enables us to **quantify the potential leakage of a model before even learning the parameters of the model** on that structure. This can be used to efficiently compare different model structures based on their susceptibility to tracing attacks. The theoretical bound can quantify the power that the attack gains/loses if a new attribute is added/removed from the data, or when the model requires capturing/removing the correlation between certain attributes. It can also determine the size of the training set for a very high-dimensional model that leaks a similar amount of information as a small model leaks on a small set of data.

We evaluate our attack against real (sensitive) data: location check-ins, purchase history, and genome data. We empirically show that the upper bound is tight, and the power (true positive rate) of the likelihood ratio test attacker is extremely close to the bound for any error (false positive rate).

2 Probabilistic Graphical Models

Probabilistic graphical models make use of a graph-based representation to encode the dependencies and conditional independence between random variables (Koller et al., 2009). Each node X_i in a graph G is a random variable, and the edges represent the dependencies. In this paper, we focus on **Bayesian networks**. However, our attack framework is easily applicable to Markov random fields. In Bayesian networks, the model structure is a directed acyclic graph, and the model $\langle G, \theta \rangle$ enables factoring the joint probability over the random variables. Given all its parents in the graph, a random variable is conditionally independent of other random variables. Therefore, the joint distribution can be factored as follows.

$$\Pr[X_1, X_2, \cdots, X_m] = \prod_{i=1}^{m} \Pr[X_i | Pa^G_{X_i}; \theta], \qquad (1)$$

where $Pa^G_{X_i}$ is the parent random variables of X_i. The parameters θ of the model encode the conditional probabilities.

We define the **complexity** $C(G)$ of a Bayesian network $\langle G, \theta \rangle$ with discrete random variables as the number of independent parameters used to define its probability distribution. Let $V(X)$ be the number of distinct values that a random variable X can take. For each conditional probability $\Pr[X_i | Pa^G_{X_i}; \theta]$, we need $|V(Pa^G_{X_i})|(|V(X_i)| - 1)$ independent parameters. Thus, the total complexity of a model is:

$$C(G) = \sum_{i=1}^{m} |V(Pa^G_{X_i})|(|V(X_i)| - 1). \qquad (2)$$

3 Problem Statement

We consider a set of n independent m-dimensional data samples from a *population*. We refer to this set as the *pool*. We do not make any assumption about the probability distribution of the data points in the general population. Given a graphical model structure G, the pool data is used to train a graphical model, i.e., to estimate the parameters $\hat{\theta}$ of the probabilistic graphical **model** $\langle G, \hat{\theta} \rangle$. The estimation can be either Max likelihood estimation (MLE) or Max a posteriori (MAP) estimate. This model is *released*. Our objective is to quantify the privacy risks of releasing such models for the members of their training data.

Let us consider an adversary who observes the released model $\langle G, \hat{\theta} \rangle$. We assume that the attacker can collect a set of independent samples from the population. We refer to this set as the *reference population*. The objective of the adversary is to perform a **tracing attack** (also known as membership inference attack) against the released model, on any target data point x: create a decision rule that determines whether x was used in the training of the parameters of $\langle G, \hat{\theta} \rangle$ or not, i.e. to classify x as being in the pool (IN) or not (OUT).

The accuracy of the tracing attack indicates the information leakage from the model about the members of its training set. We quantify the attacker's success using two metrics: the adversary's **power** (the true positive rate), and his **error** (the false positive rate). The power measures the conditional probability that the attacker classifies x as IN, given that x is indeed in the pool, i.e. $\Pr[IN|x \in \text{pool}]$. The error measures the conditional probability that the attacker classifies x as IN, given that x is not in the pool, i.e. $\Pr[IN|x \notin \text{pool}]$. The ROC curve (Receiver Operating Characteristic), which is a plot of power versus error, captures the trade-off between power and error. Thus, the area under the ROC curve (AUC) is a single metric for measuring the strength of the attack. The AUC can be interpreted as the probability that a randomly drawn data point from the pool will be assigned larger probability of being IN than a data point randomly drawn from outside the pool. So AUC = 1 implies that the attacker can correctly classify all data samples as IN or OUT.

