Journal of Privacy and Confidentiality
Vol. 9 (1) 2019
TPDP 2016
Submitted March 2018
Published March 2019

DIFFERENTIALLY PRIVATE CONFIDENCE INTERVALS FOR EMPIRICAL RISK MINIMIZATION

YUE WANG, DANIEL KIFER, AND JAEWOO LEE

Department of Computer Science and Engineering; The Pennsylvania State University
e-mail address: yuw140@cse.psu.edu

Department of Computer Science and Engineering; The Pennsylvania State University
e-mail address: dkifer@cse.psu.edu

Department of Computer Science; University of Georgia
e-mail address: jwlee@cs.uga.edu

Abstract. The process of data mining with differential privacy produces results that are affected by two types of noise: sampling noise due to data collection and privacy noise that is designed to prevent the reconstruction of sensitive information. In this paper, we consider the problem of designing confidence intervals for the parameters of a variety of differentially private machine learning models. The algorithms can provide confidence intervals that satisfy differential privacy (as well as the more recently proposed concentrated differential privacy) and can be used with existing differentially private mechanisms that train models using objective perturbation and output perturbation.

1. Introduction

Differential privacy [Dwork et al., 2006b] is now seen as a gold standard for protecting individual data records while releasing aggregate information such as noisy count queries or parameters of data mining models. There has been a great deal of focus on answering queries and building models using differential privacy but much less focus on empirically understanding their uncertainty, for example, through the length of confidence intervals for parameters or query answers. Uncertainty estimates are needed by data users to understand how much they can trust a data mining model.

Uncertainty comes from two sources: uncertainty in the data and uncertainty due to privacy mechanisms. Uncertainty in the data is often referred to as sampling error – the data are a sample from a larger population (so a different sample could yield different results). Uncertainty due to privacy mechanisms comes from the fact that any useful algorithm that satisfies differential privacy must have randomized behavior. Both must be quantified in an uncertainty estimate.

In the setting we consider, a differentially private algorithm has trained a model and released its parameters. The end user would like to obtain confidence intervals around each parameter. These confidence intervals themselves must satisfy differential privacy. There has been very little work on this topic and, to the best of our knowledge, all of it has focused on linear regression [Sheffet, 2017, Barrientos et al., 2017].

Key words and phrases: Differential Privacy, Objective Perturbation, Output Perturbation, Confidence Intervals.
Supported by NSF grants 1702760 and 1228669.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.660

On the other hand, differentially private model fitting algorithms such as objective perturbation [Chaudhuri et al., 2011] and output perturbation [Chaudhuri et al., 2011] can train a variety of models, such as logistic regression and SVM, and achieve state-of-the-art (or near state-of-the-art) accuracy on many datasets. However, they do not come with confidence intervals.

In this paper, we propose privacy-preserving algorithms for generating confidence intervals for differentially private models trained by the techniques of Chaudhuri et al. [2011]. We provide versions of these algorithms for pure ϵ-differential privacy, as well as the recently introduced concentrated differential privacy (zCDP) [Bun and Steinke, 2016].

There are three basic steps in our framework. The first is to use either the output or objective perturbation techniques [Chaudhuri et al., 2011] to provide model parameters. In the case of objective perturbation, the result satisfies both ϵ-differential privacy as well as $\frac{\epsilon^2}{2}$-zCDP. In the case of output perturbation, the algorithms for differentially privacy and zCDP (concentrated differential privacy) are different. In the second step, we use Taylor's Theorem and the Central Limit Theorem to approximate the randomness in the model coefficients that is due to both the data and the privacy mechanisms. This approximation relies on properties of the data and thus necessitates a third step of estimating them using either differential privacy or zCDP. Thus, the overall privacy budget must be split into two phases: the budget allocated to getting the model parameters and the budget allocated to estimating uncertainty in the parameters.

In our experiments, we verify the accuracy of our confidence intervals and observe that under pure differential privacy, the confidence intervals for models trained with objective perturbation are shorter than those for models trained with output perturbation. However, under concentrated differential privacy, the confidence intervals for output perturbation are much smaller.

Note that the goal of this paper is not to introduce new model fitting algorithms. The goal is to add capabilities for quantifying uncertainty in the model coefficients.

To summarize, our contributions are the following.

- To the best of our knowledge, this is the first paper that provides differentially private confidence intervals for models other than linear regression and our work is *not* limited to any specific model – it works for any model that can be trained using objective perturbation [Chaudhuri et al., 2011].
- The confidence intervals can be made to satisfy different variations of differential privacy, including pure ϵ-differential privacy [Dwork et al., 2006b], zero-mean concentrated differential privacy (zCDP) [Bun and Steinke, 2016], and approximate (ϵ, δ)-differential privacy [Dwork et al., 2006a].
- We empirically validate our confidence intervals using a variety of public datasets.
- Finally, we provide a small improvement to the original objective perturbation model fitting technique [Chaudhuri et al., 2011] by improving some of the constants in the algorithm.

We discuss related work in Section 2 and introduce the preliminaries and notation in Section 3. We derive confidence intervals for models trained with objective perturbation in Section 4. We derive confidence intervals for models trained with output perturbation in Section 5. We show how to apply our algorithms to logistic regression and support vector machines in Section 6 and present experiments in Section 7. We present conclusions and open problems in Section 8.

2. Related Work

Differentially private training of data mining models has been extensively studied, for example, in Chaudhuri et al. [2011], Friedman and Schuster [2010], Kifer et al. [2012], Yu et al. [2014b], Zhang et al. [2012], Wu et al. [2015], Bassily et al. [2014], Kasiviswanathan and Jin [2016], Zhang et al. [2013], Jain and Thakurta [2013], Wang et al. [2017], Zhang et al. [2017], Rubinstein et al. [2009], Talwar et al. [2014, 2015], Jain and Thakurta [2014], Kasiviswanathan et al. [2017], Ligett et al. [2017], Wang et al. [2018]. However, such work provides model parameters without any uncertainty

estimates (such as confidence intervals) about the parameters. To the best of our knowledge, the only exceptions are for linear regression [Sheffet, 2017, Barrientos et al., 2017].

