In Differential Privacy, There is Truth: Evaluating PATE with Monte Carlo Adversaries

by

Jiaqi Wang

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science

Department of Electrical and Computer Engineering
University of Toronto

In Differential Privacy, There is Truth:
Evaluating PATE with Monte Carlo Adversaries

Jiaqi Wang
Master of Applied Science
Department of Electrical and Computer Engineering
University of Toronto
2024

Abstract

The shift from centralized to decentralized machine learning (ML) addresses privacy concerns associated with centralized data collection. A prominent approach for learning from decentralized data is the Private Aggregation of Teacher Ensembles, or PATE, which aggregates the predictions of a collection of teacher models. Aggregation is performed through a noised voting mechanism to reveal a collective prediction for the ensemble while providing differential privacy guarantees for the training data of each teacher model. PATE's differential privacy guarantees protect only against adversaries that observe a bounded number of predictions. PATE provides virtually no privacy guarantees in the realistic setting where an adversary is allowed to query the system continuously. However, the prospects of such an attack have never been evaluated.

We contribute to the first study on the confidentiality and privacy guarantees provided by PATE. We devise and implement an attack using Monte Carlo sampling to recover the votes submitted by participants of the PATE protocol, thus breaking PATE's confidentiality guarantees. Surprisingly, we also show that our adversary is more successful in recovering voting information when the vote-aggregation mechanism introduces noise with a larger variance. Because differential privacy generally benefits from noise with greater variance, this reveals a tension between achieving confidentiality and differential privacy in collaborative learning settings.

Next, we observe that PATE and its myriad variants assume that protocol participants, who contribute model votes, are honest. We evaluate scenarios where they can be corrupted by the attacker, and find that attacks become drastically more potent as the attacker is able to control more participants.

Robustly defending against the attacks reported in this paper is non-trivial, and is likely to result in a significantly reduced utility of PATE.

Acknowledgements

I would like to express my greatest gratitude to my supervisors, Professor Nicolas Papernot and Professor David Lie, for their insightful advice, patient guidance, motivating encouragement and financial support of this research work. I also sincerely appreciate Dr. Roei Schuster and Dr. Ilia Shumailov's assistance and feedback on this project. I would like to acknowledge the sponsors, who support the research with financial and in-kind contributions: Amazon, CIFAR through the Canada CIFAR AI Chair program, DARPA through the GARD program, Intel, Meta, Microsoft, NFRF through an Exploration grant, NSERC through the Discovery Grant, the Ontario Graduate Scholarship Program, a Tier 1 Canada Research Chair, the COHESA Strategic Alliance and University of Toronto. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. I also thank members of the CleverHans Lab for their feedback.

Contents

Chapter 1

Introduction

The canonical Private Aggregation of Teacher Ensembles, PATE, is a model-agnostic approach to obtaining differential privacy guarantees for the training data of ML models [27, 28], that is widely applied [19, 34] and adapted [8, 5, 31] due to its comparatively favorable trade-off between differential privacy, utility, and ease of decentralization [5]. In PATE, one considers an ensemble of independently trained teacher models. To generate a prediction, PATE first collects the predictions of these teachers to form a *histogram* of votes. It then adds Gaussian noise to the histogram and only reveals the label achieving plurality. This label can be used directly as a prediction, or to supervise the training of a student model—in a form of knowledge transfer. Because PATE only reveals the label receiving the most votes, it comes with guarantees of differential privacy, i.e., the noisy voting mechanism allows us to bound how much information from the training data is potentially exposed [1].

However, PATE does not explicitly protect itself from leakage of a key element in its inference procedure: the histogram of votes submitted by teachers. While the histogram is used internally and not directly exposed to clients, a careful examination of PATE reveals that information about the histogram leaks to clients via query answers.

The histogram can contain highly sensitive information, not the least of which is membership in minority groups, which, if revealed, can be used to discriminate against individuals. We demonstrate this by showing how an attacker, using the vote histogram of a PATE ensemble trained to predict an individual's income, can infer wholly different attributes, such as their level of education, even when the attacker's instance does not contain any information related to education level. Why is this possible? At a high level, the histogram of votes can be interpreted as a relatively rich representation of the instance, that reveals attributes beyond what the ensemble was designed to predict.

Next, we ask, is this attack a realistic threat? We answer this in the affirmative by designing an attack that extracts PATE histograms by repeatedly querying PATE, and showing that it reconstructs internal histograms to near perfection. Our attack builds on the fact that repeated executions of the same query produce the same internal histogram and a consistent distribution of PATE's noised answers corresponding to this histogram. Our adversary can thus sample this distribution many times via querying, and use it to reconstruct the histogram.

This implies that our attack relies on the stochasticity of PATE's output, which is a product of Gaussian noise, the very mechanism that was intended to protect privacy. In fact, we find that the larger the variance of noise added to the histogram votes, the more successful our adversary is in reconstructing the histogram.

This is in sharp contrast with the known and expected effect of differential privacy, which is that a higher noise scale generally leads to stronger privacy. Put simply, differential privacy makes our attack possible.

An astute reader may observe that histogram leakage does not violate the differential privacy guarantee, which only protects individual users in the training data, which is not compromised here. While it is absolutely true that our attack does not violate differential privacy, it clearly violates societal norms and user expectations that differential privacy is often incorrectly assumed to protect. The fact that differential privacy enables the leakage we exploit nicely underscores the distinction between technical definitions of privacy and common conceptions of privacy.

The attack is difficult to mitigate. Particularly, we show that it is stealthy in the sense that PATE's own accounting of "privacy cost" considers our attacker's set of queries "cheap", meaning that revealing their answers has a relatively small effect in terms of differential privacy. Consequently, PATE's privacy-spending monitoring does not prevent our attack. Our attack also performs only a moderate number of queries in absolute numbers, the same number used by common legitimate PATE clients, so a hard limit on queries would impede PATE's utility. We will discuss other mitigation approaches, which are not robust and/or not always usable.

1.1 Contributions

To summarize, our contributions are as follows:

- We posit the novel threat of extracting PATE's internal vote histograms. We observe and show that those contain sensitive information such as minority-group membership.
- We show that differential privacy is the cause for histogram-information leakage to PATE's querying clients.
- We exploit this leakage to reconstruct the vote histogram. We achieve this by minimizing the difference between (a) the probability distribution of outcomes observed by repeatedly querying PATE and (b) an analytical counterpart that we derive.
- We experiment with standard PATE benchmarks, showing that the attack can recover high-fidelity histograms while using a low number of queries that remain well within PATE's budget intended to control leakage.

1.2 Thesis Structure

This thesis is structured as follows. The background for understanding the attack method is given by Chapter 2. The target framework, PATE, is introduced in Chapter 3. To better understand the threat model, Chapter 4 explains why the object we attack is important. Then, in Chapter 5, we formally define the attack model and method. To demonstrate the efficacy of our attack, we perform multiple experiments in Chapter 6 and visualize the results. Then, in Chapter 7, we discuss the results and survey the related work for novelty. Finally, we state the open questions and future work in Chapter 8 and imply the broader impact in Chapter 9 to conclude the thesis.

Chapter 2

Background

This section introduces the background knowledge required to understand the attack model and algorithms. The attack model is based on PATE, a differential privacy-preserving decentralized framework. The core of the attack relies on the Monte Carlo Method and Gradient Descent.

2.1 Differential Privacy

An algorithm is considered differentially private if its outputs on adjacent inputs (in our case, datasets) are statistically indistinguishable. Informally, the framework of differential privacy requires that the probabilities of an algorithm make specific outputs indistinguishable on two adjacent input datasets. Two datasets are said to be adjacent if they only differ by, at most, one training record. The degree of indistinguishability is bounded by a parameter denoted ε. The lower ε is, the stronger the privacy guarantee is for the algorithm. It is harder for an adversary to distinguish adjacent datasets, given access to the algorithm's predictions on these datasets. In the variant of differential privacy we use, we can also tolerate that the guarantee does not hold with probability δ. This allows us to achieve higher utility. The formal definition of differential privacy is given by the following formulation:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr\left[\mathcal{M}\left(D'\right) \in S\right] + \delta,$$

where M is a randomized algorithm which takes input from domain $\mathcal{D}$ and output to a range $\mathcal{R}$, $D, D' \in \mathcal{D}$ are two adjacent inputs and $S \subseteq \mathcal{R}$ is any subset of outputs.

2.2 Probability Theory

2.2.1 Conditional Probability and Independence

Two events are considered independent if the occurrence of one has no impact on the probability of the other's occurrence. If two events, A and B, are independent, then $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$, where $P(A, B)$ denotes the probability of A and B coincide, and $P(A|B)$ denotes the probability of A occurs given the fact of B occurs.