3.1 Membership Inference Attack against Graphical Models

Given the released model, the reference population, and the target data point, the adversary aims at distinguishing between two hypothesis. Each hypothesis describes a possible world that could have resulted in the observation of the adversary, where in one world the target data was part of the training set (pool), while in the other one the target data is a random sample from the population.

- Null hypothesis (H_{OUT}): The pool is constructed by drawing n independent samples from the general population. Parameters $\hat{\theta}$ of the model $\langle G, \hat{\theta} \rangle$ are learned on the pool data. Target data x is drawn from the general population, independently from the pool.
- Alternative hypothesis (H_{IN}): The pool is constructed by drawing n independent samples from the general population. Parameters $\hat{\theta}$ of the model $\langle G, \hat{\theta} \rangle$ are learned on the pool data. Target data x is drawn from the pool.

This generalizes the hypothesis test designed by Sankararaman et al. (2009), in which the released model follows a product distribution, i.e. the random variables corresponding to data attributes are independent (equivalent to a probabilistic graphical model without any dependency edges between the nodes).

We use the Likelihood Ratio test to distinguish the two hypotheses. The goal of hypothesis testing is to find whether there is **enough evidence to reject the null hypothesis *in favor of the alternative hypothesis*,** i.e. whether the likelihood L_{IN} of the alternative hypothesis is large enough compared to the likelihood L_{OUT} of the null hypothesis. The only information we know about the pool is $\hat{\theta}$, the parameters of the released model learned using the pool data. Hence, we must calculate these *exact same parameters* under null hypothesis (i.e., learn the parameters using general population). Let θ be the result of this computation, i.e., the parameters of G trained on a large reference population. We calculate L_{IN} as the likelihood of the parameters of G taking the value $\hat{\theta}$, which is equal to $\Pr[x; \langle G, \hat{\theta} \rangle]$. Similarly, we calculate L_{OUT} as the likelihood of the parameters of G taking the value θ, which is equal to $\Pr[x; \langle G, \theta \rangle]$.

Hence, the log likelihood statistic is computed as follows.

$$L(x) = \log\left(\frac{\Pr[x; \langle G, \theta \rangle]}{\Pr[x; \langle G, \hat{\theta} \rangle]}\right) \tag{3}$$

The LR test is a comparison of the log likelihood statistic $L(x)$ with a threshold. If $L(x) \leq$ threshold, then the attacker decides in favor of H_{IN} (rejects H_{OUT}); else, he decides in favor of H_{OUT} (more precisely, he fails to reject H_{OUT} because there is not enough evidence to support this rejection in favor of H_{IN}). To determine the threshold, the attacker selects a (false positive rate) error α that he is willing to tolerate. He then empirically or theoretically estimates the distribution of $L(x)$ under the null hypothesis, using his reference population. We denote the CDF of this distribution as F. Given α and F, the attacker computes a threshold value $F^{-1}(\alpha)$ and compares it to $L(x)$, to decide whether to reject the null hypothesis. This concludes the hypothesis test.

The *power* of the test, as defined earlier, can be expressed as $\Pr[L(x) \leq F^{-1}(\alpha)]$, computed under the alternative hypothesis, for an individual data point x randomly drawn from the pool. In other words, it is the fraction of pool data points that are correctly classified by the test. By varying α, and thus the threshold $F^{-1}(\alpha)$, we can draw the ROC curve and compute the AUC metric. It is worth emphasizing that according to the Neyman and Pearson (1933) lemma, the LR test achieves the **maximum power** among all decision rules with a given error (false positive rate). So, any other decision rule would result in a lower AUC.