Chaudhuri et al. [2011] studied a general class of models that (without privacy) are trained with empirical risk minimization. They proposed two general approaches, called *objective perturbation* and *output perturbation* for training such models with differential privacy. Subsequent work increased the set of models that can be trained [Kifer et al., 2012, Yu et al., 2014b, Zhang et al., 2012, 2013, Wu et al., 2015]. Kifer et al. [2012] extended the algorithm of Chaudhuri et al. [2011] by removing some differentiability requirements and allowing constraints in model training. Yu et al. [2014b] solved the problem of differentially private penalized logistic regression with elastic-net regularization by extending the objective perturbation technique to any convex penalty function. Zhang et al. [2012] proposed the functional mechanism, which approximates models by polynomials. Subsequently, Zhang et al. [2013] proposed a general solution based on genetic algorithms and a novel random perturbation technique called the enhanced exponential mechanism. Wu et al. [2015] proposed another output perturbation technique for learning tasks with convex and Lipschitz loss functions on a bounded domain. They relaxed the condition of differentiable loss functions in Chaudhuri et al. [2011]. However, we found that when both methods are applicable, the noise added by the output perturbation technique of Wu et al. [2015] is generally larger than the noise added by the output perturbation technique of Chaudhuri et al. [2011].

High dimensional regression problems were also studied in Bassily et al. [2014], Kasiviswanathan and Jin [2016]. Bassily et al. [2014] proposed new algorithms for the private convex ERM problem when the loss function is only Lipschitz and the domain of the optimization is bounded. They also proposed separate algorithms when the loss function is also strongly convex. They propose algorithms for both pure and approximate differential privacy. Kasiviswanathan and Jin [2016] improved the worst-case risk bounds of Bassily et al. [2014] under differential privacy with access to full data. Moreover, with access to only the projected data and the projection matrix, they derived the excess risk bounds for generalized linear loss functions.

There has been some work on quantifying the uncertainty for differentially private models, mostly in the form of confidence intervals and hypothesis testing.

Differentially private hypothesis testing has been studied in Uhler et al. [2013], Yu et al. [2014a], Wang et al. [2015], Gaboardi et al. [2016], Rogers and Kifer [2017], Kakizaki et al. [2017], Cai et al. [2017], Acharya et al. [2017]. Uhler et al. [2013] and Yu et al. [2014a] conducted differentially private independence testing through χ^2-tests with output perturbation, and adjusted the asymptotic distribution used to compute p-values. Using input perturbation, Wang et al. [2015], and later independently Gaboardi et al. [2016] proposed differentially private hypothesis testing for independence and goodness of fit. Rogers and Kifer [2017] later proposed new test statistics for chi-squared testing that are more compatible with privacy noise. Kakizaki et al. [2017] proposed the unit circle mechanism for independence testing on 2×2 tables with known marginal sums. Cai et al. [2017] studied the sample complexity to conduct differentially private goodness of fit test with guaranteed type I and II errors. Later work by Acharya et al. [2017] derived the upper and lower bounds on the sample complexity for goodness of fit and closeness testing under (ϵ, δ)-differential privacy.

Providing diagnostics for differentially private regression analysis was studied in Chen et al. [2016], where Chen et al. designed differentially private algorithms to construct residual plots for linear regression and receiver operating characteristics (ROC) curves for logistic regression.

Work on differentially private confidence intervals includes D'Orazio et al. [2015], Sheffet [2017], Karwa and Vadhan [2017]. D'Orazio et al. [2015] and Karwa and Vadhan [2017] did not study models, instead they constructed differentially private confidence intervals for a mean [Karwa and Vadhan, 2017] and the difference of two means [D'Orazio et al., 2015]. In the context of model coefficients, Sheffet [2017] studied (ϵ, δ)-differentially private Ordinary Least Squares Regression (OLS) and generated confidence intervals for the parameters. Barrientos et al. [2017] used differential

privacy to quantify the uncertainty of the coefficients of differentially private linear regression models. They generated differentially private t statistics for each coefficient.

Thus the closest work related to ours is Sheffet [2017] and Barrientos et al. [2017]. While their work only targets linear regression, our work targets any models that can be trained under the objective perturbation and output perturbation techniques of Chaudhuri et al. [2011], which include many models such as logistic regression and SVM, but exclude linear regression.

To obtain confidence intervals, we also need to privately estimate second order matrices from the data. Perturbing second order matrices for data are common in privacy-preserving principal component analysis (PCA). Chaudhuri et al. proposed to perturb the second order matrices with the exponential mechanism to achieve differential privacy in Chaudhuri et al. [2012]. With the SuLQ framework [Blum et al., 2005], Blum et al. added Gaussian noise to the second moment matrix and used it in the PCA to protect a notion of (ϵ, δ, T)-Privacy. Jiang et al. [2016] studied the problem of publishing differentially private second order matrices by adding proper Laplace or Wishart noise. Dwork et al. worked on projecting the second moment matrix of data into the low dimensional space using the notion of approximate differential privacy in Dwork et al. [2014]. Later in Sheffet [2015], Sheffet also discussed three techniques to get the second moment matrix while preserving the approximate differential privacy, with the matrices being positive-definite.

Due to the structure of the matrices needed by our techniques, a spherical version of the Laplace Mechanism, introduced in the objective perturbation method [Chaudhuri et al., 2011] to achieve differential privacy, or the Gaussian Mechanism [Bun and Steinke, 2016] to achieve zero-mean concentrated differential privacy [Bun and Steinke, 2016], are most appropriate.

3. Preliminaries and Notation

In this section, we introduce notation used in the paper and then review the background of differential privacy and its variants, empirical risk minimization, and its applications to logistic regression and support vector machines.

Let $\mathcal{D} = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$ be a set of n records. Each record i has a d-dimensional vector $\vec{x}_i$ of real numbers known as a *feature vector* and each $y_i \in \{-1, 1\}$ is called the *target*. Following Chaudhuri et al. [2011], we require that each record is normalized so that $||\vec{x}_i||_2 = 1$.

3.1. Differential Privacy.

Definition 1. *(Differential Privacy [Dwork et al., 2006b]). Given an $\epsilon > 0$ and $\delta \geq 0$, a randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy if for all pairs of databases $\mathcal{D}, \mathcal{D}'$ differing on the value of a record, and all $V \subseteq \text{range}(\mathcal{M})$,*

$$Pr(\mathcal{M}(\mathcal{D}) \in V) \leq e^{\epsilon}\ Pr(\mathcal{M}(\mathcal{D}') \in V) + \delta.$$

When $\delta = 0$, we refer to it as both ϵ-differential privacy and pure differential privacy. When $\delta > 0$, we refer to it as both (ϵ, δ)-differential privacy and approximate differential privacy. Another relaxation of differential privacy is known as zero-mean concentrated differential privacy, or ρ-zCDP for short. It relies on the concept of α-Rényi Divergence, which is defined as follows.

