Privately Publishable Per-instance Privacy

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider how to release personalized privacy losses using per-instance differential privacy (pDP), focusing on private empirical risk minimization over the class of generalized linear models. Standard differential privacy (DP) gives us a worst-case bound that might be orders of magnitude larger than the privacy loss to a particular individual relative to a fixed dataset. The pDP framework provides a more fine-grained analysis of the privacy guarantee to a target individual, but the per-instance privacy loss itself might be a function of sensitive data. In this paper, we analyze the per-instance privacy loss of releasing a private empirical risk minimizer learned via objective perturbation, and propose a group of methods to privately and accurately publish the pDP losses at little to no additional privacy cost.

1 Introduction

An explosion of data has fueled innovation in machine learning applications and demanded, in equal turn, privacy protection for the sensitive data with which machine learning practitioners train and evaluate models.

Differential privacy (DP) (Dwork et al., 2006, 2014a) has become a mainstay of privacy-preserving data analysis, replacing less robust privacy definitions such as k-anonymity which fail to protect against sufficiently powerful de-anonymization attacks (Narayanan & Shmatikov, 2008) . In contrast, DP offers provable privacy guarantees that are robust against an arbitrarily strong adversary.

The data curator could trivially protect against privacy loss by reporting a constant function, or by releasing only data-independent noise. The key challenge of DP is to release privatized output that retains utility to the data analyst.

While the data-independent formulation of standard DP is inarguably advantageous, in many cases the bound on the worst-case privacy loss guaranteed by DP is overly conservative for particular individuals with respect to a fixed dataset. On the flip side, a data curator might spend a large portion of the privacy budget protecting outliers in the dataset at the cost of model accuracy.

The privacy-utility trade-off challenges the practicality of differential privacy. A desired level of utility in a machine learning application might necessitate a high value of ϵ, but the privacy guarantees degrade quickly past $\epsilon = 1$. (Triastcyn & Faltings, 2020) construct an example whereby a differentially private algorithm with $\epsilon = 2$ allows an attacker to use a maximum-likelihood estimate to conclude with up to 88% accuracy that an individual is in a dataset.

Moreover, differentially private applications in practice commonly use high values of ϵ. A study of Apple's deployment of differential privacy revealed that the overall daily privacy loss permitted by the system was as high as $\epsilon = 6$ for Mac OS 10.12.3 and $\epsilon = 14$ for iOS 10.1.1 (Tang et al., 2017) – offering only scant privacy protection!

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Per-instance differential privacy shows promise as a means of navigating the privacy-utility trade-off. The privacy loss to a particular individual relative to a fixed dataset might be orders of magnitude smaller than the worst-case bound guaranteed by standard DP. In this case, an algorithm meeting a desired level of utility but providing weak DP guarantees may, for the same level of utility, achieve drastically more favorable *per-instance* DP guarantees.

The remaining challenge is that the per-instance privacy loss is a function of the entire dataset; publishing it directly would negate the purpose of privately training a model in the first place! In this paper, we propose a methodology to privately release the personalized privacy losses associated with private empirical risk minimization. Our contributions are as follows:

- We present a novel analysis of the per-instance privacy losses incurred by the objective perturbation mechanism, demonstrating that these pDP losses are orders of magnitude smaller than the worst-case guarantee of differential privacy.
- We propose a group of methods to privately and accurately release the pDP losses. In the particular case of generalized linear models, we show that we can accurately publish the private pDP losses using a dimension- and dataset-independent bound. Furthermore, we present an alternative approach to privately release data-dependent bounds that provide a provably tight multiplicative approximation to the pDP losses and low privacy-loss overhead.

1.1 Related Work

This paper builds upon (Wang, 2019), which proposed the per-instance DP framework and left as an open question the matter of publishing the pDP losses. Notably, we embellish the pDP framework to provide privacy guarantees that adapt even more fluidly to data-dependent properties of our algorithms. Another fundamental ingredient in our privacy analysis is the objective perturbation algorithm (`Obj-Pert`) of (Chaudhuri et al., 2011), further analyzed by (Kifer et al., 2012), which privately releases the minimizer of an empirical risk by adding a linear perturbation to the objective function before optimizing.

Our paper joins a body of work that seeks to provide stronger privacy guarantees by taking into account properties of the data. (Wang, 2019) gives an overview of several of these methodologies, including propose-test-release (Dwork & Lei, 2009) and local sensitivity (Nissim et al., 2007). In addition, Bayesian differential privacy (Triastcyn & Faltings, 2020) provides data-dependent privacy guarantees that afford strong protection to "typical" data (drawn from the same distribution as the dataset). The pDP definition that we use in this paper differs from the Bayesian DP formulation in that we condition on the differing individual's data, rather than making distributional assumptions about the sensitive data. The Rényi-based privacy filters of (Feldman & Zrnic, 2020) are also closely related to our work; the authors study composition of personalized (but not per-instance) privacy losses using adaptively-chosen privacy parameters. Our privacy losses also depend on the output of the computations, but we choose to analyze this through the lens of *ex-post* differential privacy (Ligett et al., 2017). The work of (Papernot et al., 2018) initiated the problem of privately publishing data-dependent privacy losses but considered neither *per-instance* nor *ex-post* settings.