2.2.2 Law of Large Numbers

The law of large numbers (LLN) states that the true value of a variable could be approximated by taking the average of a large number of independent random samples. More specifically, the LLN describes that the sample mean converges to the true mean if a large number of values are sampled independently and identically distributed.

2.2.3 Monte Carlo Method

Monte Carlo methods (MCMs) are a wide range of computational algorithms that use random sampling to estimate deterministic results. A Monte Carlo sample stimulates a variable with uncertainty by repeatedly assigning it a random value. After repeating the sampling again and again, different values are assigned to the variable. By LLN, instead of integral, the expected value of the variable can be approximated by taking the average of these independent sample values. The precision of the MCM is characterized by its confidence interval. The narrower the confidence interval, the more precise the estimate is. The 95% confidence interval is $\bar{x} \pm 1.96\frac{\sigma}{\sqrt{M}}$, where $\bar{x}$ is the estimated value of a random variable, σ is the standard deviation of the random variable and M is the sample size. So, to achieve a high-precision estimate, MCMs need a large sample size when the standard deviation of the random variable is large.

2.3 Gradient Descent

Our attack relies on optimization to recover vote histograms. In our case, we use gradient descent as the optimizer, an iterative algorithm to find a function's local minimum. The idea is to take a step in the opposite direction of the gradient because that is the direction that leads to the largest decrease in the function's value.

The concept of a gradient can be visualized as the slope of a curve for a univariate function. For a function with multiple variables, such as n-dimensional function $f(\vec{x})$, the gradient at a specific point $\vec{x_0}$ is a vector as follows.

$$\nabla f(x_1, x_2, \ldots, x_n) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x_1, x_2, \ldots, x_n) \\ \frac{\partial f}{\partial x_2}(x_1, x_2, \ldots, x_n) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

This vector can be thought of as the slope of a multi-dimensional surface. Just like in the univariate case, at the local minimum, the slope of the curve at that point is zero. This means that the gradient vector is $\vec{0}$ at the point where the function $f(\vec{x})$ reaches its minimum value.

In summary, the Gradient Descent method takes the following steps.

- Define the problem, i.e., find the objective to minimize, which should be in a function form and differentiable to the random variable.
- Initialization, i.e., choose a value for xv as the start point.
- Take a step opposite the direction of the gradient. The general formulation of the gradient descent to find the local minimum of the function $F(\vec{x})$, at the n^{th} iteration, is $\vec{x}_{n+1} \leftarrow \vec{x_n} - \lambda \nabla F(\vec{x_n})$.
- Loop over steps two and three until the step size is smaller than the threshold

Chapter 3

Private Aggregation of Teacher Ensembles (PATE)

3.1 Primer

The PATE framework begins by independently training an ensemble of models, called teachers, on partitions of the private data. There is no particular requirement for the training procedure of each of these teacher models; the only constraint is that the partitions be disjoint.

Queries made by clients are answered as follows: (1) each teacher model predicts a label on the instance, (2) the PATE aggregator builds a histogram of class votes submitted by teachers, (3) Gaussian noise is added to this histogram, and (4) the client receives the noised histogram's argmax class (henceforth *result class*). This noisy voting mechanism gives PATE its differential privacy (DP) guarantee, in what is an application of the Gaussian mechanism [26].

To preserve differential privacy, PATE tracks the *privacy cost* of the set of past queries, and stops answering queries once the cost surpasses the *privacy budget*. The cost computation is parameterized by a size δ. The key differential privacy guarantee of PATE can be stated as follows: for a given set of queries with cost ε, PATE is (ε, δ) differentially private. Put succinctly, ε bounds an adversary's ability to distinguish between any adjacent training datasets, whereas δ bounds the (usually small) probability, over PATE's randomness, of this bound not holding.

3.2 The PATE framework

To train a PATE framework, first, each data owner trains a local *teacher* model using its private dataset. Then, teachers are aggregated to form an ensemble whose predictions are guaranteed not to leak those of individual models. Next, the teacher ensemble's predictions are used to label unlabelled public data. Finally, the now-labeled data is used to train the *student* model. The resulting student model can be shared or queried and, due to the aggregator's guarantee, this is, in turn, guaranteed not to leak information on any participant's private data. The training process is demonstrated by figure 3.1.

Recently, fully decentralized settings have been proposed where the central aggregator is replaced with a cryptographic multi-party computation protocol [5], thus providing "end-to-end" privacy guarantees. There

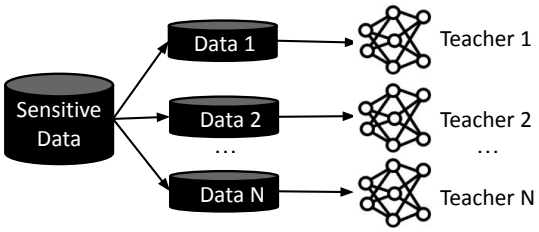


Figure 3.1: In the training phase, sensitive data is divided into N disjoint subsets, and each subset of data is trained into a teacher model.

are several different implementations of PATE, but they default to using the Gaussian Mechanism—because it is best suited for the analysis of PATE in the framework of Rényi Differential Privacy [22].

3.3 Differential Privacy Guarantees for PATE

Throughout PATE's inference-time operation, it is possible — given the aggregator's knowledge, including all past queries' vote histograms and a constant δ and noise variance σ — to compute ϵ such that a mechanism that answers these queries by following the PATE protocol is (ϵ, δ)-differentially private. (See section 2.1 for background on differential privacy and what this means.) We henceforth refer to ϵ as the *privacy cost* of a series of queries. Given a *privacy budget* ϵ_{max}, PATE forms a (ϵ_{max}, δ)-differentially private mechanism by ceasing to answer when the cost approaches the budget, i.e., enforcing that $\epsilon < \epsilon_{max}$. To avoid dynamically changing guarantees, PATE sometimes operates with a *privacy budget* ϵ_{max}. It stops answering once ϵ approaches the budget, thus forming a (ϵ_{max}, δ)-differentially private mechanism regardless of queries.

3.4 PATE Inference

Clients can query the ensemble to perform inference [5], or use it to label their own data (or public data) and train their own models. [28, 27]. The inference process is described by figure 3.2 on a high level.

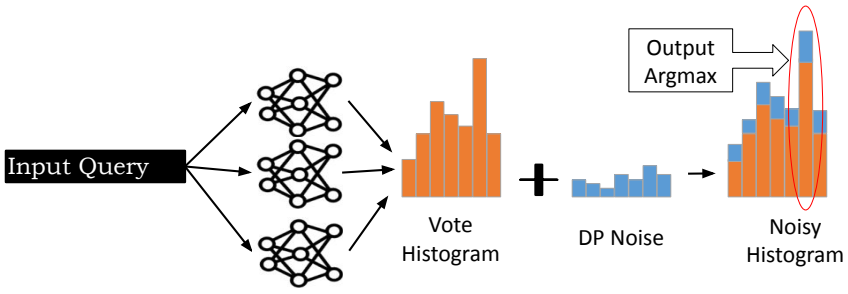


Figure 3.2: In the inference phase, the input query is sent to each teacher who makes a prediction independently. The predictions are aggregated, and a random noise for the DP guarantee is added. Finally, the class with the maximum number is returned.

Let a be the input (i.e., an image, a medical record, etc.) used for a $c-$class classification inference. Let N denote the number of teachers, and m the number of labels. Each teacher $i \in [N]$ [1] owns a program P_i, in this case, a classifier that is trained with its private data. Each teacher uses their model to evaluate and output a vote for a. For example, $P_i(a) = x_i$ is the vote of teacher i. Each teacher only knows their own vote while not knowing the others' votes.

Let O be the trained PATE which takes as input a set of teachers and the inference sample a, and outputs the classification result of the given teachers on the sample, $O(\{1, 2, \ldots, N\}, a) = \{P_1(a), P_2(a), \ldots, P_N(a)\} = \{x_1, x_2, \ldots, x_N\}$. Then, for all N teachers, the generated votes $X = \{x_1, x_2, \ldots, x_N\}$ is used as input to the joint computation function. For PATE, the joint computation function is a composition of two functions: the first is *Count*, which takes all the votes and outputs their value histogram, and the second is an aggregation (*Agg*), which adds Gaussian noise to each label's vote count and outputs the result class. Formally, *Count* is defined as follows:

$$Count : \mathbb{Z}^N \mapsto \mathbb{Z}^c$$

where the output is a histogram of vote values *i.e.,* $Count(X) = H = [h_1, h_2, \ldots, h_c]$ where h_i is the number of votes for class i.