4 Theoretical analysis - Bound on power of attack

Our objective is to compute the maximum power β for any false positive error α of an adversary that observes the released model $\langle G, \hat{\theta} \rangle$ which has been trained on a pool of size n. In our main result, Theorem 1, we show which combinations of α and β are possible for the attacker, and we find the major factors that determine these combinations, as a function of the model complexity and size of the dataset. To derive our main result about the best achievable power-error tradeoff, we assume that the released parameters satisfy the below conditions.

- The value of every released parameter is learned from a large enough number of samples for the central limit theorem to hold good.
- The value of every released parameter is non-trivial i.e., it is bounded away from 0 and 1 (Sankararaman et al., 2009).

These are valid assumptions to make on part of the model publisher, as it is not beneficial to publish statistically insignificant or trivial estimates. In fact, the recently published methodology of learning Bayesian Networks on Cancer Analysis System (CAS) database in the National Cancer Registration and Analysis Service (NCRAS) has similar assumptions (they use only the parameters that are learned using at least 50 samples)[1]

Theorem 1. *Let β and α be the power and error of the LR test, for the membership inference attack, respectively. Let n be the size of the pool (model's training set), and $C(G)$ be the complexity of the released probabilistic graphical model $\langle G, \hat{\theta} \rangle$. Then, the tradeoff between power and error follows the following relation:*

$$z_\alpha + z_{1-\beta} \approx \sqrt{\frac{C(G)}{n}}, \tag{4}$$

where z_s is the quantile at level $1-s, 0 < s < 1$ of the Standard Normal distribution.

Proof sketch. To compute $\beta = \Pr_{pool}\{L(x) \leq F^{-1}(\alpha)\}$, the power of the LR test for the inference attack, for any error α, we need the distribution of $L(x)$ when x is drawn from the pool and when x is drawn from the population. Our approach to estimating the distributions of $L(x)$ is through computing its moments $\mathrm{E}(L^k), k > 0$. To approximate the distribution using its moments, we use an established statistical principle for fitting a distribution with known moments: the maximum-entropy principle. This principle states that the probability distribution which best represents the current state of knowledge is the one with largest entropy (Jaynes, 1957a,b).

To simplify the computation of the moments, we take advantage of the Bayesian decomposition to split this $L(x)$ as sum of simpler terms (one for each attribute X_i). We start by expanding (3) to give the following expression for $L(x)$:

$$\begin{aligned} L(x) =& \log\left(\frac{\prod_{i=1}^{m} \Pr[X_i | Pa^G_{X_i}; \theta]}{\prod_{i=1}^{m} \Pr[X_i | Pa^G_{X_i}; \hat{\theta}]}\right) \\ =& \sum_{i=1}^{m} \log\left(\frac{\Pr[X_i | Pa^G_{X_i}; \theta]}{\Pr[X_i | Pa^G_{X_i}; \hat{\theta}]}\right) \end{aligned} \tag{5}$$

where the X_i are the attributes of the data point x, which is now a random variable as it is drawn from the pool (or population), as just mentioned. We define L_i as the contribution of attribute X_i to the likelihood ratio L. Hence the value of L_i can be calculated as:

$$L_i = \log\left(\frac{\Pr[X_i | Pa^G_{X_i}; \theta]}{\Pr[X_i | Pa^G_{X_i}; \hat{\theta}]}\right) \tag{6}$$

We calculate the first two moments of $L(x)$ for our approximation. The mean and variance of $L(x)$ are $\mu_0 = \frac{C(G)}{2n}, \sigma_0^2 = \frac{C(G)}{n}$ under the null hypothesis and $\mu_1 = -\frac{C(G)}{2n}, \sigma_1^2 = \frac{C(G)}{n}$ under the alternative hypothesis (See proof in Appendix A). For a known mean μ and variance σ^2, the max-entropy distribution that matches the target distribution is a Gaussian $N(\mu, \sigma^2)$.