2 Preliminaries

2.1 Notation

We use conventional notation for common statistical objects: $\Pr[\cdot]$ for probability, $\mathbb{E}[\cdot]$ for expectation, etc. Adopting a standard abuse of notation, we write the output of a randomized algorithm $\mathcal{A}$ as $\mathcal{A}(\cdot)$, and for continuous distributions we take $\Pr[\mathcal{A}(D) = o]$ to be the value of the probability density function at output o.

We will let z refer to both an individual and their data; i.e., individual z holds data $z = (x, y)$. $D_{\pm z} \in \mathcal{Z}^*$ denotes the fixed dataset $D = \{z_1, \ldots, z_n\} \in \mathcal{Z}^*$ with the data point z removed if $z \in D$, and the point z added if $z \notin D$.

Our paper considers a perturbed version of the following optimization problem:

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} L(\theta; D) + r(\theta),$$

where $L(\theta; D) = \sum_{i=1}^{n} \ell(\theta; z_i)$. Throughout, we assume that $\ell(\theta; z)$ and $r(\theta)$ are convex and twice-differentiable.

We distinguish between ϵ as fixed input to a DP algorithm, and $\epsilon(\cdot)$ as a function parameterized according to a particular DP relaxation — e.g., $\epsilon(o, D, z)$ means the *ex-post* per-instance privacy loss conditioned on output o, dataset D, and data point z.

2.2 Differential Privacy

Let $\mathcal{Z}$ denote the data domain, and $\mathcal{R}$ the set of all possible outcomes of algorithm $\mathcal{A}$. Fix $\epsilon \geq 0, \delta \geq 0$.

Definition 1. (Differential privacy) A randomized algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{R}$ satisfies (ϵ, δ)-DP if for all datasets $D \in \mathcal{Z}^*$ and data points $z \in \mathcal{Z}$, and for all measurable sets $S \subset \mathcal{R}$, it holds that

$$\Pr\big[\mathcal{A}(D) \in S\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D_{\pm z}) \in S\big] + \delta.$$

Differential privacy guarantees that the presence or absence of any particular data record has little impact on the output distribution of a randomized algorithm. In this paper we use the "add/remove" notion of DP, by which we construct neighboring dataset $D_{\pm z}$ by adding or removing an individual z from dataset D.

DP is powerful and universal in that its guarantee applies to any D, z and any output events. However, there are often situations where the privacy losses of $\mathcal{A}$ vary drastically depending on its input data, and the privacy loss bound ϵ (protecting even the worst-case pair of neighboring datasets) may not be informative of the privacy loss incurred to individuals when the input to $\mathcal{A}$ is typical. This motivated Wang (2019) to consider a per-instance version of the DP definition.

Definition 2. (Per-instance differential privacy) A randomized algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{R}$ satisfies $\big(\epsilon(D, D_{\pm z}), \delta\big)$-pDP if for dataset D and data point z, and for all measurable sets $S \subset \mathcal{R}$, it holds that

$$\Pr\big[\mathcal{A}(D) \in S\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D_{\pm z}) \in S\big] + \delta,$$
$$\Pr\big[\mathcal{A}(D_{\pm z}) \in S\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D) \in S\big] + \delta.$$

The pDP definition can be viewed as using a function $\epsilon(D, D_{\pm z})$ that more precisely describes the privacy guarantee in protecting a fixed data point z when $\mathcal{A}$ is applied to dataset D.

As it turns out, it is more convenient for us to work with an even more *instance-specific* description of the privacy loss that is further parameterized by the realized output of $\mathcal{A}$ *ex-post* — after the random coins of $\mathcal{A}$ are flipped and the outcome released.

Definition 3. (*Ex-post* per-instance differential privacy) A randomized algorithm $\mathcal{A}$ satisfies $\epsilon(\cdot)$-*ex-post* per-instance differential privacy for an individual z and a fixed dataset D at an outcome $o \in \mathrm{Range}(\mathcal{A})$ if

$$\left| \log \left(\frac{\Pr\big[\mathcal{A}(D) = o\big]}{\Pr\big[\mathcal{A}(D_{\pm z}) = o\big]} \right) \right| \leq \epsilon(o, D, D_{\pm z}).$$

This definition generalizes the *ex-post* DP definition (Ligett et al., 2017) (introduced for a different purpose) to a *per-instance* version that depends on a given pair of neighboring datasets. The above quantity is essentially the absolute value of the log-odds ratio, used extensively in hypothesis testing. Intuitively, the *ex-post* per-instance privacy loss $\epsilon(o, D, D_{\pm z})$ describes how confidently an attacker could infer, given the output of algorithm $\mathcal{A}$, whether or not individual z is in dataset D.