Formally, *Agg* is defined as follows:

$$Agg : \mathbb{Z}^m \mapsto \mathbb{Z}$$

Given a zero mean normal distribution with variance σ^2 (representative of the distribution to sample noise from), the Gaussian mechanism for noisy vote aggregation returns:

$$Agg(H; \sigma) = \underset{j}{\operatorname{argmax}}\{h_i + \mathcal{N}(0, \sigma^2)\}, \quad \forall i \in [m]$$

The whole PATE protocol can be described as a function

$$F = O([N], a) \circ Count \circ Agg$$

[1] $[N] = \{1, \ldots, N\}$

Chapter 4

Vote Histograms are Sensitive Information

We consider the leakage of an ensemble's vote histogram, such as those computed internally in PATE. Clearly, such histograms contain a lot more information on PATE's inner workings than simply its revealed decision, but it is important to clarify that there are common contexts in which this leakage can actually be used to hurt individuals as they contain sensitive information about them.

Differential privacy addresses the contribution of an individual, while minority-group membership, in the context of machine learning, is the group of data with a certain feature that makes up a smaller proportion of the population compared to the group of data without. The definition of minority-group membership could be considered as a group-differential-privacy, which is not covered by the $(\epsilon, \delta)-$differential privacy. When the classification task is asked to exclude the feature that is imbalanced distributed in the population, the minority-group membership becomes sensitive information; moreover, a skewed feature proportion is usually sensitive by default; for example, the disease diagnosis in a medical dataset is a minority group but is sensitive information for the patients.

As a prominent example, minority-group membership often leaks via histograms and can, of course, be used to discriminate against group members. To understand this, let's consider a minority group that is under-represented in the training data distributed across PATE's teachers. Each teacher observes some outliers and misrepresentative phenomena, such as coincidental correlations or out-of-distribution examples. When data on group members is scarce, each model will tend to over-fit to the outlier phenomena within its own data, creating inter-model inconsistencies and resulting in disagreement, or low consensus, when predicting similar inputs at test time—which readily presents itself on vote histograms. Thus, *we expect histograms to reveal members of minority group members via low consensus values.* Next, we illustrate this via a simple experiment.

4.1 Extracting Sensitive Attributes from UCI Adult-Income Histograms

We now simulate an attack that receives the vote histogram of a salary-predictor ensemble and uses it to detect a small minority of the population, specifically, PhD holders. Following the above observation, our

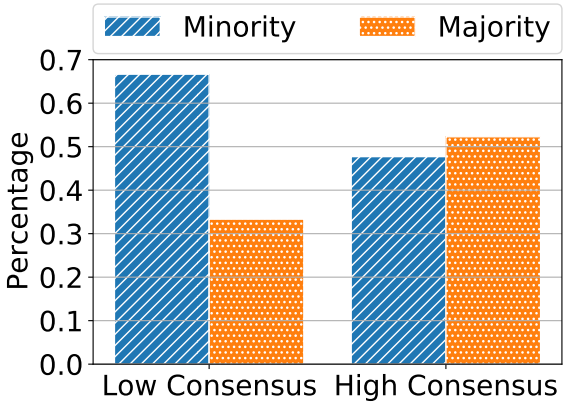


Figure 4.1: High vs. low-consensus distributions of the PhD-detection attack: vote histograms of minority-group members present lower consensus, allowing an attacker to identify them.

attack will simply classify all highly-consensus (consensus > 75%) predictions as non-PhD-holders, whereas low-consensus (< 75%) predictions will be classified as PhD holders. This is a heuristic attack that relies on intuition rather than learning the ensemble's behavior using a labeled dataset. On the one hand, it may underestimate the attacker's ability to detect PhD holders; on the other hand, it does not require a labeled dataset and only assumes that the attacker sees the vote histogram.

We use the UCI Adult-Income dataset [7], containing around 41,000 data points with basic personal information on people such as age, work hours, weight, education, marital status, and more. PhD holders form about 1% of this dataset. We randomly selected 80% of the dataset for training, and held out the rest for testing. We randomly partitioned the data into 250 disjoint sets. For each, we fitted a random forest model predicting whether income is above or below $50,000. For both training and test data, we removed the data columns explicitly indicating education levels; that is, training and test individuals do not contain any feature that directly distinguishes PhD from non-PhD holders.

Fittig Random Forests Every one of our teachers in Section 4 fits a random forest classifier using the sklearn package; each teacher performed a grid search over the following hyperparameters, and picked the values that lead to the lowest training loss.

- max_depth: the maximum number of levels that a tree has, an integer is chosen between 1 and 11 inclusively;
- max_features: the maximum number of features, while splitting a node, one of sqrt(number of features), log(number of features), 0.1*(number of features), 0.2*(number of features), 0.3*(number of features), 0.4*(number of features), 0.5*(number of features), 0.6*(number of features), 0.7*(number of features), 0.8*(number of features), 0.9*(number of features);
- n_estimators: the number of trees that the forest has, an integer chosen between log(9.5) and log(300.5);
- criterion: the loss function, one of gini impurity and entropy;
- min_samples_split: the minimum number of instances for a node to split, one of 2, 5, 10;

- bootstrap: one of True or False

Figure 4.1 shows the distribution of high-consensus and low-consensus on the test set (to clarify the effect, we balanced the minority and majority groups in the test set by randomly removing most of the non-PhD samples). We observe that low consensus indeed indicates minority-group membership. Our attacker's precision is not particularly high (75% on the balanced set), but they can still use this signal to discriminate against minority groups.

Chapter 5

How to Extract PATE Histograms

Having established that vote histogram leakage poses a risk to privacy and fairness, we proceed to provide a generic method for extracting vote histograms from PATE.

5.1 Problem Formulation and Attack Model

This section discusses the attack model by defining its goal and resources. We also define what we mean by corruption, a threat model where we assume that not all the teachers are honest but curious.

Attacker's motivation. Our attacker's goal is to recover the histogram of the vote counts when PATE labels an instance. Formally, given N predictors $\{P_1, ..., P_N\}$ and a target input a, our attacker wants to infer $H \equiv Count(P_1(a), \ldots, P_N(a)) = [h_1, \ldots, h_c]$ where $Count$ counts the number of appearances of each element in $[c]$. Vote histograms can be used to extract potentially sensitive information about an instance, such as its race, gender, or religion (see Section 4).

Attacker's access and knowledge. Our attacker can send queries to the aggregator and receive the label predicted by PATE (i.e., the output of the noisy voting mechanism). This may be possible because the aggregator willfully exposes the predictions of PATE, e.g., through an MLaaS API. Alternatively, fully decentralized implementations of PATE have been proposed where the central aggregator is replaced with a cryptographic multi-party computation protocol [5], and its output is exposed directly. Figure 5.1 visualizes the workflow of our attack.

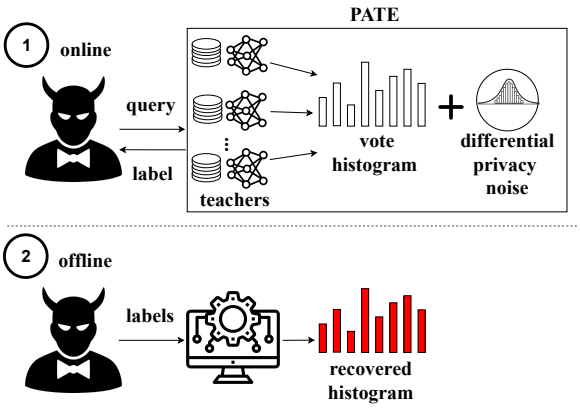


Figure 5.1: In an online phase, the attacker repeatedly sends a specific query to PATE and receives labels output by the noisy argmax. Offline, the attacker uses the labels to recover the histogram by constructing and solving an optimization problem.