Definition 2. *(Rényi Divergence [Rényi, 1961]). Let P and Q be probability distributions defined on the domain Ω, then for $\alpha \in (1, \infty)$, the α-Rényi Divergence between P and Q is defined as*

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \left(\int_\Omega P(z)^\alpha Q(z)^{(1-\alpha)}\, \mathrm{d}z \right).$$

Definition 3. *(Zero-Concentrated Differential Privacy (zCDP) [Bun and Steinke, 2016]). A randomized mechanism $\mathcal{M}$ satisfies ρ-zero-concentrated differential privacy (i.e., ρ-zCDP) if for all pairs of databases $\mathcal{D}$ and $\mathcal{D}'$ that differ on the value of a single record and all $\alpha \in (1, \infty)$,*

$$D_\alpha(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) \leq \rho\alpha,$$

where $D_\alpha(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}'))$ is the α-Rényi divergence between the distribution of $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$.

ρ-zCDP is weaker than pure differential privacy and stronger than approximate differential privacy. The following results make the relations between them precise.

Proposition 1. *[Bun and Steinke, 2016]. If $\mathcal{M}$ satisfies ϵ-differential privacy, then $\mathcal{M}$ satisfies $(\epsilon^2/2)$-zCDP.*

Proposition 2. *[Bun and Steinke, 2016]. If $\mathcal{M}$ satisfies ρ-zCDP then it satisfies $(\rho+2\sqrt{\rho \log(1/\delta)}, \delta)$-differential privacy.*

Thus, we only focus on pure differential privacy and ρ-zCDP in this paper. All ρ-zCDP algorithms can be converted into algorithms for approximate differential privacy using Proposition 2.

The algorithms studied in this paper rely on the concept of L_2 sensitivity:

Definition 4. *(L_2-Sensitivity [Chaudhuri et al., 2011, Bun and Steinke, 2016]). The L_2-sensitivity for a (scalar- or vector-valued) function f is*

$$\Delta_2(f) = \max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$$

for all pairs of databases $\mathcal{D}, \mathcal{D}' \in \text{domain}(f)$ differing on the value of at most one entry.

For example, the L_2 sensitivity is used to set the variance of the Gaussian Mechanism for ρ-zCDP.

Proposition 3. *(Gaussian Mechanism [Bun and Steinke, 2016]). Let f be a vector-valued function (whose output is a vector of dimension d) with L_2 sensitivity $\Delta_2(f)$. Let $\sigma = \Delta_2(f)/\sqrt{2\rho}$. The Gaussian Mechanism, which outputs $f(\mathcal{D}) + N(\vec{0}, \sigma^2\mathbf{I}_d)$, satisfies ρ-zCDP.*

Both differential privacy and ρ-zCDP are invariant under post-processing [Dwork et al., 2006b, Bun and Steinke, 2016]. That is, if a mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (resp., ρ-zCDP), and if A is any algorithm whose input is the output of $\mathcal{M}$, then the composite algorithm, which first runs $\mathcal{M}$ on the input data and then runs A on the result satisfies ϵ-differential privacy (resp., ρ-zCDP).

Another useful property of these definitions is *composition*, which allows the privacy parameter of a complicated algorithm be derived from the privacy parameters of its sub-components.

Proposition 4. *(Composition [Dwork et al., 2006b, Bun and Steinke, 2016]). Let $\mathcal{M}$ be a randomized mechanism that satisfies ϵ-differential privacy (resp., ρ-zCDP) and $\mathcal{M}'$ be a randomized mechanism that satisfies ϵ'-differential privacy (resp., ρ'-zCDP). Then the composite algorithm $\mathcal{M}^*$ that, on input $\mathcal{D}$ outputs the tuple $(\mathcal{M}(\mathcal{D}), \mathcal{M}'(\mathcal{D}))$ satisfies $(\epsilon + \epsilon')$-differential privacy (resp., $(\rho + \rho')$-zCDP).*

3.2. **Empirical Risk Minimization.** Empirical risk minimization is a common way of training machine learning models. There is an assumption that the dataset $\mathcal{D} = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$ is independently sampled from some unknown distribution F_0. In this setting, the model has a parameter vector θ and a prediction function g. Its prediction for y is $g(\vec{x}, \theta)$.

To train the model, in the setting assumed by Chaudhuri et al. [2011], one specifies a loss function in the form of $f(\vec{x}, y, \theta) = f(y\theta^T \cdot \vec{x})$, and finds the θ that minimizes the empirical risk:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left[f(\vec{x}_i, y_i, \theta) + c\|\theta\|_2^2 \right]. \tag{3.1}$$

To satisfy differential privacy, Chaudhuri et al. [2011], proposed the *objective perturbation technique* to add noise to the objective function and then produce minimizer of the perturbed objective:

$$\arg\min_{\theta} J_n(\theta, \mathcal{D}) = \arg\min_{\theta} \left[L_n(\theta, \mathcal{D}) + \frac{1}{n}\beta^T\theta \right] \equiv \arg\min_{\theta} \left(\frac{1}{n} \sum_{i=1}^{n} \left[f(\vec{x}_i, y_i, \theta) + c||\theta||_2^2 \right] + \frac{1}{n}\beta^T\theta \right),$$

where β is a zero-mean random variable with density

$$\mathrm{v}(\beta) = \frac{1}{u} e^{-\gamma\|\beta\|_2}, \tag{3.2}$$

where u is the normalizing constant, and γ depends on the privacy budget and the L_2-sensitivity of $L_n(\cdot)$.