[1] Generating the Simulacrum A methodology overview https://simulacrum.healthdatainsight.org.uk/wp/wp-content/uploads/2018/11/Methodology-Overview-Nov18.pdf

Deriving higher order moments of $L(x)$ requires information about the exact distribution that generated the data. Note that in our analysis, we do not make any assumption on the distribution from which data is generated. We just assume that the pool and the population are from the same distribution. Making assumptions on the distribution that generated the data limits the practical utility of such bounds (as we want to estimate potential leakage from a model, before touching the data). See Appendix A for details on how the data generator distribution affects the distribution of the log-likelihood ratio.

Given this approximation, and the computed mean and variance, the relationship between power β, and error α is

$$\mu_0 - z_\alpha \sigma_0 = \mu_1 - z_\beta \sigma_1 \tag{7}$$

where z_s is the quantile at level $1-s, 0<s<1$ of the standard normal distribution. This equation can be derived by equating quantiles at level β, α in the pool and population distribution respectively.

Substituting $\mu_0, \sigma_0, \mu_1, \sigma_1$ into (7), we derive the main result. □

In case of high-dimension models, the log-likelihood ratio distribution is very close to normal distribution (as it is a sum of large number of independent random variables) and assuming it follows a normal distribution is a good approximation for most practical purposes. As we will show in the latter sections (Section 5.2 and Section 4.1), the contribution of higher order moments to the estimates of privacy risk and understanding of the sources of information leakage is marginal. We illustrate this by comparing our theoretical bound with the empirically observed maximum power for any error and explaining the power of attacks using parameter estimation errors.

The intuition behind our result is that the centers ($\mu_0 = \frac{C(G)}{2n}$ and $\mu_1 = -\frac{C(G)}{2n}$) of $L(x)$ under the null and alternative hypotheses are separated by a distance of $\frac{C(G)}{n}$. The overlap between the distributions is determined by variance $\frac{C(G)}{n}$ of the statistic, and the amount of the overlap between the two distributions determines the power $\beta = \Pr_{pool}\{L(x) \leq F^{-1}(\alpha)\}$ for any error α.

Our result generalizes that of Sankararaman et al. (2009) on releasing independent marginals. In their case, the released graph has no edges and nodes are binary variables. The complexity of such a graph is equal to the number of nodes m. Hence, for independent marginals we recover Sankararaman et al's relation:

$$z_\alpha + z_{1-\beta} = \sqrt{\frac{m}{n}}. \tag{8}$$

4.1 Insights from the bound:

The bound in Theorem 1 is independent of the exact values of the data in the pool and depends only on the metadata of the model: pool size n, number of attributes m and model structure G. This implies that the analysis is robust to varying the details of the dataset, but it is expressive enough to capture and resolve questions like the following:

- Which one of many model structures has the largest/smallest leakage?
- What is the additional leakage caused by releasing one more attribute for each data point in the pool?
- How do the dependencies among a certain group of attributes affect the leakage?
- How exactly does the pool size affect leakage?

Using the bound, we can observe and quantify the effect of releasing a model in terms of its complexity $C(G)$. Releasing more parameters helps the attacker, and we also see that e.g. quadrupling $C(G)$ would double the sum $z_\alpha + z_{1-\beta}$, thus reducing the error or increasing the power or both. The amount of improvement depends on how large the sum already is and there are diminishing returns. In contrast, increasing the pool size n has the opposite effect to increasing $C(G)$: the attack performance becomes worse. This makes sense, as a larger pool is more similar to (has more overlap with) the population, so it is more difficult for the attacker to distinguish between them.

It is also possible to see whether a heuristic attack can be improved by comparing its error and power to the ones implied by the main theorem for a given complexity and pool size. From the heuristic attack's error and power, we can compute the corresponding Standard Normal quantiles and compare their sum to $\sqrt{\frac{C(G)}{n}}$. If the sum is far from the bound, then the attack can be improved. From a defender's point of view, we can quantify the maximum leakage associated with releasing various models **without** having to train each model and **without** having to perform any attack. We can reason about the ultimate/maximum power of the attacker, e.g. one with perfect knowledge about the population, so as to guide our choice of a model to release.