Algorithm 1 Attack pseudocode

Input:

$N \in \mathbb{N}$ ▷ total number of teachers (see Section 5.1 for why this is needed)

$\mathbb{O}$ ▷ PATE instance

T, λ ▷ optimization termination threshold and learning rate

Output: $\hat{H}$

$S \leftarrow sample(\mathbb{O}, M)$ ▷ sampling PATE M times and storing into $S \in 1, ..., K^M$

for $i = 1, 2, \ldots, M$ **do**

  for $j = 1, 2, \ldots, c$ **do**

    $q_j^i = int(S^i == j)$ ▷ $q_j^i = 1$ if $S^i = j$, 0 otherwise

  end for

end for

$\bar{q} \leftarrow 0^c$ ▷ initialization of $\bar{q}$ with a 0 vector

for $i = 1, 2, \ldots, M$ **do**

  $\bar{q}[i] \leftarrow \frac{1}{M} \sum_{j=1}^{M} q_i^j$

end for

$\hat{H} \leftarrow 0^c$ ▷ initialization of $\hat{H}$, here we use an all-zero array of length c

while $\|Q(\hat{H}) - \bar{q}\|_2 > T$ **do**

  $\hat{H} \leftarrow \hat{H} - \lambda \nabla_{\hat{H}} \|Q(\hat{H}) - \bar{q}\|_2$

end while

$\hat{H} \leftarrow \hat{H} + [\frac{N - \sum \hat{H}}{c}]^c$ ▷ shift $\hat{H}$ to sum to N

return $\hat{H}$

In PATE, the parameters (mean, variance) of the noise added during aggregation are public domain [28]; we, therefore, assume the attacker knows them. We also assume the attacker knows the number of teacher models N, which may or may not be public. This assumption is only necessary to shift the attacker's learned distribution by a constant to attain a low L_1 approximation error when reconstructing histograms (Section 5). We note that the attacker could just as easily exploit the leakage (e.g. to learn sensitive attributes or differentiate between training sets) without it but we chose to instead make this assumption to simplify

result presentation and interpretation.[1]

5.2 Histogram Reconstruction Attack

The idea behind the attack, given in pseudo-code in Algorithm 1, is as follows: let Q be a function that computes output-class probability distribution of PATE given a vote histogram H. First, our attacker will sample PATE to find an estimate for this distribution $\bar{q} \approx Q(H)$. Second, the attacker will use gradient descent to find $\hat{H}$ that minimizes the Euclidean distance between $Q(\hat{H})$ and $\bar{q}$. Finally, they shift the estimated histogram $\hat{H}$ by a constant to account for the number of teachers (this step assumes the number of teachers is known, but is done mostly for presentation purposes, see Section 5.1). We now detail these 3 steps.

Step 1: Monte Carlo approximation. The first step will sample PATE M times and estimate the distribution over PATE's outputs $\bar{q} \approx Q(H)$ by setting each class probability as its Monte Carlo estimated mean frequency, i.e. $\bar{q}_i \leftarrow \frac{1}{M}\sum_{j=1}^{M} q_i^j$ where q_i^j indicates whether class i was sampled in the jth step. By the law of large numbers, as M increases, $\bar{q}_i$ converges to i's sampling probability $Q(H)_i$, and we can expect the attacker's estimate produced in the next steps to be more accurate. Our attacker would want to increase M as much as possible, until they exceed PATE's privacy budget.

Indeed, in PATE, the privacy leakage expended by each individual query can then be composed over multiple queries to obtain the total privacy cost ε needed to answer the set of queries. Once the total privacy cost ε exceeds a maximum tolerable privacy budget, PATE must stop answering queries to preserve differential privacy. Section 6 shows that the attack succeeds for values of M that remain well below PATE's privacy budget, and are also moderate in absolute value, as they are similar to the query number of student models that use PATE.

Step 2: constructing the optimization objective. Our attacker wants to find $\hat{H}$ such that $Q(\hat{H}) - \bar{q}_2$ is minimized where $\cdot_2$ denotes the Euclidean norm. Given a (differentiable) closed-form expression for Q, it becomes natural to program and solve this with modern gradient-based optimization frameworks. Theorem 1 provides a closed-form expression, and our attacker will use a differentiable approximation of this expression, as explained below.

Theorem 1. *Let $H = [H_1, \ldots, H_c]$ be the vote histogram for the c classes, and let PATE's Gaussian-mechanism function $Agg(H) \equiv \text{argmax}\{H_i + \mathcal{S}_i\}$ where $\mathcal{S} = [\mathcal{S}_1, \ldots, \mathcal{S}_M]$ is a vector of M samples from a zero-mean normal distribution with variance σ^2. Then the probability that the randomized aggregator outputs the class k is given by $Q(H)_k = \mathbb{P}(Agg(H) = k) = \int_{-\infty}^{\infty} \prod_{i=1}^{c, i\neq k} \Phi_i(\alpha)\phi_k(\alpha)\mathrm{d}\alpha$ where $\Phi_i(\cdot)$ is the cumulative probability distribution (CDF) of $\mathcal{N}(H_i, \sigma^2)$ (normal distribution with mean H_i and variance σ) and $\phi_k(\cdot)$ is the probability density function (PDF) of $\mathcal{N}(H_k, \sigma^2)$.*

Proof. $Q(H) = \mathbb{P}(Agg(H) = k)$, is the probability that $H_k + S_k = \max\{Agg(H)\}$. For any k, $H_k + \mathcal{S}_k$ is a random variable that follows a normal distribution with mean equal to H_k and variance equal to σ^2. Let $g_k = H_k + \mathcal{S}_k$,

[1] Indeed, we could avoid this assumption while still retaining low error if we measured the attacker's error with shift-invariant distances, like the Pearson correlation.

then $g_k \sim \mathcal{N}(H_k, \sigma^2)$. $Q(H)_k$ is the probability of g_k is greater than $g_j, \forall j \in \{1, \dots k-1, k+1, \dots c\}$

$$\begin{aligned} Q(H)_k &= \mathbb{P}(Agg(H) = k) \\ &= \mathbb{P}(g_k > g_1, \dots, g_k > g_{k-1}, g_k > g_{k+1}, \dots, g_k > g_c) \\ &= \int_{-\infty}^{\infty} \prod_{i=1}^{c, i\neq k} \mathbb{P}\left(g_i < \alpha \mid g_k = \alpha\right) \mathbb{P}(g_k = \alpha) \mathrm{d}\alpha \\ &= \int_{-\infty}^{\infty} \prod_{i=1}^{c, i\neq k} \Phi_i(\alpha)\phi_k(\alpha)\mathrm{d}\alpha \end{aligned}$$

□

The expression in Theorem 1 is not usable in automatic differentiation and optimization frameworks; we, therefore, use an approximation of the integral by the trapezoid formula. We select points with higher probability and sum up their values to get an approximation of the integral with infinite bounds. Then we decide what values to select. In the integral $\int_{-\infty}^{\infty} \prod_{i=1}^{m, i\neq k} \Phi_i(\alpha)\phi_k(\alpha)\mathrm{d}\alpha$, α is the value of $g_k \sim \mathcal{N}(H_k, \sigma^2)$. Therefore α has the highest probability at H_k, and has the higher probability closer to H_k. More specifically, properties of the normal distribution give us that $\mu \pm 6 * \sigma$ covers 99% of the values of Gaussian random variable $z \sim \mathcal{N}(\mu, \sigma)$. Therefore, values of α between $H_k \pm 6 * \sigma$ cover 99% of the integral area. Therefore,

$$\int_{-\infty}^{\infty} \prod_{i=1}^{c, i\neq k} \Phi_i(\alpha)\phi_k(\alpha)\mathrm{d}\alpha \approx \sum_{H_k - 6\sigma}^{H_k + 6\sigma} \prod_{i=1}^{c, i\neq k} \Phi_i(\alpha)\phi_k(\alpha),$$

which is differentiable and is handled well by most automatic differentiation packages.

Step 3: accounting for the number of teachers. The distribution estimate produced by our optimization may be skewed by a constant because $Q(H)_k$ only depends on the differences between g_k and $g_1, \dots, g_{k-1}, g_{k+1}, \dots, g_c$, so the attacker shifts each element of $\hat{H}$ by $(N - \sum \hat{H})/c$ so that the new histogram $\hat{H}$ sums up to $\sum \hat{H} + c * (N - \sum \hat{H})/c = N$.

The following theorem shows that the number of teachers does not affect the attacker's computation.