Their proof of privacy depends on the concept of strong convexity:

Definition 5. *(Strong-Convexity). A function $f(\theta)$ over $\theta \in \mathbb{R}^d$ is said to be λ-strongly convex if for all $\alpha \in (0, 1)$, θ and η,*

$$f(\alpha\theta + (1-\alpha)\eta) \leq \alpha f(\theta) + (1-\alpha) f(\eta) - \frac{1}{2}\lambda\alpha(1-\alpha)\|\theta - \eta\|_2^2.$$

3.3. **Logistic Regression and SVM.** In the paper, we will work with the applications of logistic regression and support vector machines (SVM)[1]. In logistic regression, the goal is to predict $P(y = 1 \mid \vec{x})$ and this is done by modeling $P(y = 1 \mid \vec{x}) = S(\theta \cdot \vec{x})$, where S is the sigmoid function:

$$S(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}.$$

Logistic regression is trained in the ERM framework using the loss function

$$f(\vec{x}, y, \theta) = \log\left[1 + \exp(-y\theta \cdot \vec{x})\right].$$

In support vector machines, the prediction for y is 1 if $\theta \cdot \vec{x} \geq 0$ and is -1 otherwise. To train it in the ERM framework, we will use the Huberized hinge Loss [Chapelle, 2007], defined as follows:

$$f_{Huber}(\vec{x}, y, \theta) = \begin{cases} 0 & \text{if } z > 1 + h \\ \frac{1}{4h}(1 + h - z)^2 & \text{if } |1 - z| \leq h \\ 1 - z & \text{if } z < 1 - h, \end{cases}$$

where $z = y\theta \cdot \vec{x}$ and where h is a fixed constant [Chapelle, 2007].

4. Confidence Intervals for Objective Perturbation

In this section, we show how to obtain confidence intervals for models trained by objective perturbation [Chaudhuri et al., 2011]. For completeness, we present a slightly improved version of the algorithm in Section 4.1 and then derive the confidence interval algorithm in Sections 4.2, 4.3, and 4.4.

[1]We use these two applications as examples, but our algorithms are not restricted to them.

4.1. **Objective Perturbation.** The objective perturbation algorithm modifies the ERM framework by randomly drawing a noise vector β from a spherical analogue of the Laplace distribution (see Equation 3.2). Then, instead of minimizing the original ERM objective $\frac{1}{n}\sum_{i=1}^{n}\left[f(\vec{x}_i, y_i, \theta) + c||\theta||_2^2\right]$, it modifies it by adding $\frac{1}{n}\beta^T\theta$ and then minimizes it with respect to θ. The version of the techniques shown in Algorithm 1 slightly differs from the original [Chaudhuri et al., 2011] in the first line, allowing it to use less noise.

Algorithm 1: Objective Perturbation

input : Data $\mathcal{D} = \{(\vec{x}_i, y_i)\}_{i=1}^n$, privacy budget ϵ, loss function f with $|f''(\cdot)| \leq t$, coefficient c with $c > \frac{t}{2n(e^\epsilon - 1)}$

$\epsilon' \leftarrow \epsilon - \log\left(1 + \frac{t}{2nc}\right)$

Sample a d-dimensional vector β with density from Equation 3.2 with $\gamma = \epsilon'/2$

$\tilde{\theta} \leftarrow \arg\min_\theta \left(\frac{1}{n}\sum_{i=1}^{n}\left[f(\vec{x}_i, y_i, \theta) + c||\theta||_2^2\right] + \frac{1}{n}\beta^T\theta\right)$

Output $\tilde{\theta}$

Theorem 1 . *If the loss function $f(\cdot)$ is convex and doubly differentiable, with $|f'(\cdot)| \leq 1$ and $|f''(\cdot)| \leq t$, then Algorithm 1 satisfies ϵ-differential privacy whenever all the feature vectors $\vec{x}_i$ have $||\vec{x}_i||_2 \leq 1$.*

The proof of Theorem 1 is in Appendix A.1.

In order to achieve ρ-zCDP, we use Proposition 1 to conclude that the algorithm satisfies $\frac{\epsilon^2}{2}$-zCDP.

4.2. **Confidence Intervals.** In this section, we describe one of our main contributions – the construction of confidence intervals for objective perturbation. Set $J_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[f(\vec{x}_i, y_i, \theta) + c||\theta||_2^2\right] + \frac{1}{n}\beta^T\theta$.

Let $\tilde{\theta}$ be the privacy preserving parameters output by the objective perturbation algorithm. Let θ_0 be the non-private solution we would get if we had infinite data (i.e. the true parameter vector of the distribution from which data is sampled). Since the noise in Algorithm 1 is divided by n, then θ_0 is also the privacy-preserving solution one would obtain with infinite data and $E[\nabla J_n(\theta_0, \mathcal{D})] = \vec{0}$, where the expectation is taken over the data and β (note that β has $\vec{0}$ mean).

Expanding the Taylor series of ∇J_n around $\tilde{\theta}$ and noting that the gradient of J_n at $\tilde{\theta}$ is 0 by construction (since $\tilde{\theta}$ minimizes J_n), we have

$$\nabla J_n(\theta_0) \approx \nabla J_n(\tilde{\theta}) + H[J_n(\tilde{\theta})](\theta_0 - \tilde{\theta}) = H[J_n(\tilde{\theta})](\theta_0 - \tilde{\theta}),$$

where $H[J_n(\tilde{\theta})]$ is the Hessian (matrix of second derivatives) of J_n evaluated at $\tilde{\theta}$.

Now, $\nabla J_n(\theta_0)$ is equal to $\frac{1}{n}\beta$ plus the average n terms – one for each $\vec{x}_i$. This means that by the Central Limit Theorem, $\sqrt{n}\nabla J_n(\theta_0)$ can be approximated by the sum of $\frac{1}{\sqrt{n}}\beta$ and $N(\vec{0}, \Sigma)$, where $N(\vec{0}, \Sigma)$ is a zero-mean Gaussian with covariance matrix:

$$\Sigma = E\left[\left(\nabla\left(f(\vec{x}, y, \theta_0) + c||\theta_0||_2^2\right)\right)\left(\nabla\left(f(\vec{x}, y, \theta_0) + c||\theta_0||_2^2\right)\right)^T\right].$$

If the Hessian and covariance matrices were known, we could combine the two approximations for $\nabla J_n(\theta_0)$ as follows. Let G be a random variable with distribution $N(\vec{0}, \Sigma)$. Let Q be an

independent random variable having the same distribution as $G + \beta/\sqrt{n}$. Then we can approximate the distribution of $\sqrt{n}H[J_n(\tilde{\theta})](\theta_0 - \tilde{\theta})$ with the distribution of Q.