It is also interesting to note that the natural complexity behind the structure of a graphical model captures its privacy risk. The difference between an estimated parameter value (calculated from the pool) and the actual parameter value (calculated from the general population) is an 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 parameters captures the potential privacy risk of the model. See Appendix G for a detailed discussion on why the estimation errors of parameters in graphical models are independent.

5 Experiments

We use two methods for performing and evaluating the attack:

1. **Theoretical:** Given a false positive rate and released model structure G, we use our main result (Theorem 1) to calculate the power, error, and AUC.

2. **Empirical:** In empirical analysis, we vary the threshold of LR test from $-\infty$ to $+\infty$ and calculate the power at each value of false positive rate. This is the maximum possible power that can be achieved. Hence we use the power and AUC values calculated here to compare with the bound presented in Theorem 1.

5.1 Data Sets

A summary of all the data sets which are used in our experiments is provided in Table 1.

Location: This is a binary data set containing the Foursquare location check-ins by individuals in Bangkok (Shokri et al., 2017). Each record corresponds to an individual and consists of binary attributes reflecting visits to different locations.

Purchase: This is a binary data set containing information about individuals and their purchases (Shokri et al., 2017). Each record corresponds to an individual and each attribute represents a product. A value of 1 at attribute j means that the individual purchased the product corresponding to attribute j.

Genome: OpenSNP[2] is an open source data sharing website, where people can share their genomic data test results. We obtained the data provided by OpenSNP and considered only the individuals sequenced by 23andme. We randomly selected 1000 SNPs on chromosome 1. Individuals with more than 2 missing values were filtered out. After this pre-processing, we were left with 2497 individuals and 1000 SNPs for each individual.

Bayesian Networks have been used to model genome sequences in (Agrahari et al., 2018; Su et al., 2013).

Data Set	# Attributes	Original Size	Augmented Dataset Size
Location	446	5010	30000
Purchase	600	30000	30000
Genome	1000	2497	10000

Table 1: **Summary of Datasets used:** As the size of the original dataset is small for Location and Genome data, we augment the original dataset with synthetic data generated independently from a Bayesian network with $\eta = 3$ learned on the original dataset, where η is the maximum number of parents a node can have. We use the full augmented set as the general population. See Appendix I for complete details on data synthesis.

We use a similar approach to model the SNPs as a Bayesian Network. Since humans are diploid, at each position, we have two bases i.e. three possible values. While releasing graphical models constructed from genomic data, we only estimate the Minor and Major Allele Frequencies. To calculate the probability of any combination, we assume independence and compute it as the product of Allele Frequencies.

Data augmentation and Evaluation method: As the size of the original dataset is small for Location and Genome data, we augment the original dataset with synthetic data sampled independently from a Bayesian network with learned $\eta = 3$ (maximum number of parents per node) on the original dataset. We use the full augmented set as the general population. See Appendix I for details on data synthesis, structure learning, and parameter learning in Bayesian networks. In all the experiments, the pool and reference population are sampled independently from the general population. To evaluate the attack, all available samples from the general population are used to compute power and false positives. The pool size and reference population size for experiments with the Location and Purchase datasets are 3000 and 15000 respectively. The pool size and reference population size for experiments with the Genome dataset are 1000 and 5000 respectively. We perform each experiment with 50 different and independent splits of pool and reference population and report the average statistics. **This random splitting and averaging ensures that the results are not biased by data augmentation or by a single instance of sampling the pool.**

5.2 Validity of the theoretical bound

We first present how the complexity of released graphical model affects the power of tracing attack. Table 2 and Figure 1 (column 1) show the AUC and power respectively of the tracing attack when models of various complexities are released. In Figure 1 (columns 2, 3), we compare the observed power of tracing attack with the bound from Theorem 1. Columns 2 and 3 correspond to releasing Bayesian Networks learned with