Theorem 2. *For two histograms, $H^1 = [h_1^1, \dots, h_m^1]$ and $H^2 = [h_1^2, \dots, h_m^2]$, $Q^{H^1,\sigma} = Q^{H^2,\sigma}$ if $h_i^1 - h_i^2 = h_j^1 - h_j^2$ for all $i, j = 1, \dots, m$.*

Proof. let $d = h_i^1 - h_i^2 = h_j^1 - h_j^2$ for all $i, j = 1, \dots, m$.

$$\begin{aligned} \mathbb{P}(g_i^2 > g_j^2) &\sim \mathcal{N}((h_i^2 - h_j^2), 2\sigma^2) \\ &= \mathcal{N}((h_i^1 + d) - (h_j^1 + d), 2\sigma^2) \\ &= \mathcal{N}(h_i^1 - h_j^2, 2\sigma^2) \\ &= \mathbb{P}(g_i^1 > g_j^1) \end{aligned}$$

for all $i, j = 1, \dots, m$.

$$
\begin{aligned}
Q_k^{H^1,\sigma} &= \mathbb{P}([g_k^1 > g_1^1, \ldots, g_k^1 > g_{k-1}^1, \\
&\quad g_k^1 > g_{k+1}^1, \ldots, g_k^1 > g_m^1]) \\
&= \mathbb{P}([g_k^2 > g_1^2, \ldots, g_k^2 > g_{k-1}^2, \\
&\quad g_k^2 > g_{k+1}^2, \ldots, g_k^2 > g_m^2]) \\
&= Q_k^{H^2,\sigma}
\end{aligned}
$$

□

What Theorem 2 states is, if the difference between two histograms is uniform, then the probability distribution of the outcomes is the same. With the support of Theorem 2, H can be safely shifted by a constant amount to sum up to the number of teachers, N.

Chapter 6

Evaluation

We evaluate our attack against instantiations of PATE on common benchmarks. We show that the extracted histograms only differ slightly from the true ones underlying PATE's decision. This is despite the low privacy cost of the attacker's queries, which remains well within budgets enforced by common PATE instantiations. We also quantify the impact of the choice of scale for the noise being added to preserve DP: we show that *higher noise values result in increased attack success* for a given number of queries. We offer a hypothesis to explain this ostensibly surprising observation.

6.1 Experimental Setup

Data. We use the experimental results from Papernot et al. [27] to simulate our attack environment. Papernot et al. released the histograms obtained by PATE using 250 teachers for two 10-class computer-vision benchmarks, MNIST [18] and SVHN [25]. There are 9,000 histograms generated by MNIST experiments and 26,032 histograms generated by SVHN experiments, corresponding to the sizes of these datasets' test sets.

We define a histogram's *consensus* as its maximum value, and divide each dataset into three equal-sized groups corresponding to high consensus, medium consensus, and low consensus. Figure 6.1 illustrates this. We sample five histograms randomly from each group, and mount our attack for various noise levels.

Chosen histograms for evaluation. Table 6.1 shows the histograms we chose for evaluation in the 3 consensus-level categories.

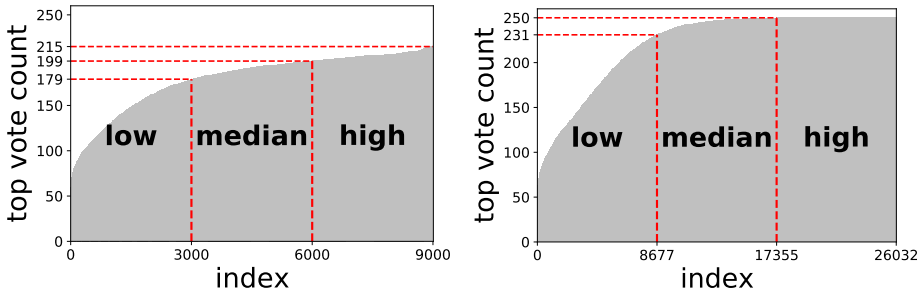


Figure 6.1: Divisions of the 9,000 and 26,032 histograms of MNIST (left) and SVHN (right) datasets into 3 consensus levels, measured by top-agreed label percentage. The dashed red lines delineate the 33.3% and 66.7% quantiles.

	MNIST	SVHN
High consensus		
H1	[4, 7, 6, 8, 4, 2, 0, 214, 4, 1]	[0, 0, 0, 0, 250, 0, 0, 0, 0, 0]
H2	[4, 7, 207, 10, 4, 4, 0, 10, 3, 1]	[0, 0, 250, 0, 0, 0, 0, 0, 0, 0]
H3	[5, 205, 7, 8, 4, 3, 0, 11, 6, 1]	[0, 0, 0, 250, 0, 0, 0, 0, 0, 0]
H4	[4, 7, 6, 7, 4, 200, 4, 10, 7, 1]	[0, 250, 0, 0, 0, 0, 0, 0, 0, 0]
H5	[4, 7, 210, 7, 4, 4, 0, 10, 3, 1]	[0, 0, 0, 0, 0, 0, 250, 0, 0, 0]
Median consensus		
H1	[5, 183, 9, 16, 4, 3, 1, 10, 17, 2]	[0, 0, 1, 0, 249, 0, 0, 0, 0, 0]
H2	[6, 7, 6, 30, 4, 181, 0, 10, 5, 1]	[0, 10, 1, 232, 1, 3, 0, 1, 0, 2]
H3	[4, 7, 6, 10, 13, 4, 0, 17, 3, 186]	[0, 0, 0, 6, 0, 243, 0, 0, 0, 1]
H4	[6, 18, 184, 7, 10, 4, 7, 10, 3, 1]	[236, 0, 0, 7, 0, 0, 6, 0, 1, 0]
H5	[7, 7, 8, 7, 4, 9, 193, 10, 4, 1]	[234, 2, 0, 4, 0, 0, 0, 1, 9, 0]
Low consensus		
H1	[12, 7, 6, 30, 4, 161, 0, 10, 19, 1]	[1, 1, 20, 12, 0, 0, 2, 207, 7, 0]
H2	[4, 8, 7, 11, 38, 16, 1, 13, 8, 144]	[0, 158, 1, 6, 4, 38, 0, 40, 1, 2]
H3	[4, 7, 15, 33, 6, 5, 0, 171, 5, 4]	[0, 184, 0, 2, 3, 0, 0, 61, 0, 0]
H4	[4, 7, 117, 99, 4, 4, 0, 10, 4, 1]	[0, 0, 24, 0, 0, 0, 0, 0, 0, 226]
H5	[4, 17, 6, 11, 154, 4, 0, 11, 5, 38]	[10, 1, 2, 19, 7, 109, 73, 0, 19, 10]

Table 6.1: The 30 MNIST and SVHN vote histograms sampled from the collection of histograms provided by Papernot et al. [27] (divided into 3 equally-sized consensus groups). We refer to histograms denoted here by H1-5 in the different consensus groups throughout the presentation of our results.

Attack parameterization. We simulated attackers with two types of query limits: first, an attacker limited by PATE's canonical privacy budget; we used the parameterization from Papernot et al. [27], i.e., budgets of 1.97 and 4.96 for MNIST and SVHN and $\sigma = 40$. Second, an attacker with a hard limit of 10^4 queries; this is a moderate number of queries for clients wishing to train their own "student" model using the aggregator's labels (see [27, 28]). We applied this attack against PATE instantiations for MNIST and SVHN with noise levels $\sigma \in 40, 60, 80, 100$.

For optimization (see Section 5), we use an adaptive learning rate: at the beginning of training, we use a learning rate of $\frac{10}{\nabla_{\hat{H}} J_2}$, where $J = Q(\hat{H}) - \tilde{q}$ is the optimization objective. As the optimization starts to converge, $\frac{10}{\nabla_{\hat{H}} J_2}$becomes too large, so we switch to a learning rate of $\frac{1}{\nabla_{\hat{H}} J_2}$. This results in changes to the histogram of the magnitude of one vote for each update. We use 0.01 as a threshold on the loss to establish convergence, and thus, when $\|J\|_2 < 0.01$, we stop optimizing. For the attacks against canonical settings, we stopped once estimated histograms started presenting negative values, which we found to be a slightly better strategy. (We could also try to constrain it to only-positive values; we discuss improving this optimization procedure further in Section 7).

Metrics. For every attack, we measured the *error rate* and *privacy cost*. The error rate is defined as the normalized L_1 distance between the ground-truth histogram $H = [H_1, \ldots, H_c]$ and our attacker's estimate $\hat{H} = [\hat{H}_1, \ldots, \hat{H}_c]$, i.e., $\sum_i \left|H_i - \hat{H}_i\right| / 2 \sum_i |H_i|$. (While the optimization minimizes Euclidean distance, we report L1 errors because they can be interpreted as corresponding to the number of miscounted votes.)

We define and compute the privacy cost incurred by the adversary using established practices. At a high level (see details in [27]), we model PATE as a Rényi-differentially-private mechanism and leverage known privacy-preserving-composition theorems; we attain (non-Rényi) differential privacy via a known reduction from differential privacy to Rényi differential privacy.

The parameter δ is set as 10^{-5} for MNIST and 10^{-6} for SVHN, following Papernot et al. [27].