Despite (or perhaps because of) its precise accounting for privacy, *ex-post* pDP could reveal sensitive information about the dataset, as the following example explicitly illustrates.

Example 4 (The privacy risk of exposing *ex-post* pDP)**.** *Consider a standard Gaussian mechanism $\mathcal{A}$ that adds noise to a counting query Q applied to dataset D, i.e. $\mathcal{A}(D) = Q(D) + \mathcal{N}(0, \sigma^2)$. Q has global sensitivity $\Delta_Q = 1$. We will show that an attacker, knowing only the output o of algorithm $\mathcal{A}$, her* ex-post *pDP loss and that her individual data is not contained in dataset D, can conclusively uncover the sensitive quantity $Q(D)$ protected by algorithm $\mathcal{A}$.*

The output o of algorithm $\mathcal{A}$ is distributed as $o \sim \mathcal{N}(Q(D), \sigma^2)$. So the ex-post *pDP can be calculated as $\epsilon(o, D, D_{\pm z}) = \frac{|Q(D) - Q(D_{\pm z})||2o - Q(D) - Q(D_{\pm z})|}{2\sigma^2}$.*

Enter attacker z, who has auxiliary information: she knows that her own individual data is not contained in D. After algorithm $\mathcal{A}$ is applied to D, attacker z receives output $o = 1$ and is informed of her ex-post *pDP $\epsilon(o, D, D_{+z})$. Since $Q(D_{+z}) = Q(D) + 1$ is known, attacker z can solve for $Q(D)$ and obtain $Q(D) = o - 0.5 \pm \sigma^2\epsilon(o, D, D_{+z})$. With probability 1, only one of the two possibilities is an integer*[1]*. Therefore, exposing* ex-post *pDP in this case completely reveals $Q(D)$.*

Problem statement. The lesson of Example 4 is that we cannot directly reveal the *ex-post* pDP losses without potentially nullifying the algorithm's privacy benefits. How, then, can we privately and accurately publish the *ex-post* pDP losses?

The goal of this paper is to develop an algorithm that publishes a *function* $\tilde{\epsilon} : \mathcal{Z} \to \mathbb{R}$ whose output estimates the *ex-post* pDP loss to an individual z of releasing the solution to a private ERM problem. Any individual (not just those whose data is contained in the dataset) can plug in her own data z into this function in order to receive a high probability bound on her *ex-post* pDP loss which does not depend directly on any sensitive data except her own.

This requirement offers the same type of privacy protection as joint differential privacy (Kearns et al., 2014), which relaxes the standard DP definition by allowing an algorithm's output to individual z to be sensitive only in her own private data. Our notion of privacy is slightly more general in that it holds for individuals both in and out of the dataset. The difference lies in how the algorithm's output space is defined; whereas a joint DP algorithm produces a fixed-length tuple partitioning the output to each individual in the dataset, our algorithm outputs a function whose domain includes any data point $z \in \mathcal{Z}$. As a result, our methods are robust against collusion by arbitrary coalitions of adversaries, allowing repeated queries by any group of individuals without invalidating the privacy guarantees promised by the pDP losses.

2.3 Problem Setting

We consider a general family of problems known as *private empirical risk minimization* (ERM), which aim to approximate the solution to an ERM problem while preserving privacy. That is, we wish to privately solve optimization problems of the form

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\theta; D) + r(\theta),$$

where $r(\theta)$ is a convex and twice-differentiable regularizer and $L(\theta; D) = \sum_{i=1}^{n} \ell(\theta; z_i)$ is a loss function. Dataset D is given by $D = \{z_i\}_{i=1}^{n}$, and $z_i = (x_i, y_i)$ for $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} \subseteq \mathbb{R}$. $\Theta \subseteq \mathbb{R}^d$ is a convex domain.

Our assumptions on the data distribution are fairly mild. We posit only that $x \in \mathcal{X}$ are sampled from some distribution on the unit ball, so that $||x|| \leq 1$ for all $x \in \mathcal{X}$. We also assume that $|y| \leq 1$ for all $y \in \mathcal{Y}$.

Our pDP analysis will consider objective perturbation, a well-known approach for privacy-preserving ERM. We review this algorithm and its privacy guarantees in the next section.

2.4 Objective Perturbation

The objective perturbation algorithm solves

$$\hat{\theta}^P = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\theta; D) + r(\theta) + \frac{\lambda}{2}||\theta||^2 + b^T\theta, \tag{1}$$

where $b \sim \mathcal{N}(0, \sigma^2 I_d)$ .Below, we present a simplified version of the objective perturbation algorithm and state its privacy guarantees.

[1] Take $Q(D) = 0$ and $o = 0.1$ as an example, the two possibilities are 0 and -0.8.

Algorithm 1 Release $\hat{\theta}^P$ via `Obj-Pert` (Kifer et al., 2012)

Input: Dataset D, noise parameter σ, regularization parameter λ, loss function $L(\theta; D) = \sum_i \ell(\theta; z_i)$, convex and twice-differentiable regularizer $r(\theta)$.
Output: $\hat{\theta}^P$, the minimizer of the perturbed objective.
Draw noise vector $b \sim \mathcal{N}(0, \sigma^2 I)$.
Compute $\hat{\theta}^P$ according to (1).