[2] https://opensnp.org/snps

Data set	No. of Nodes	η	No. of Edges	Complexity	AUC (Empirical)	AUC (Theoretical)
Location	446	0	0	446	0.5928	0.6074
		1	343	789	0.6337	0.6415
		2	566	1222	0.6655	0.6741
		3	757	1905	0.6998	0.7134
Purchase	600	0	0	600	0.5700	0.6241
		1	496	1096	0.6266	0.6654
		2	941	1942	0.6885	0.7153
		3	1358	3431	0.7541	0.7752
Genome	1000	0	0	1000	0.6729	0.7602
		1	729	1729	0.7875	0.8237
		2	1244	2706	0.8495	0.8776
		3	1712	4323	0.9058	0.9292

Table 2: **AUC comparison for model structures we learned on different datasets, with different complexities**. We compare the AUC values for empirical attack with the corresponding values computed using the theoretical bound. The variable η represents maximum number of parents a node can have in the graph. We can observe that the empirical values of AUC are closer to the bound and increase with increasing complexity of the model.

$\eta = 0$ and $\eta = 3$ respectively. The variable η represents maximum number of parents a node can have in the graph.

As shown in Table 2, the AUC values are comparatively smaller for the Purchase data set compared to that of the Genomes dataset. This is because, in case of Purchase data, we only have 600 attributes for a pool size of 3000. In case of Genomic data, we have 1000 attributes for a pool size of 1000. Even then, on Purchase data, if a Bayesian network with $\eta = 3$ is released, we can achieve an AUC value of approximately 0.75, compared to 0.57 when only marginals are released. The complexity of a graphical model represents the number of independent parameters in the model. Each of these parameters is learned from individuals in the pool and hence can leak more information about membership in the pool. This leakage contributes to the power of the tracing attack. Hence, we confirm that the higher the complexity of the released model, the higher the power of the tracing attack.

We can clearly observe that the empirical and theoretical power in column 2 and 3 of Figure 1 are very close to each other except for the case of $\eta = 0$ on genome data. When $\eta = 0$, the model cannot capture any dependency in the data. If the released model does not capture all the dependencies among attributes in data (under fitted), then estimation errors of parameters in released graphical model become correlated. This effectively reduces the amount of information available to perform membership inference. Overall, we can see that the theoretical bound can capture the empirically observed power very effectively. **The observed power of tracing attack is close to the value calculated from the theoretical bound demonstrating the validity and usefulness of the bound.**

6 Related Work

Homer et al. (2008) developed a statistical test based on likelihood ratio for inferring the presence of a genome sequence, given the Allele frequencies. Sankararaman et al. (2009) extend this work and provide tight bounds on the power of tracing attack for any adversary. This was further extended for continuous Gaussian variables (Micro RNA) data in (Backes et al., 2016). Similar attacks on genomic data using statistics published in association studies are performed in (Shringarpure and Bustamante, 2015; Wang et al., 2009). Dwork et al. (2015) take a different approach and provide a generic framework for tracing attacks based on distance metric, when noisy statistics are released. Im et al. (2012), the authors use a correlation statistic to perform a tracing attack against regression coefficients from quantitative phenotypes. All these works assume independence among data attributes, whereas our work addresses the case of dependent attributes.

Shokri et al. (2017) perform membership inference attacks against black-box machine learning models. The adversary constructs *shadow models* that mimic the behavior of the target model. The attack is treated as a binary classification problem and the decision rule is a machine learning model trained on data from the shadow models. Salem et al. (2018) follow a similar framework as (Shokri et al., 2017) but relax certain assumptions on the knowledge and power of adversary. Similar attacks were performed against aggregate location data (Pyrgelis et al., 2017), generative adversarial networks (Hayes et al., 2018) and in a collaborative learning setting (Melis et al., 2018; Nasr et al., 2019). These works provide empirical analysis of tracing attack on complex models. A theoretical formulation of Bayes-optimal attack for membership inference against neural networks was given in (Sablayrolles et al., 2019),