However, since the Hessian and the covariance matrix are unknown, we will need to obtain privacy preserving estimates $\tilde{H}$ and $\tilde{\Sigma}$. Now let $\tilde{G}$ be a random variable having distribution $N(\vec{0}, \tilde{\Sigma})$ and let $\tilde{Q}$ be an independent random variable having the same distribution as $\tilde{G} + \beta/\sqrt{n}$. Substituting in $\tilde{G}$ for G and $\tilde{H}$ for H, we now approximate the distribution of $\sqrt{n}\tilde{H}[J_n(\tilde{\theta})](\theta_0 - \tilde{\theta})$ with the distribution of $\tilde{Q}$. Multiplying by $\tilde{H}^{-1}$, we get the following approximation:

$$\text{Distribution of } \theta_0 - \tilde{\theta} \;\approx\; \text{Distribution of } \tilde{H}[J_n(\tilde{\theta})]^{-1}\tilde{Q}/\sqrt{n}. \tag{4.1}$$

We note that the right hand side of Equation 4.1 is easy to sample from. We next discuss how to estimate the Hessian and covariance matrix and use them with Equation 4.1 to produce confidence intervals for each component of θ_0.

4.3. **Computations of the Hessian and Covariance Matrix.** If privacy was not a concern, the Hessian would be computed as:

$$H[J_n(\theta)] = \frac{1}{n}\sum_{i=1}^{n} H[f(\vec{x}_i, y_i, \theta)] + 2c\mathbf{I}, \tag{4.2}$$

and the covariance matrix Σ would be estimated as:

$$\begin{aligned}
\Sigma =& E\left[\left(\nabla\Big(f(\vec{x}, y, \theta_0) + c||\theta_0||_2^2\Big)\right)\left(\nabla\Big(f(\vec{x}, y, \theta_0) + c||\theta_0||_2^2\Big)\right)^T\right] \\
=& E\left[\left(\nabla f(\vec{x}, y, \theta_0) + 2c\theta_0\right)\left(\nabla f(\vec{x}, y, \theta_0) + 2c\theta_0\right)^T\right] \\
=& E\left\{\nabla(f(\vec{x}, y, \theta_0))[\nabla f(\vec{x}, y, \theta_0)]^T\right\} + 2cE[\nabla f(\vec{x}, y, \theta_0)]\theta_0^T + 2c\theta_0 E[\nabla f(\vec{x}, y, \theta_0)]^T + 4c^2\theta_0\theta_0^T \\
=& E\left\{\nabla(f(\vec{x}, y, \theta_0))[\nabla f(\vec{x}, y, \theta_0)]^T\right\} - 4c^2\theta_0\theta_0^T \\
\approx& \frac{1}{n}\sum_{i=1}^{n} \nabla f(\vec{x}_i, y_i, \tilde{\theta})[\nabla f(\vec{x}_i, y_i, \tilde{\theta})]^T - 4c^2\tilde{\theta}\tilde{\theta}^T,
\end{aligned} \tag{4.3}$$

where the second-to-last step is obtained from the fact that $E[\nabla J_n(\theta_0)] = \vec{0}$ from which it follows that $E[\nabla f(\vec{x}, y, \theta_0)] + 2c\theta_0 = \vec{0}$.

However, since privacy is indeed a concern, we need to obtain estimates of the Hessian and covariance matrix using either ϵ-differential privacy or ρ-zCDP. The same algorithm works for both matrices and is shown in Algorithm 2.

The algorithm takes the matrix M, which is either the Hessian (computed as in Equation 4.2) or the covariance matrix (computed as in Equation 4.3). It also takes the L_2 sensitivity of these matrices (we show how to compute the sensitivities for logistic regression and SVM in Section 6). It uses the L_2 sensitivity to determine the variance of the noise that must be added. The distribution of this noise depends on whether we want to use pure differential privacy or zCDP.

These resulting noisy matrices might not be symmetric positive-semidefinite (even though the Hessian and covariance matrices must have those properties). Thus we add a postprocessing step to make the matrix symmetric and have all eigenvalues at least $2c$.

Lemma 1. *Algorithm 2 satisfies ϕ-differential privacy and ϕ-zCDP.*

The proof of Lemma 1 is in Appendix A.2.

Algorithm 2: Private Symmetric Positive Definite Matrix (PrivSPDMat)

input : Matrix $M \in \mathbb{R}_{d\times d}$, L_2 sensitivity $Sens(M)$, privacy budget ϕ, parameter c

if *Requiring ϵ-differential privacy with $\epsilon = \phi$* **then**

  Sample a d^2-dimensional vector η with density from Equation 3.2 with $\gamma = \frac{\phi}{Sens(M)}$

else

  `// for ρ-zCDP with ρ = φ`

  Sample a noise vector η from $N\left(\vec{0}, \frac{Sens(M)^2}{2\phi}\mathbf{I}_d\right)$

end

Reshape η to a $d \times d$ matrix $\mathrm{mat}(\eta)$

$\tilde{M} \leftarrow M + \mathrm{mat}(\eta)$

$\tilde{M} \leftarrow (\tilde{M} + \tilde{M}^T)/2$

Let $V \,\mathrm{diag}(\Lambda) = \tilde{M}V$ be the eigen-decomposition for $\tilde{M}$

`// columns of V are orthonormal eigenvectors`

for $i \leftarrow 1$ **to** d **do**

  $\Lambda[i] \leftarrow \max(\Lambda[i], 2c)$

end

$\tilde{M} \leftarrow V \,\mathrm{diag}(\Lambda) V^T$

Output $\tilde{M}$

4.4. **Putting It All Together: Confidence Intervals Generation.** The overall algorithm is shown in Algorithm 3. It first splits the privacy budget into 3 pieces ϕ_1, ϕ_2, ϕ_3. Using privacy budget ϕ_1, it runs the objective perturbation algorithm to provide privacy-preserving model parameters $\tilde{\theta}$. Privacy budget ϕ_2 is used to provide a privacy-preserving estimate of the Hessian $\tilde{H}$ and privacy budget ϕ_3 is used to provide a privacy preserving estimate of the covariance matrix $\tilde{\Sigma}$. Once these quantities are obtained, it can use Equation 4.1. This equation says that the distribution of $\theta_0 - \tilde{\theta}$ can be approximated by sampling $\tilde{G}$ from $N(\vec{0}, \tilde{\Sigma})$, $\tilde{\beta}$ from Equation 3.2, computing $\tilde{Q}$ from $\tilde{G} + \tilde{\beta}/\sqrt{n}$, and then plugging $\tilde{Q}$ into Equation 4.1. By obtaining many such samples $z_1, \ldots, z_m$ where each z_i is a d-dimensional vector (because θ_0 and $\tilde{\theta}$ are d-dimensional), for each dimension j we take an interval (a_j, b_j) that covers $1-\alpha$ (e.g., 95%) of the $z_i[j]$. Then the estimated confidence interval for $\theta_0[j]$ is $(\tilde{\theta}[j] + a_j, \tilde{\theta}[j] + b_j)$. Note that this sampling step is strict postprocessing – never accesses the original data and it only uses privacy preserving estimates from the previous steps.