Theorem 5 (Privacy guarantees of Algorithm 1 (Kifer et al., 2012))**.** *Consider dataset $D = \{z_i\}_{i=1}^n$; loss function $L(\theta; D) = \sum_i \ell(\theta; z_i)$; convex regularizer $r(\theta)$; and convex domain Θ. Let β be an upper bound on the eigenvalues of the Hessian $\nabla^2 \ell(\theta; z_i)$, for all $z_i \in \mathcal{X} \times \mathcal{Y}$ and for all $\theta \in \Theta$. Let $\ell(\cdot)$ have a bounded gradient such that $||\nabla \ell(\theta; z_i)|| \leq \xi$ for all $z_i \in \mathcal{X} \times \mathcal{Y}$ and for all $\theta \in \Theta$. For $\lambda \geq \dfrac{2\beta}{\epsilon_1}$ and $\sigma = \dfrac{\xi^2(8\log(2/\delta) + 4\epsilon_1)}{\epsilon_1^2}$, Algorithm 1 satisfies (ϵ_1, δ_1)-differential privacy.*

The objective perturbation algorithm chooses a regularization strength to ensure that the objective function is strongly convex, and also requires smoothness. This means that the objective function is well-conditioned and therefore robust to small perturbations. In other words, for sufficiently large λ, the minimizer of an objective function that is λ-strongly convex will be insensitive to any particular data point.

3 Privately Publishable pDP

3.1 pDP Analysis of Objective Perturbation

Our goal in this section is to derive the personalized privacy losses associated with observing the output $\hat{\theta}^P$ of objective perturbation under Definition 3. As it turns out, this *ex-post* perspective is not only more adaptive, but also more convenient for our analysis of Algorithm 1, whose privacy parameters are a function of the data. Since we are analyzing the per-instance privacy cost of *releasing* $\hat{\theta}^P$, it makes perfect sense to condition the pDP loss on the privatized output of the computation.

Our first technical result is a precise calculation of the *ex post* pDP loss of objective perturbation.

Theorem 6 (*ex-post* pDP loss of objective perturbation for a convex loss function)**.** *Let $J(\theta; D) = L(\theta; D) + r(\theta) + \frac{\lambda}{2}||\theta||^2$ such that $L(\theta; D) + r(\theta) = \sum_i \ell(\theta; z_i) + r(\theta)$ is a convex and twice-differentiable regularized loss function, and sample $b \sim \mathcal{N}(0, \sigma^2 I)$. Then for every privacy target $z = (x, y)$, releasing $\hat{\theta}^P = \operatorname{argmin}_{\theta \in \mathbb{R}^d} J(\theta; D) + b^T\theta$ satisfies $\epsilon_1(\hat{\theta}^P, D, z)$-ex-post per-instance differential privacy with*

$$\epsilon(\hat{\theta}^P, D, D_{\pm z}) = \left| -\log \prod_{j=1}^{d} \left(1 \mp \mu_j\right) + \frac{1}{2\sigma^2}||\nabla \ell(\hat{\theta}^P; z)||^2 \pm \frac{1}{\sigma^2} \nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z) \right|,$$

where $\mu_j = \lambda_j u_j^T \left(\nabla \mathbf{b}(\hat{\theta}^P; D) \mp \sum_{k=1}^{j-1} \lambda_k u_k u_k^T\right)^{-1} u_j$ according to the eigendecomposition $\nabla^2 \ell(\theta; z) = \sum_{k=1}^d \lambda_k u_k u_k^T$.

Proof sketch. Following the analysis of (Chaudhuri et al., 2011), we establish a bijection between the mechanism output $\hat{\theta}^P$ and the added noise b, and use a change-of-variables defined by the Jacobian mapping between $\hat{\theta}^P$ and b in order to rewrite the log-probability ratio in terms of the probability density function of b. Next we borrow a trick from (Kifer et al., 2012), observing that since $\hat{\theta}^P$ is the minimizer of the private objective function $J(\theta; D) + b^T\theta$, we can set its gradient to 0 and solve directly for the distribution of b. To calculate the first term of the above equation, we use the eigendecomposition of the Hessian $\nabla^2 \ell(\hat{\theta}^P; z)$ and recursively apply the matrix determinant lemma. The rest of the proof is straightforward algebra.

□

The above expression holds for any convex loss function, but is a bit unwieldy. The calculation becomes much simpler when we assume $\ell(\cdot)$ to be a linear loss function, with inner-product form $\ell(\theta; z) = f(x^T\theta; y)$. For the sake of interpretability, we will defer further discussion of the *ex-post* pDP loss of objective perturbation until after presenting the following corollary.