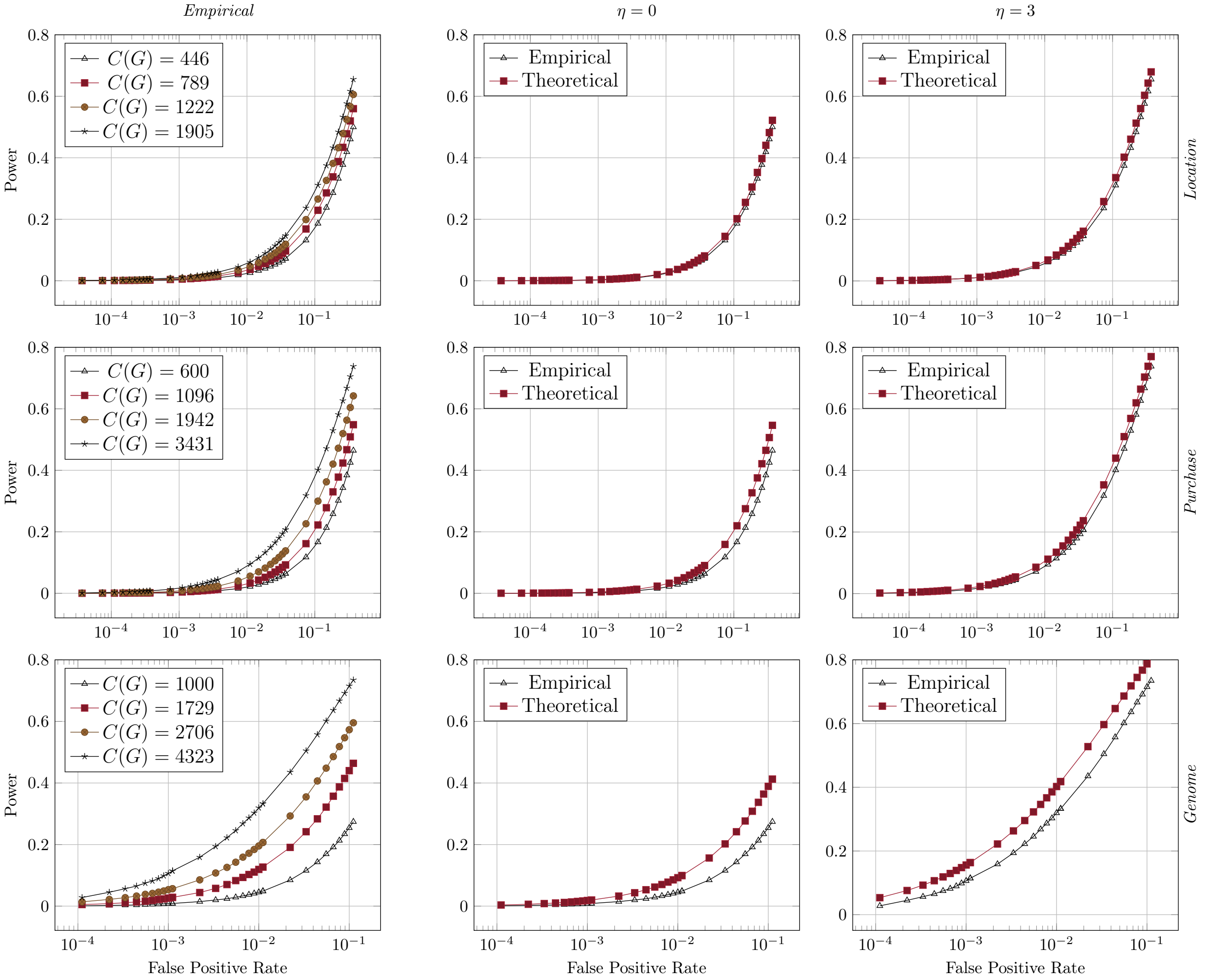

Figure 1: **Power of Attack on Real world data:** The effect of model complexity on the power of the attack is shown in the first column. When graphical models of increasing complexity are released, the power of attack increases, as we have more parameters that can leak information. In the second and third columns, we compare the observed powers with their corresponding theoretical bounds. We can clearly observe that the two curves (empirically observed power and theoretical bound) are very close to each other demonstrating the validity and usefulness of the bound.