Theorem 2 . *Under the conditions of Theorem 1, Algorithm 3 satisfies $(\phi_1 + \phi_2 + \phi_3)$-differential privacy and $(\phi_1^2/2 + \phi_2 + \phi_3)$-zCDP.*

The proof of Theorem 2 is in Appendix A.3.

5. Confidence Intervals for Output Perturbation

In this section, we provide confidence intervals for model parameters learned with output perturbation rather than objective perturbation. Again, we will have algorithms for both differential privacy and zCDP. We will follow similar steps as Section 4 to obtain the intervals.

5.1. **Output Perturbation.** We first review the output perturbation method of Chaudhuri et al. [2011]. Then we will explain how to obtain confidence intervals for the resulting parameters in Section 5.2 (recall that they must account for noise due to the data being a sample as well as noise due to privacy).

Algorithm 3: Private $(1-\alpha)$-Confidence Intervals for θ_0 trained with Objective Perturbation

input : Data $\mathcal{D} = \{(\vec{x}_i, y_i)\}_{i=1}^n$, privacy budgets ϕ_1, ϕ_2 and ϕ_3, parameters c, t, f used by objective perturbation (Algorithm 1), the number of postprocessing samples m to generate, confidence level α

$\tilde{\theta} \leftarrow$ ObjPerturb($\mathcal{D}$, ϕ_1, t, c) `// calling Algorithm 1`

$\epsilon' \leftarrow \epsilon'$ in Algorithm 1

$H[J_n(\tilde{\theta})] \leftarrow \frac{1}{n}\sum_{i=1}^n H[f(\vec{x}_i, y_i, \tilde{\theta})] + 2c\mathbf{I}$

$\tilde{H}[J_n(\tilde{\theta})] \leftarrow$ PrivSPDMat($H[J_n(\tilde{\theta})]$, $Sens(H[J_n(\tilde{\theta})])$, ϕ_2, c) `// calling Algorithm 2`

$\Sigma \leftarrow \frac{1}{n}\sum_{i=1}^n \nabla f(\vec{x}_i, y_i, \tilde{\theta})[\nabla f(\vec{x}_i, y_i, \tilde{\theta})]^T - 4c^2\tilde{\theta}\tilde{\theta}^T$

$\tilde{\Sigma} \leftarrow$ PrivSPDMat(Σ, $Sens(\Sigma)$, ϕ_3, c)

Generate m i.i.d. samples $\tilde{G}_i$ $(i = 1, \ldots, m)$ from $N(\vec{0}, \tilde{\Sigma})$

Generate m i.i.d samples β_i $(i = 1, \ldots, m)$ with density from Equation 3.2 with $\gamma = \epsilon'/2$ (same γ parameter as used in Algorithm 1)

for $i \leftarrow 1$ **to** m **do**

  $\tilde{Q}_i \leftarrow \tilde{G}_i + \beta_i/\sqrt{n}$

  $\theta^{(i)} \leftarrow \tilde{\theta} + \tilde{H}[J_n(\tilde{\theta})]^{-1}\tilde{Q}_i/\sqrt{n}$

end

for $j \leftarrow 1$ **to** d **do**

  $(\theta_L[j], \theta_R[j]) \leftarrow (1-\alpha)$-confidence interval for $\theta^{(1)}[j], \ldots, \theta^{(m)}[j]$

end

Output θ_L, θ_R

In output perturbation, the first step is to compute the non-private parameters $\hat{\theta}$:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}\left[f(\vec{x}_i, y_i, \theta) + c\|\theta\|_2^2\right], \tag{5.1}$$

and then add noise to them [Chaudhuri et al., 2011]. The L_2 sensitivity of $\hat{\theta}$ is $1/(nc)$ [Chaudhuri et al., 2011] and so for ϵ-differential privacy, they release $\hat{\theta} + \beta$, where β has the distribution from Equation 3.2 with parameter $\gamma = nc\epsilon$. To obtain ρ-zCDP one uses the Gaussian Mechanism instead, and samples β from the multivariate normal distribution $N\left(\vec{0}, \frac{1}{2\rho(nc)^2}\mathbf{I}_d\right)$. Algorithm 4 summarizes their output perturbation technique.

Algorithm 4: Output Perturbation (ERMOutput)

input : Data $\mathcal{D} = \{(\vec{x}_i, y_i)\}_{i=1}^n$, privacy budget ϕ, regularization coefficient c.

$\hat{\theta} \leftarrow \arg\min_\theta \frac{1}{n}\sum_{i=1}^n \left[f(\vec{x}_i, y_i, \theta) + c\|\theta\|_2^2\right]$

if *Requiring ϵ-differential privacy with $\epsilon = \phi$* **then**

  Sample a noise vector β with density from Equation 3.2 with $\gamma = nc\phi$

else

  `// for` ρ`-zCDP with` $\rho = \phi$

  Sample a noise vector $\beta \sim N\left(\vec{0}, \frac{1}{2\phi(nc)^2}\mathbf{I}_d\right)$

end

$\tilde{\theta} \leftarrow \hat{\theta} + \beta$

Output $\tilde{\theta}$

Theorem 3 . *([Chaudhuri et al., 2011]) If the loss function $f(\cdot)$ is convex and differentiable with $|f'(\cdot)| \leq 1$, then Algorithm 4 satisfies ϕ-differential privacy and ϕ-zCDP.*

5.2. **Confidence Intervals.** Now we discuss our main contribution in this section, obtaining confidence intervals for the parameters returned by output perturbation. Recall θ_0 is the infinite sample minimizer to $L_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[f(\vec{x}_i, y_i, \theta) + c\|\theta\|_2^2\right]$ (i.e. when $n \to \infty$) while $\hat{\theta}$ is the finite sample minimizer and $\tilde{\theta}$ is the privacy preserving output of Algorithm 4 that we get by using privacy budget ϕ_1, i.e., $\tilde{\theta} = \hat{\theta} + \beta$.