Corollary 7 (*ex-post* pDP loss of objective perturbation for GLMs)**.** *Let* $J(\theta; D) = L(\theta; D) + r(\theta) + \frac{\lambda}{2}||\theta||^2$ *such that* $L(\theta; D) = \sum_i \ell_i(\theta)$ *is a linear loss function, and sample* $b \sim \mathcal{N}(0, \sigma^2 I)$*. Then for every privacy target* $z = (x, y)$*, releasing* $\hat{\theta}^P = \operatorname{argmin}_{\theta\in\Theta} J(\theta; D) + b^T\theta$ *satisfies* $\epsilon_1(\hat{\theta}^P, D, D_{\pm z})$*-ex-post per-instance differential privacy with*

$$\epsilon(\hat{\theta}^P, D, D_{\pm z}) \leq \left| -\log\left(1 \pm f''(\cdot)\mu(x)\right) + \frac{1}{2\sigma^2}||\nabla\ell(\hat{\theta}^P; z)||^2 \pm \frac{1}{\sigma^2}\nabla J(\hat{\theta}^P; D)^T\nabla\ell(\hat{\theta}^P; z)\right|,$$

where $\mu(x) = x^T\left(\nabla^2 J(\hat{\theta}^P; D)\right)^{-1}x$*,* $\nabla\ell(\hat{\theta}^P; z) = f'(x^T\hat{\theta}^P; y)x$ *and* $f''(\cdot)$ *is shorthand for* $f''(\cdot) = f''(x^T\hat{\theta}^P; y)$*.*

Note that the quantity $\mu(x)$ in the first term is the *generalized leverage score* (Wei et al., 1998), quantifying the influence of a data point on the model fit. The second and third terms are a function of the gradient of the loss function and provide a complementary measure of how well the fitted model predicts individual z's data.

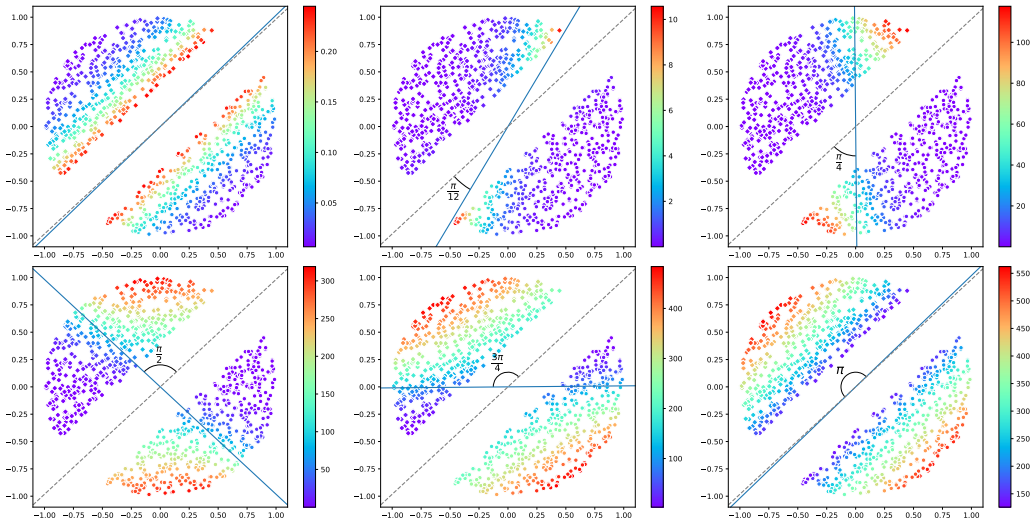

Figure 1: Visualization of *ex-post* pDP losses for logistic regression ($n = 1000, d = 2$).

Since the *ex-post* pDP is a function of $\hat{\theta}^P$, we don't even need to run Algorithm 1 to calculate *ex-post* pDP losses – we can plug in directly to Corollary 7 in order to calculate the pDP distribution induced by any hypothetical $\hat{\theta}^P$. For Figure 1, we use a synthetic dataset D sampled from the unit ball with two linearly separable classes separated by margin $m = 0.4$. Then we solve for $\hat{\theta} = \operatorname{argmin} J(\theta; D)$ with $\lambda = 1$ to minimize the logistic loss, and directly perturb the output by rotating it by angle $\omega \in [0, \frac{\pi}{12}, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi]$. The color scale is a function of the *ex-post* pDP loss of data point z.

Figure 1 illustrates how the mechanism output $\hat{\theta}^P$ affects the *ex-post* pDP distribution of objective perturbation for our logistic regression problem. For $\omega \in [0, \frac{\pi}{12}]$, the data points closest to the decision boundary have the highest *ex-post* pDP loss. These data points have a strong effect on the learned model and would therefore have high *leverage scores*, making the first term dominate. As the perturbation (and model error) increases, the second and third terms dominate; the more badly a model predicts a data point, the less protection this data point has.

Hidden in this analysis are the δ's of Algorithm 1, which along with the choice of σ and λ could affect which of the three terms is dominant. Fortunately, the probability of outputting something like $\hat{\theta}^P = \theta + \pi$ is astronomically low for any reasonable privacy setting!