which shows that existing techniques based on shadow models (Shokri et al., 2017) are approximations of this optimal attack. It is also shown that the power of optimal attack on black box models is the same as that on white box models, but no bound on this power is provided. We present a theoretical bound on the power of the attack, which is independent of the training data and the auxiliary knowledge of the adversary.

Differential Privacy (Dwork et al., 2006) has been accepted as the de facto standard notion of privacy. Zhang et al. (2017) learn a Bayesian network in a differentially private way and then use a noisy version of it to generate synthetic data. Bindschaedler et al. (2017) introduce the notion of plausible deniability and propose a mechanism that achieves it by generating synthetic data that is statistically similar to the given input data set. By definition, differential privacy decreases the power of tracing attack and its effect on our bound is discussed in Appendix H.

7 Summary

We provide a theoretical analysis of tracing attacks against probabilistic graphical models to address the existing gap between theoretical analysis for simple average statistics on data with independent attributes and empirical demonstrations for complex models on data with correlated attributes. Our bound quantifies the maximum attack performance measured with the error (false positive rate) and power (true positive rate) of a likelihood-ratio test. We experimentally validate and complement our results using sensitive datasets - location check-ins, purchase history, genomic data.

Acknowledgments

This work is partially supported by the Singapore Ministry of Education Academic Research Fund, R-252-000-660-133, the NUS Early Career Research Award (NUS ECRA), grant number NUS ECRA FY19 P16, and the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative.

References

Rupesh Agrahari, Amir Foroushani, T Roderick Docking, Linda Chang, Gerben Duns, Monika Hudoba, Aly Karsan, and Habil Zare. Applications of bayesian network models in predicting types of hematological malignancies. *Scientific reports*, 8(1):6951, 2018.

Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. Membership privacy in microrna-based studies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 319–330. ACM, 2016.

Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment*, 10(5):481–492, 2017.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 650–669. IEEE, 2015.

Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

J Hayes, L Melis, G Danezis, and E De Cristofaro. Logan: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, number 1. De Gruyter, 2018.

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

Hae Kyung Im, Eric R Gamazon, Dan L Nicolae, and Nancy J Cox. On sharing quantitative trait gwas results in an era of multiple-omics data and the limits of genomic privacy. *The American Journal of Human Genetics*, 90(4):591–598, 2012.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957a.

Edwin T Jaynes. Information theory and statistical mechanics. ii. *Physical review*, 108(2):171, 1957b.

Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. *arXiv preprint arXiv:1805.04049*, 2018.

M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 1022–1036, 2019. doi: 10.1109/SP.2019.00065. URL `doi.ieeecomputersociety.org/10.1109/SP.2019.00065`.

Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706): 289–337, 1933.

Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*, 2017.

Alexandre Sablayrolles, Matthijs Douze, Yann Ollivier, Cordelia Schmid, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. *arXiv preprint arXiv:1908.11229*, 2019.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965, 2009.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017.

Suyash S Shringarpure and Carlos D Bustamante. Privacy risks from genomic data-sharing beacons. *The*

American Journal of Human Genetics, 97(5):631–646, 2015.

Chengwei Su, Angeline Andrew, Margaret R Karagas, and Mark E Borsuk. Using bayesian networks to discover relations between genes, environment, and disease. *BioData mining*, 6(1):6, 2013.

Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 534–544. ACM, 2009.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):25, 2017.