We apply Taylor's theorem around $\hat{\theta}$ to $\nabla L_n(\theta_0)$:

$$\nabla L_n(\theta_0) \approx \nabla L_n(\hat{\theta}) + H[L_n(\hat{\theta})](\theta_0 - \hat{\theta}) = H[L_n(\hat{\theta})](\theta_0 - \hat{\theta}),$$

where $H[L_n(\hat{\theta})]$ is the Hessian of L_n evaluated at $\hat{\theta}$.

As in Section 4.2, by the Central Limit Theorem, $\sqrt{n}\nabla L_n(\theta_0)$ approximately follows a Gaussian distribution $N(\vec{0}, \Sigma)$. Moreover, the formulas for the Hessian $H[L_n(\cdot)]$ (which is equal to $H[J_n(\cdot)]$) and the covariance matrix Σ are the same as Equations 4.2 and 4.3, respectively, from Section 4.2.

Let G be a random variable with distribution $N(\vec{0}, \Sigma)$. Then we can approximate the distribution of $\sqrt{n}H[L_n(\hat{\theta})](\theta_0 - \hat{\theta})$ with the distribution of G.

Again, since the Hessian and the covariance matrix are unknown, we will need to obtain privacy preserving estimates $\tilde{H}$ and $\tilde{\Sigma}$. Now let $\tilde{G}$ be a random variable having distribution $N(\vec{0}, \tilde{\Sigma})$. Substituting in $\tilde{G}$ for G and $\tilde{H}[L_n(\tilde{\theta})]$ for $H[L_n(\hat{\theta})]$, we now approximate the distribution of $\sqrt{n}\tilde{H}[L_n(\tilde{\theta})](\theta_0 - \hat{\theta})$ with the distribution of $\tilde{G}$. That is,

$$\text{Distribution of } \sqrt{n}\tilde{H}[L_n(\tilde{\theta})][\theta_0 - (\tilde{\theta} - \beta)] \approx \text{Distribution of } \tilde{G}.$$

Multiplying by $\tilde{H}^{-1}$ on both sides, and let $\tilde{Q}$ be a random variable having the same distribution as $\tilde{H}[L_n(\tilde{\theta})]^{-1}\tilde{G}/\sqrt{n} - \beta$, we get the following approximation:

$$\text{Distribution of } \theta_0 - \tilde{\theta} \approx \text{Distribution of } \tilde{Q}. \tag{5.2}$$

We note that the right hand side of Equation 5.2 is easy to sample from. This equation also says that the difference between θ_0 and the privacy preserving estimate is approximately the same as the distribution on the right hand side, which only depends on privacy preserving quantities (and not the original data).

For differentially private confidence intervals, as before, we sample many times from the distribution of $\tilde{Q}$ from the right hand side of Equation 5.2 to obtain approximate samples $z_1, \ldots, z_m$ from the distribution of $\theta_0 - \tilde{\theta}$. For each j, we find an interval (a_j, b_j) that contains $(1-\alpha)$ of the $z_i[j]$. Since $\tilde{\theta}$ is a privacy preserving estimate, our privacy preserving confidence interval for $\theta_0[j]$ is $(\tilde{\theta}[j] + a_j, \tilde{\theta}[j] + b_j)$.

On the other hand, if we use Gaussian noise in Algorithm 4, the algorithm for computing confidence intervals becomes much more efficient. In this case $\tilde{Q}$ is the multivariate Gaussian $N(\vec{0}, U)$ where

$$U = \frac{1}{2\phi(nc)^2}\mathbf{I}_d + \frac{1}{n}\tilde{H}[L_n(\tilde{\theta})]^{-1}\tilde{\Sigma}\tilde{H}[L_n(\tilde{\theta})]^{-1},$$

and ϕ is the privacy budget used in Algorithm 4 to perturb $\hat{\theta}$. Therefore we could compute the confidence intervals for θ_0 directly instead of doing Monte Carlo sampling. For each j, we directly compute the confidence interval for $\theta_0[j]$ as

$$\left[\tilde{\theta}[j] - z_{\alpha/2}\sqrt{U_{jj}}\ ,\ \tilde{\theta}[j] + z_{\alpha/2}\sqrt{U_{jj}}\right],$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal distribution. The complete algorithm is shown in Algorithm 5. Note that once we have privacy preserving estimates of $\tilde{\theta}$, $\tilde{H}$, and $\tilde{\Sigma}$ using privacy budgets ϕ_1, ϕ_2, ϕ_3, respectively, everything else is post-processing and thus does not affect the privacy cost.

Algorithm 5: Private $(1-\alpha)$-Confidence Intervals for θ_0 trained with Output Perturbation

input : Data $\mathcal{D} = \{(\vec{x}_i, y_i)\}_{i=1}^n$, privacy budgets ϕ_1, ϕ_2 and ϕ_3, regularization coefficient c, the number of samples m, confidence level α.

$\tilde{\theta} \leftarrow$ ERMOutput($\mathcal{D}$, ϕ_1, c) `// Calling Algorithm 4`

$H[L_n(\tilde{\theta})] \leftarrow \frac{1}{n}\sum_{i=1}^n H[f(\vec{x}_i, y_i, \tilde{\theta})] + 2c\mathbf{I}$

$\tilde{H}[L_n(\tilde{\theta})] \leftarrow$ PrivSPDMat($H[L_n(\tilde{\theta})]$, $Sens(H[L_n(\tilde{\theta})])$, ϕ_2, c) `// calling Algorithm 2`

$\Sigma \leftarrow \frac{1}{n}\sum_{i=1}^n \nabla f(\vec{x}_i, y_i, \tilde{\theta})[\nabla f(\vec{x}_i, y_i, \tilde{\theta})]^T - 4c^2\tilde{\theta}\tilde{\theta}^T$

$\tilde{\Sigma} \leftarrow$ PrivSPDMat(Σ, $Sens(\Sigma)$, ϕ_3, c)

if *Requiring pure-differential privacy* **then**

  Generate m i.i.d. samples G_i $(i = 1, \ldots, m)$ from $N(\vec{0}, \tilde{\Sigma})$