3.2 Releasing the pDP losses

Next we consider: after having released $\hat{\theta}^P$ and calculated the per-instance privacy losses of doing so, how do we privately release these pDP losses?

Observe that the expression from Theorem 6 depends on the dataset D only through two quantities: the leverage score $\mu = x^T\left(\nabla^2 J(\hat{\theta}^P; D)\right)^{-1} x$ and the inner product $\nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z)$. As a result, if we can find a data-independent bound for these two terms, or privately release them with only a small additional privacy cost, then we are done.

3.2.1 Data-independent bound of *ex-post* pDP losses

Below, we present a pair of lemmas which will allow us to find a high-probability, data-independent bound on the *ex-post* pDP loss.

Theorem 8. *Let $\ell(\cdot)$ be β-smooth, such that $\nabla^2 \ell(\theta; z) \prec \beta I_d$ for all $\theta \in \mathbb{R}^d$ and $z \in \mathcal{Z}$.*

$$\left| -\log \prod_{j=1}^{d} \left(1 \mp \mu_j\right) \right| \leq -\sum_{j=1}^{d} \log(1 - \frac{\lambda_j}{\lambda}),$$

where $\mu_j = \lambda_j u_j^T \left(\nabla \mathbf{b}(\hat{\theta}^P; D) \mp \sum_{k=1}^{j-1} \lambda_k u_k u_k^T\right)^{-1} u_j$ according to the eigendecomposition $\nabla^2 \ell(\hat{\theta}^P; z) = \sum_{k=1}^{d} \lambda_k u_k u_k^T$. When specializing to linear loss functions such that $\ell(\theta; z) = f(x^T\theta; y)$, $\lambda_j = 0$ for all $j > 1$ and the above bound can be simplified to $-\log\left(1 - \frac{f''(x^T\hat{\theta}^P)||x||^2}{\lambda}\right)$.

Theorem 9. *Let $\hat{\theta}^P$ be a random variable such that $\hat{\theta}^P = \operatorname{argmin}\left(J(\theta; D) + b^T\theta\right)$ as in (1), where $b \sim \mathcal{N}(0, \sigma^2 I_d)$ and $\ell(\theta; z)$ is a convex and twice-differentiable loss function. Then for $z \in \mathcal{Z}$, the following holds with probability $1 - \rho$:*

$$\left|\nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z)\right| \leq \sigma\sqrt{2\log(2d/\rho)} \|\nabla \ell(\hat{\theta}^P; z)\|_1.$$

For linear loss functions the bound can be substantially strengthened to

$$\left|\nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z)\right| \leq f'(x^T\hat{\theta}^p, y)\sigma||x||\sqrt{2\log(2/\rho)}.$$

We make a few observations on the bounds. First, the general bound in Theorem 9 holds simultaneously for all z and it depends only logarithmically in dimension when the features are *sparse*. Second, the bound for a linear loss function is dimension-free and somewhat surprising because we are actually bounding an inner product of two dependent random vectors (both depend on $\hat{\theta}^p$).

Finally, we remark that the bounds in this section are data-independent in that they do not depend the rest of the dataset beyond already released information $\hat{\theta}^p$. It allows us to reveal a pDP bound of each individual when she plugs in her own data without costing any additional privacy budget!

3.3 The privacy report

For certain regimes, we may wish to consider privatizing the data-dependent quantities of the pDP losses, at an additional privacy cost, as an alternative to using data-independent bounds. Of course, it only makes sense to do so if we can show that (a) these data-dependent estimates are more accurate than the data-independent bounds; (b) the overhead of releasing additional quantities (the additional privacy cost in terms of both DP and pDP) is not too large; and (c) we can share the pDP losses of the private reporting algorithm using data-independent bounds (so we do not have to recursively publish such reports).

Full details are in the appendix. We show that by adding slightly more regularization than required by `Obj-Pert` (i.e., making λ just a bit larger so that the minimum eigenvalue of the Hessian $H = \nabla^2 J$ is above a certain threshold), we can find a multiplicative bound that estimates $\mu(x) = x^T H^{-1} x$ uniformly for all x. We do so by adding noise to the Hessian using a natural variant of "Analyze Gauss" (Dwork et al., 2014b), hence privately releasing $\overline{\mu^P} : \mathcal{X} \to \mathbb{R}$. See Algorithm 2 for

details. For brevity, we use the short-hands $f'(\cdot) := f'(x^T\hat{\theta}^P; y)$ and $f''(\cdot) := f''(x^T\hat{\theta}^P)$, where $\ell(\theta; z) = f(x^T\theta; y)$.