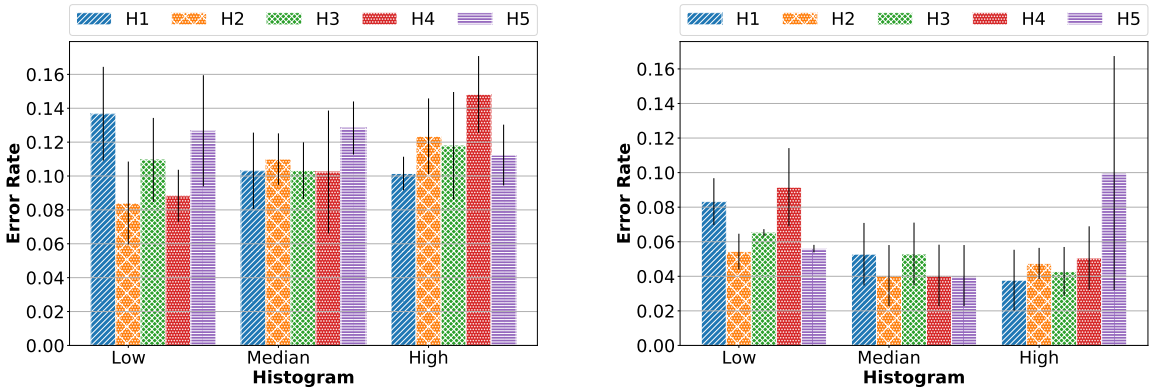
Implementation. Our implementation is provided in Python, and the optimization uses the Jax library. Our code is open-sourced at `https://github.com/cleverhans-lab/monte-carlo-adv`. We ran the optimization on an Intel Xeon Processor E5-2630 v4; it takes about 2.5 hours to complete for a single histogram.

6.2 Results

Our attack has high performance within canonical privacy budgets. We first evaluate our attack on canonical PATE from Papernot et al. [27]. Figure 6.2a and 6.2b show our attacker's error rates for the different histograms, averaging 0.11 on the MNIST setup and 0.05 on the SVHN setup.

Our attack extracts high-fidelity histograms and has low privacy costs. Figure 6.2 reports the performance of the privacy-budget limited attack; Figures 6.4 and 6.5 show our hard-query-limit attacker's error rate and query costs for different noise levels, i.e. values of σ. We observe that, across attacks, we attain very low error rates, often as low as 0.03, translating to 3% of the votes being miscounted. For the hard-query-limit attack, privacy costs roughly range between 1 to 12, which is the order of magnitude for the budget one would plausibly use, for example to attain guarantees similar to Papernot et al. [28] (which uses budgets of up to 8 in a directly comparable setting to ours) or Abadi et al. [2] (which also employs a $(8, 10^{-5})$-differentially private mechanism for MNIST).

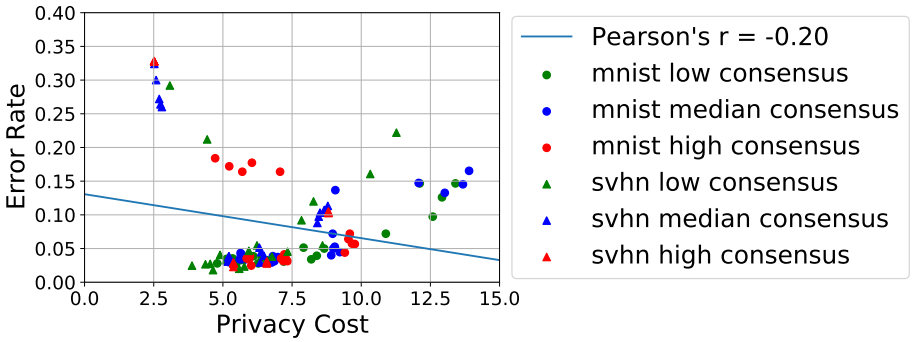
Adding more noise helps the attacker. Perhaps the most surprising result of this work is that the higher the noise scale, the lower the attacker's error is. This is *not* necessarily aligned with using up more of the privacy budget. In fact, in many cases, increasing the noise decreases both the attacker's privacy cost and their error; Figure 6.3 shows the correlation between cost and error.



(a) Error rates on attacking a canonical MNIST PATE with privacy budget = 1.97 and $\sigma = 40$

(b) Error rates on attacking a canonical SVHN PATE with privacy budget = 4.96 and $\sigma = 40$

Figure 6.2: Error rates on the budget-limited attack on the canonical PATE [27], for our 15 low/median/high-consensus sample histograms.



R0.9

Figure 6.3: Our attack's average privacy cost vs. error rate on the histograms extracted with 10^4 queries. A weak inverse correlation implies cheaper attacks are often more accurate.

This is counter-intuitive, as larger Gaussian scales σ usually correspond to tighter privacy guarantees. That is, more protection is expected against attacks. Specifically, our Monte Carlo estimation should be less accurate when higher-variance noise is added, as convergence to the mean is slower. Nevertheless, our attack actually performs *better* with higher noise levels.

To explain this, consider the aggregator's output distribution. When it is uniform, classes are sampled with equal probabilities, contributing equal information to each Monte Carlo estimator. Conversely, when some classes have a lower probability than others, their estimator will receive less samples. Sharp output distributions, for example, have a peak that essentially "eclipses" other classes. To illustrate this, consider the case where no noise is added at all; here, the output is always the plurality vote, and a black-box querying adversary cannot learn *anything* about the histogram except its top-voted class, which is already known after a single query.

Our results indicate that the mitigation of this eclipsing effect by increasing the noise, can be more dominant than the adverse effect that increasing noise has on Monte Carlo convergence. Interestingly, this is not always reflected in PATE's privacy-cost score, which is often lower for setups that leak more on vote histograms. Technically, there is no contradiction: privacy cost measures differential privacy, which does not necessarily translate to protection against vote-histogram leakage.

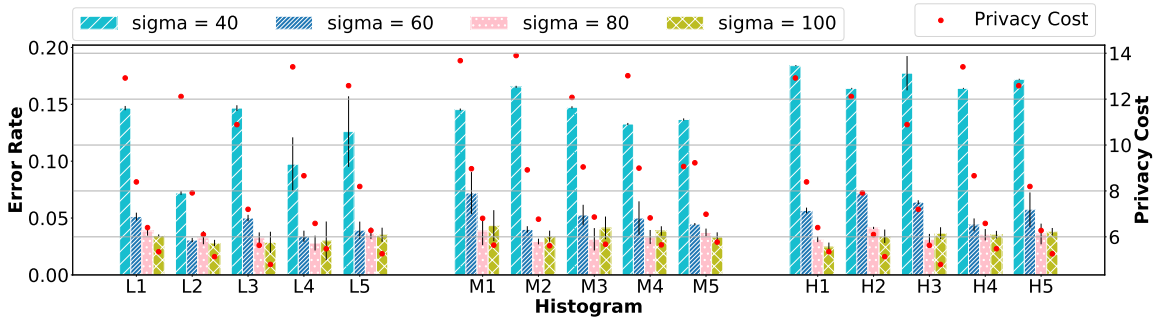


Figure 6.4: Our attack's error extracting 15 MNIST histograms with low/medium/high consensus (L1-5, M1-5, and H1-5 respectively) using different noise scales and a query limit of 10^4. The red dots and the right axis show the privacy cost of the attack on each histogram.

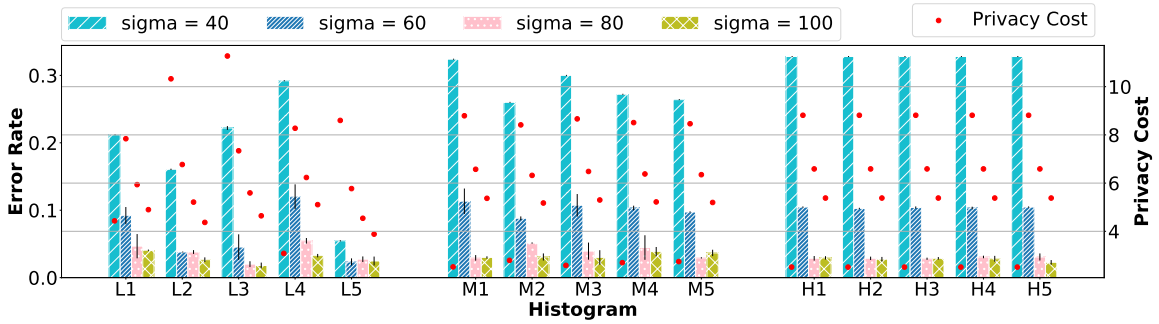


Figure 6.5: Our attack's error extracting 15 SVHN histograms with low/medium/high consensus (L1-5, M1-5, and H1-5 respectively) using different noise scales and a query limit of 10^4. The red dots and the right axis show the privacy cost of the attack on each histogram.