Algorithm 2 Privacy report for `Obj-Pert` on GLMs

Input: $\hat{\theta}^p$ from `Obj-Pert`, noise parameter σ, σ_2; regularization parameter λ; Hessian $H := \sum_i \nabla^2 \ell(\hat{\theta}^p; z_i) + \lambda I_d$, Boolean B $\in$ [`DATA-INDEP`, `DATA-DEP`], failure probability ρ
Output: Reporting function $\tilde{\epsilon} : (x, y), \delta \to \mathbb{R}^2_+$
if B = `DATA-INDEP` **then**
  Set $\epsilon_2(\cdot) := 0$.
  Set $\overline{\mu^p}(x) := \frac{\|x\|^2}{\lambda}$.
else if B = `DATA-DEP` **then**
  Privately release $\hat{H}^p$ by a variant of "Analyze Gauss"[2]with parameter σ_2.
  Set $\epsilon_2(\cdot)$ according to Statement 2 of Theorem 10.
  Set $\tau = F^{-1}_{\lambda_1(\mathrm{GOE}(d))}(1 - \rho/2)$.

  if $\lambda \geq 2\tau\sigma_2$ **then**
    Set $\overline{\mu^p}(x) = \frac{3}{2} x^T [\hat{H}^p]^{-1} x$.
  end if
end if
Set $\overline{\epsilon_1^p}(z) := \max\left\{ \left| -\log\left(1 \pm f''(\cdot)\overline{\mu^p}(x)\right) + \frac{|f'(\cdot)|^2 ||x||^2}{2\sigma^2} \pm \frac{|f'(\cdot)|||x||F^{-1}_{\mathcal{N}(0,1)}(1-\rho/2)}{\sigma} \right| \right\}$.
Output the function $\tilde{\epsilon}(z) := (\overline{\epsilon_1^p(z)}, \epsilon_2(z))$.

Note that the pDP function $\epsilon_2(\cdot)$ – which we use to report the additional pDP loss of releasing $\hat{H}^P$, the private estimate of the Hessian – does not depend on the dataset, and thus is not required to be separately released.

Theorem 10. *There is a universal constant C such that if $\lambda > C\sigma_2\sqrt{d}(1 + (\log(1/\rho))^{2/3})$, then Algorithm 2 satisfies the following properties*

1. $(\frac{\beta^2}{4\sigma_2^2} + \frac{\beta\sqrt{\log(1/\delta)}}{\sigma_2}, \delta)$*-DP*
2. $(\frac{f''(x,\hat{\theta}^p)^2\|x\|^4}{4\sigma_2^2} + \frac{f''(x,\hat{\theta}^p)\|x\|^2\sqrt{\log(1/\delta)}}{\sigma_2}, \delta)$*-pDP for all $x \in \mathcal{X}$ and $0 \leq \delta < 1$.*
3. *For a fixed input z, and all $\rho > 0$, the privately released privacy report $\tilde{\epsilon}(\cdot)$ satisfies that $\epsilon_1(z, \hat{\theta}^p) \leq \overline{\epsilon^p}_1(z, \hat{\theta}^p) \leq 12\epsilon_1(z, \hat{\theta}^p)$ with probability $1 - 2\rho$ where ϵ_1 is the expression from Theorem 6.*

Constant approximation with low privacy cost. This theorem suggests that if we choose $\sigma_2 \asymp \sigma$ and use a slightly large λ in ObjPert we could obtain *constant multiplicative approximation* of the per-instance privacy loss for all individuals, with only a constant blow-up in both the DP and pDP losses. Moreover, while using a large λ may appear to introduce additional bias, the required choice of $\lambda \asymp \sqrt{d}\sigma$ is actually exactly the choice to obtain the minimax rate in general convex private ERM (Bassily et al., 2014). We discuss a more adaptive algorithm in the appendix that adapts based on a well-conditioned H matrix that avoids using a large λ and achieves a stronger approximation.

Joint DP interpretation. Finally, we can also interpret our results from a joint-DP perspective (Kearns et al., 2014). Given any realized output $\tilde{\theta}^p \in \mathbb{R}^d$, the tuple of $\{\tilde{\epsilon}(z_1, \hat{\theta}^p), ..., \tilde{\epsilon}(z_1, \hat{\theta}^p)\}$ satisfies joint-DP and joint-pDP with the same ϵ parameter as in Theorem 10.

[2]Instead of adding "analyze-gauss" noise, we sample from the Gaussian Orthogonal Ensemble (GOE) distribution to obtain a random matrix (Appendix A). Under this model we show that τ is on the order of $O(\sqrt{d}(1 + \log(C/\rho)^{3/2}))$.

4 Experiments

We evaluate our methods to release the pDP losses using both real-word and synthetic data, focusing for now on logistic regression.

4.1 True pDP loss ϵ_1 vs. released bound $\tilde{\epsilon_1}$

In this experiment we use a synthetic dataset, sampling $\theta \sim \mathcal{N}(0, 1)$, sampling $X \in \mathbb{R}^{n \times d}$ from the d-dimensional unit ball. Y is then a deterministic function of X and θ such that $\Pr[Y = 1] = \texttt{sigmoid}(X^T\theta) + E$ for noise matrix E. Figure 2 plots the pDP distributions for the true ϵ_1 and its data-independent bound $\tilde{\epsilon}$ calculated according to Algorithm 2, demonstrating both that the true *ex-post* pDP loss is an order of magnitude improvement compared to the worst case DP bound ϵ, and that the data-independent bound provides a good approximation to the true pDP loss.

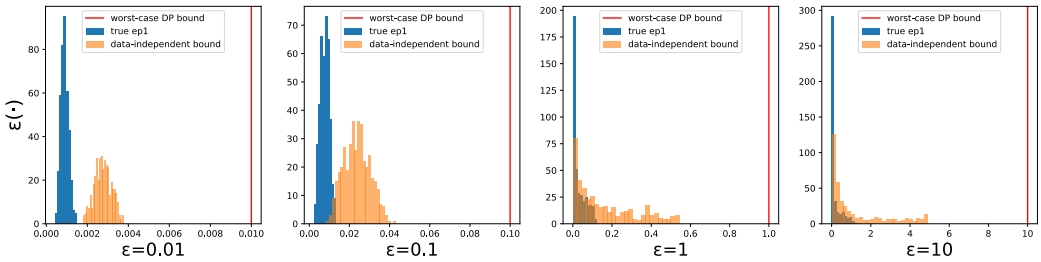

Figure 2: pDP distributions for the true ϵ_1 and its data-independent bound.

4.2 Dimension

We generate synthetic data as described in the previous experiment, this time varying $d \in [1, 5, 10, 20, 30, 40, 50, 60]$. We choose $\epsilon = 0.5$ as the DP bound and $\delta = 10^{-3}$, and set σ, λ according to Theorem 5. Figure 3 plots the true worst-, best- and median-case *ex-post* pDP loss over all individuals z against its data-independent counterpart (i.e., the worst-case pDP loss is calculated as $\max_{z \in D} \epsilon(\cdot, D_{\pm z})$ and the worst-case data-independent bound as $\max_{z \in D} \tilde{\epsilon}(\cdot, z)$).

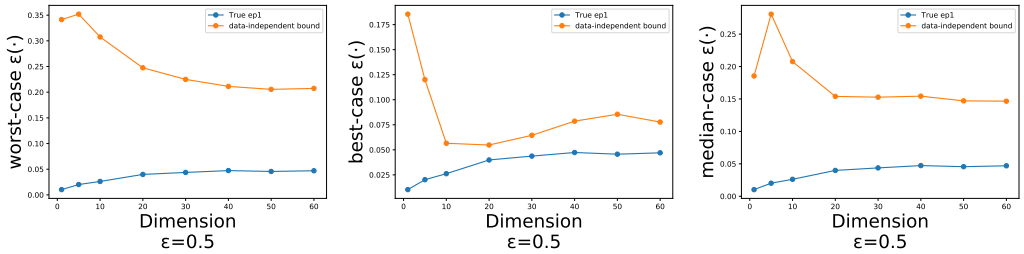

Figure 3: pDP losses across varying data dimensions.

5 Conclusion

In this paper we show how to privately release the personalized privacy losses incurred by the objective perturbation mechanism. We present both a simple data-independent bound on the pDP losses, as well as a data-dependent approach which provides stronger per-instance privacy guarantees under certain regimes. Our theoretical and experimental results show that our methods provide strong, adaptive privacy guarantees and that we can release privatized pDP losses which are good approximations of the true data-dependent pDP losses.

Our framework applies to a wide range of learning problems, but our pDP analysis limits us to unconstrained optimization. In future work we will explore how to extend our approach to optimization over any convex domain.

References

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Dwork, C. and Lei, J. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 371–380, 2009.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014a.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, 2014b.

Feldman, V. and Zrnic, T. Individual privacy accounting via a renyi filter. *arXiv preprint arXiv:2008.11193*, 2020.

Kearns, M., Pai, M., Roth, A., and Ullman, J. Mechanism design in large games: Incentives and privacy. In *Conference on Innovations in theoretical computer science (ITCS-14)*, pp. 403–410, 2014.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1. JMLR Workshop and Conference Proceedings, 2012.

Ligett, K., Neel, S., Roth, A., Waggoner, B., and Wu, Z. S. Accuracy first: Selecting a differential privacy level for accuracy-constrained erm. *Advances in Neural Information Processing Systems*, 2017:2567–2577, 2017.

Narayanan, A. and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125. IEEE, 2008.

Nissim, K., Raskhodnikova, S., and Smith, A. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 75–84, 2007.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Úlfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations (ICLR-18)*, 2018.

Tang, J., Korolova, A., Bai, X., Wang, X., and Wang, X. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.

Triastcyn, A. and Faltings, B. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning*, pp. 9583–9592. PMLR, 2020.

Wang, Y.-X. Per-instance differential privacy. *Journal of Privacy and Confidentiality*, 9(1), 2019.

Wei, B.-C., Hu, Y.-Q., and Fung, W.-K. Generalized leverage and its applications. *Scandinavian Journal of statistics*, 25(1):25–37, 1998.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]