6.3 End-to-end Sensitive-attribute Inference

In Section 4, we showed that histograms leak by mounting an attack that classifies histograms to low-consensus and high-consensus groups, which reveals information about minority-group membership. In Section 6, we showed that we can extract histograms by querying PATE instances. Now, we combine these two attacks, to extract minority-group membership information directly from a PATE instance. Our setting mirrors the setting from Section 4, but the attacker does not have direct access to histograms of individuals, and instead, they extract them from PATE's answers using our methodology (Section 5). We used the same ensemble from Section 4, but this time, the 250 teachers' vote histogram was noised, again using $\sigma = 40$, $\delta = 0.00001$ and a privacy budget of 1.9 as in [28]. We sampled 10 low-consensus and 10 high-consensus members of the test set, and ran the attack on them: we queried PATE with each member's data record until exhausting the privacy budget, computed the Monte Carlo estimators, ran the optimization to recover the vote histogram, and then classified it to low-consensus/high-consensus as in Section 4. Results are given in Figure 6.6, and indeed, they mirror the results of the attack in Section 4.

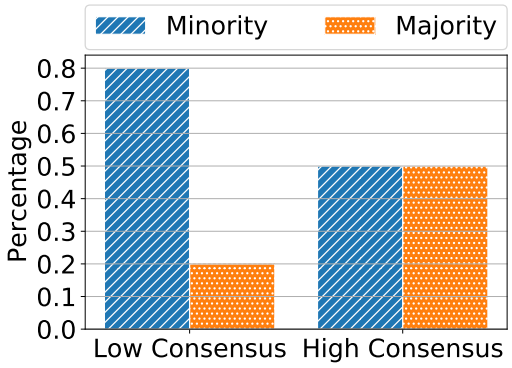


Figure 6.6: High vs. low-consensus distributions of the PhD-detection attack on PATE: vote histograms of minority-group members present lower consensus, allowing an attacker to identify them.

6.4 Edge Values for Noise

Here, our purpose is to evaluate our attack given extremely low and extremely high values of σ. We repeated the query-number-limited attack from Section 6.1 where adversaries perform 10^4 queries. This time, we used a σ value approaching 0 and a very high one (400). Figure 6.7 shows that when noise is close to 0, the error rate is the highest; it then drops and climbs again as we increase the error. This is consistent with what we would expect: we know that when $\sigma = 0$, the attacker cannot learn anything but the argmax class, whereas if σ is infinitely large, PATE's output distribution is uniform regardless of the underlying votes, and the attacker again cannot learn anything.

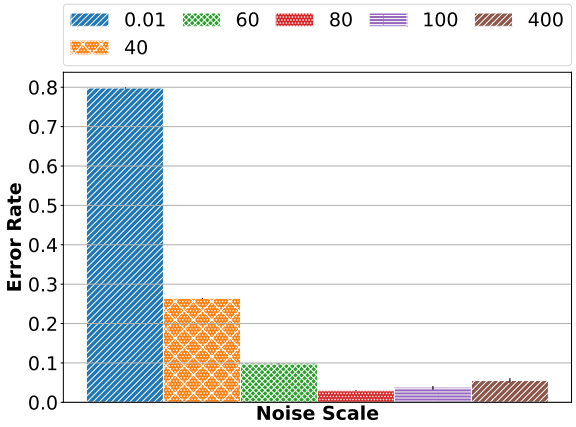


Figure 6.7: Error rates with baselines of a median-consensus histogram (from H3) in SVHN. When the noise is close to 0, the error is the largest; at some point, the error starts moderately increasing as the noise increases.

Chapter 7

Discussion

7.1 Mitigation

The possibility of this attack is inherent to PATE's aggregation mechanism, as long as the attacker can make multiple queries to PATE. Our experiments in Section 6 show that (1) using tighter privacy budgets does not necessarily mitigate the attack, as there is no strong correspondence between the privacy cost and the attack's success, and (2) it would be hard to limit the number of queries some other way without crippling PATE's utility, because our attack is successful while using the same number of queries used in common scenarios from the literature.

Theoretically, the attack would be mitigated if PATE returned a consistent answer for each query. PATE can thus try to cache answers to past queries and not recalculate them. Unfortunately, this defense would be exposed to adversarial perturbations that try to evade the caching mechanism without affecting predictions, and would not be possible for settings that keep queries confidential and/or include decentralized aggregation [5].

Finally, we can try to prevent sensitive information from leaking onto vote histograms. Particularly, models that generalize well across subgroups will be more immune to an attacker inferring group membership via consensus. This reduces to the problem of *subgroup fairness*, an active line of work with many proposed approaches [3, 15, 16, 23, 6] but no silver-bullet solutions.

7.2 Limitations

Empirical analysis of sensitive-attribute leakage onto vote histograms (Section 4) can be expanded to improve more sophisticated attackers, other scenarios, and also other forms of sensitive information that can leak onto histograms. We instead focus our work on extracting histograms from PATE, noting that this can be used as a foundation for various different attacks.

A full optimization procedure takes a noticeably long time (roughly 10 minutes for a single step and 13 hours to convergence on a histogram), which prevents us from fully optimizing its hyper-parameter choices. This is, however, a limitation of our current experimental setup, not of the attack, bearing the main consequence that we are potentially under-estimating our attack's capabilities.

7.3 Related Work

7.3.1 PATE Extensions and Applications

PATE is a widely adapted framework to get differentially-private ML. Long et al. [20] proposed a framework based on PATE that trained a data generator that is scalable and differentially private. Yang et al. [35] used PATE and adversarial autoencoder to provide differential privacy for the speech classifications. To improve PATE's privacy, inspired by Ghazi et al. [10]'s work on Label differential privacy, Esmaeili et al. [8] applied their technique to PATE using the observation of PATE's main assumption that there are promised to exist free public dataset that is available to access. To improve PATE's querying party's confidentiality, Choquette-Choo et al. [5] proposed CaPC, a protocol as an extension of PATE. CaPC is PATE combined with multiparty computation (MPC) and Homomorphic Encryption (HE), where MPC and HE provide the querying party's confidentiality, and teachers' confidentiality still comes from PATE's additive noise mechanism. To improve the utility while keeping privacy-preserving, Wang et al. [32] allow teachers to vote on a gradient vector using gradient compression techniques. However, the differential privacy guarantees of these applications and improvements of PATE frameworks are inherited from PATE's framework, i.e., using additive noise to protect label privacy. Our attack is generally applicable to many of those frameworks, which inherit their privacy analysis from PATE.

7.3.2 Federated Learning

Another prominent decentralized ML framework, Federated Learning (FL) [17], has been extensively investigated from a privacy perspective. As we did for PATE in this work, prior work attacking FL uncovered numerous forms of leakage. For example, Hitaj et al. [11] reconstructed the average training set representation of each class; Geiping et al. [9] reconstructed training data with high fidelity; Nasr et al. [24] mounted a membership inference attack against the clients; Wang et al.[33] showed how a malicious server could distinguish multiple properties of data simultaneously; and Melis et al. [21] inferred the clients' training data sensitive properties. These prior efforts all focus on FL, and are orthogonal to ours. We are the first to evaluate any attack against PATE.

Chapter 8

Future Work

The goal of the research focuses on the privacy and utility of decentralized learning schemes. As a first case study, we considered for the first time the popular decentralized learning framework PATE. It turns out that the privacy-addressing scheme, PATE, which provides ostensibly strong guarantees expressed in the powerful formalism of differential privacy, does not always offer the protection it set out to achieve. This is not because their theoretical differential privacy guarantees are violated, but rather, because attackers can break confidentiality even without violating these guarantees. PATE's differential privacy mechanism ensures that (stated informally) a single prediction does not expose any information on a single data point used for training, but given multiple queries, attackers can, in fact, reconstruct the entire vote histogram, thus entirely breaking any guarantee that differential privacy offers. Moreover, this fine-grained form of leakage, i.e., learning the entire histogram, is possible because of the differential privacy mechanism, which, by repeatedly noising the histogram, in fact adds a diversity of information to the attacker's queries' answers. Revealed and motivated by the project, this chapter discusses future works, open problems, and challenges.

8.1 Challenge of Defense

Since Our attack requires the adversary to send the same query over and over again repeatedly, it is intuitive to think that PATE could prevent Monte Carlo sampling for the same instance by tracking individual instance-query counts (e.g., via maintaining a mapping between a query hash to its count), and reject queries that repeat too many times [4]. However, the adversary can break this defense by adding small perturbations to the instance, which do not change the prediction result or change them very slightly. Perhaps the defense can adapt to this minor-perturbation attack by clustering received queries and detecting large clusters of queries that are very "close" to each other, for example, within a small $L2$-distance ball of each other, and then limit the number of queries from that cluster. And yet again, the attacker could adapt to this defense by mounting "semantic collision" attacks [30, 12, 13], that significantly modify inputs without modifying the model's outputs or internal representations of these inputs. However, all the above are on a high level and have not been tested by experiments. In future work, we could add the defense mechanism to PATE and test against the improved attack.

8.2 Membership Inference Attack

Based on histogram reconstruction, this work could be extended to an end-to-end membership inference attack. In membership inference [29], an attacker tries to learn whether a specific point of interest was in or out of the victim model's training set (in this case, any one of the teacher models). The design of membership inference in PATE builds on the fact that, in many real-world scenarios, data points that are not in the training set would get assigned nearly-uniformly-drawn labels by the set of teachers. Conversely, data points that are in one or more of the teachers' training sets will typically get assigned the correct label in a relatively higher portion of teacher models. It is possible that an attacker can distinguish these two cases by observing the vote histogram that they extracted using the existing attack. Such an attack can be used to audit the privacy that PATE offers in practice and provides a guaranteed lower bound [14] .

8.3 Characterization of Privacy-utility Tradeoff

The final goal of this project attempts to answer the core question of whether practical and secure decentralized learning is possible. Studying existing frameworks reveals a trade-off between utility and privacy, where it is hard to attain both to a sufficient degree. For example, PATE, in its original form, provides high utility, yet its confidentiality against practical adversaries is limited. Attacks can be mitigated by limiting queries, but then, again, utility is hindered. Future work will focus on better characterizing this tradeoff. First, by devising mitigations, such as the sophisticated query-limiting regime we envision for PATE, that detects attacks that dispatch the same query multiple times, even in the presence of adversarial perturbations that try to evade it. Second, by designing novel approaches for decentralized learning, and particularly fully decentralized ones that do not rely on a central party who can be corrupted.

Chapter 9

Conclusion

We are the first to audit the confidentiality of PATE from an adversarial perspective. Our attack extracts histograms of votes, which can reveal attributes of the input, such as race or gender, or help attackers characterize teacher partitions. The attacker's success is not highly correlated with the privacy cost of their queries, which is monitored by PATE. Thus, mitigation of this attack is nontrivial and/or significantly hinders prediction utility. Particularly, using larger Gaussian noise, even when it fortifies the differential privacy guarantee, actually increases the risk to the confidentiality of the vote histogram. This surprising tension demonstrates that care must be taken to analyze differential privacy protection within a given threat model, rather than treat it as a silver bullet protecting against any form of leakage.

9.1 Broader Impact

Our work studies information leakage in a widely adopted system, thus promoting our understanding of its risks. Our adversarial method can be used by developers and auditors to evaluate the confidentiality and privacy promises of PATE-based frameworks.
Our observation that differential privacy does not prevent but rather enables the attack is the first of its kind in that it reveals a discrepancy between differential privacy and societal norms of privacy. Characterizing this distinction is essential to building technology that uses technical definitions of privacy as an instrument to protect privacy norms.

Bibliography

[1] Martin Abadi et al. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Oct. 2016). DOI: 10.1145/2976749.2978318. URL: http://dx.doi.org/10.1145/2976749.2978318.

[2] Martin Abadi et al. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[3] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[4] Steven Chen, Nicholas Carlini, and David Wagner. "Stateful Detection of Black-Box Adversarial Attacks". In: *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*. SPAI '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 30–39. ISBN: 9781450376112. DOI: 10.1145/3385003.3410925. URL: https://doi.org/10.1145/3385003.3410925.

[5] Christopher A. Choquette-Choo et al. "Ca{PC} Learning: Confidential and Private Collaborative Learning". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=h2EbJ4_wMVq.

[6] Yin Cui et al. "Class-balanced loss based on effective number of samples". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9268–9277.

[7] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[8] Mani Malek Esmaeili et al. "Antipodes of Label Differential Privacy: PATE and ALIBI". In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021. URL: https://openreview.net/forum?id=sR1XB9-F-rv.

[9] Jonas Geiping et al. "Inverting Gradients - How easy is it to break privacy in federated learning?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 16937–16947. URL: https://proceedings.neurips.cc/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf.

[10] Badih Ghazi et al. *Deep Learning with Label Differential Privacy*. 2021. arXiv: 2102.06062 [cs.LG]. URL: https://arxiv.org/abs/2102.06062.

[11] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning". In: *CoRR* abs/1702.07464 (2017). arXiv: 1702.07464. URL: http://arxiv.org/abs/1702.07464.

[12] Jörn-Henrik Jacobsen et al. "Excessive invariance causes adversarial vulnerability". In: *arXiv preprint arXiv:1811.00401* (2018).

[13] Jörn-Henrik Jacobsen et al. "Exploiting excessive invariance caused by norm-bounded adversarial robustness". In: *arXiv preprint arXiv:1903.10484* (2019).

[14] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. "Auditing differentially private machine learning: How private is private sgd?" In: *arXiv preprint arXiv:2006.07709* (2020).

[15] Michael Kearns et al. "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2564–2572.

[16] Michael J Kearns, Robert E Schapire, and Linda M Sellie. "Toward efficient agnostic learning". In: *Machine Learning* 17.2 (1994), pp. 115–141.

[17] Jakub Konečný et al. "Federated Learning: Strategies for Improving Communication Efficiency". In: *NIPS Workshop on Private Multi-Party Machine Learning*. 2016. URL: https://arxiv.org/abs/1610.05492.

[18] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[19] Yunhui Long et al. "G-PATE: Scalable Differentially Private Data Generator via Private Aggregation of Teacher Discriminators". In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021.

[20] Yunhui Long et al. *G-PATE: Scalable Differentially Private Data Generator via Private Aggregation of Teacher Discriminators*. 2021. arXiv: 1906.09338 [cs.LG]. URL: https://arxiv.org/abs/1906.09338.

[21] Luca Melis et al. "Inference Attacks Against Collaborative Learning". In: *CoRR* abs/1805.04049 (2018). arXiv: 1805.04049. URL: http://arxiv.org/abs/1805.04049.

[22] Ilya Mironov. "Rényi differential privacy". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[23] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. "Agnostic federated learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4615–4625.

[24] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning". In: *2019 IEEE Symposium on Security and Privacy (SP)* (May 2019). DOI: 10.1109/sp.2019.00065. URL: http://dx.doi.org/10.1109/SP.2019.00065.

[25] Yuval Netzer et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[26] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. "Smooth sensitivity and sampling in private data analysis". In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. 2007, pp. 75–84.

[27] Nicolas Papernot et al. *Scalable Private Learning with PATE*. 2018. arXiv: 1802.08908 [stat.ML].

[28] Nicolas Papernot et al. *Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data*. 2017. arXiv: 1610.05755 [stat.ML].

[29] Reza Shokri et al. “Membership inference attacks against machine learning models”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 3–18.

[30] Congzheng Song, Alexander M. Rush, and Vitaly Shmatikov. *Adversarial Semantic Collisions*. 2020. arXiv: 2011.04743 [cs.CL].

[31] Boxin Wang et al. “DataLens: Scalable Privacy Preserving Training via Gradient Compression and Aggregation”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Nov. 2021). DOI: 10.1145/3460120.3484579. URL: http://dx.doi.org/10.1145/3460120.3484579.

[32] Boxin Wang et al. “DataLens: Scalable Privacy Preserving Training via Gradient Compression and Aggregation”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021).

[33] Zhibo Wang et al. “Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning”. In: *CoRR* abs/1812.00535 (2018). arXiv: 1812.00535. URL: http://arxiv.org/abs/1812.00535.

[34] Chao-Han Huck Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. “PATE-AAE: Incorporating Adversarial Autoencoder into Private Aggregation of Teacher Ensembles for Spoken Command Classification”. In: *CoRR* abs/2104.01271 (2021). arXiv: 2104.01271. URL: https://arxiv.org/abs/2104.01271.

[35] Chao-Han Huck Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. “PATE-AAE: Incorporating Adversarial Autoencoder into Private Aggregation of Teacher Ensembles for Spoken Command Classification”. In: *CoRR* abs/2104.01271 (2021). arXiv: 2104.01271. URL: https://arxiv.org/abs/2104.01271.