  Generate m i.i.d samples β_i $(i = 1, \ldots, m)$ with density from Equation 3.2 with $\gamma = nc\phi_1$

  for $i \leftarrow 1$ **to** m **do**

    $Q_i \leftarrow \tilde{H}[L_n(\tilde{\theta})]^{-1} G_i/\sqrt{n} - \beta_i$

    $\theta^{(i)} \leftarrow \tilde{\theta} + Q_i$

  end

  for $j \leftarrow 1$ **to** d **do**

    $(\theta_L[j], \theta_R[j]) \leftarrow (1-\alpha)$-confidence interval for $\theta^{(1)}[j], \ldots, \theta^{(m)}[j]$

  end

else

  `// for zCDP`

  $z_{\alpha/2} \leftarrow (1-\alpha/2)$-quantile of standard normal

  $U = \frac{1}{2\phi_1(nc)^2}\mathbf{I}_d + \frac{1}{n}\tilde{H}[L_n(\tilde{\theta})]^{-1}\tilde{\Sigma}\tilde{H}[L_n(\tilde{\theta})]^{-1}$

  for $j \leftarrow 1$ **to** d **do**

    $\theta_L[j] \leftarrow \tilde{\theta}[j] - z_{\alpha/2}\sqrt{U_{jj}}$

    $\theta_R[j] \leftarrow \tilde{\theta}[j] + z_{\alpha/2}\sqrt{U_{jj}}$

  end

end

Output θ_L, θ_R

Theorem 4 . *Under the same conditions as Theorem 3, Algorithm 5 satisfies* $(\phi_1 + \phi_2 + \phi_3)$*-differential privacy and* $(\phi_1 + \phi_2 + \phi_3)$*-zCDP.*

The proof of Theorem 4 is in Appendix A.4.

6. Applications to Logistic Regression and SVM

We now apply our confidence interval algorithms to logistic regression and support vector machines. Both models can be learned by objective and output perturbation [Chaudhuri et al., 2011]. In order to apply our confidence interval algorithms, we need to compute the L_2 sensitivity of the Hessian and covariance matrices, as those quantities are needed to calibrate the amount of perturbation

of those matrices that we need to protect privacy (Algorithm 2). Again, we will assume that the feature vector x has $\|x\|_2 \leq 1$ and the label $y \in \{-1, 1\}$.

For logistic regression, the gradient and Hessian are well known:

$$\nabla f(y\theta^T\vec{x}) = -yS(-y\theta^T\vec{x})\vec{x},$$
$$H[f(y\theta^T\vec{x})] = S(-y\theta^T\vec{x})S(y\theta^T\vec{x})\vec{x}\vec{x}^T,$$

where S is the sigmoid function. It is also well known that the loss function is convex and doubly differentiable with $|f'(z)| \leq 1$ and $|f''(z)| \leq 1/4$.

For SVM, it is well-known that the piecewise gradient and Hessian for the Huber loss $f_{Huber}(y\theta^T\vec{x})$ are:

$$\nabla f_{Huber}(y\theta^T\vec{x}) = \begin{cases} \vec{0} & \text{if } y\theta^T\vec{x} > 1+h \\ \frac{y}{2h}(y\theta^T\vec{x} - 1 - h)\vec{x} & \text{if } |1 - y\theta^T\vec{x}| \leq h \\ -y\vec{x} & \text{if } y\theta^T\vec{x} < 1-h, \end{cases}$$

and

$$H[f_{Huber}(y\theta^T\vec{x})] = \begin{cases} \frac{y^2}{2h}\vec{x}\vec{x}^T & \text{if } |1 - y\theta^T\vec{x}| \leq h \\ 0_{d\times d} & \text{otherwise.} \end{cases}$$

Huber loss is convex and differentiable, and piecewise doubly-differentiable, with $|f'_{Huber}(z)| \leq 1$ and $|f''_{Huber}(\cdot)| \leq \frac{1}{2h}$ [Chaudhuri et al., 2011]. Even though the second derivative does not exist at a few isolated points, Chaudhuri et al. [2011] proved that objective and output perturbation algorithms for SVM still preserve privacy.

We now derive the L_2 sensitivity for the Hessian and the covariance matrix for logistic regression and SVM.

Lemma 2. *The L_2-sensitivity of the covariance matrix Σ (defined in Equation 4.3) for logistic regression is at most $2S(\|\theta_0\|_2)^2/n$.*

The proof of Lemma 2 is in Appendix A.5.

Lemma 3. *The L_2-sensitivity of the Hessian $H[J_n(\tilde{\theta})]$ (defined in Equation 4.2) for logistic regression is at most $1/(2n)$.*

The proof of Lemma 3 is in Appendix A.6.

Lemma 4. *The L_2-sensitivity of the covariance matrix Σ (defined in Equation 4.3) for SVM is at most $2/n$.*

The proof of Lemma 4 is in Appendix A.7.

Lemma 5. *The L_2-sensitivity of the Hessian $H[J_n(\tilde{\theta})]$ (defined in Equation 4.2) for SVM is at most $1/(nh)$.*

The proof of Lemma 5 is in Appendix A.8.

7. Experiments

To test the differentially private confidence interval algorithms, we run comprehensive experiments on several real datasets, which are described in Section 7.1. We discuss evaluation metrics in Section 7.2.

The experiments are then organized as follows. The algorithms have to allocate portions of the privacy budget across several sub-tasks – obtaining the differentially private coefficients, estimating the covariance matrix, and estimating the Hessian. Thus, we first experiment with the allocation of the privacy budget across sub-tasks in Section 7.3. We empirically find the split of the privacy

budget for the three parts such that the private confidence intervals are short in length. We then use the chosen split throughout the rest of the experiments.

In Section 7.4, we present results about the sample size needed to achieve a desired coverage percentage from the differentially private confidence intervals. We also present the corresponding interval lengths (coverage and length must be considered together, since a confidence interval with 100% coverage but near-infinite length is of no use). Then in Section 7.5, we compare the empirical sample complexity of private and non-private confidence intervals. More precisely, given a sample size n', we map it to n such that the length of the non-private confidence intervals computed using n' data points is equivalent to that of the differentially private confidence intervals computed using n data points.

In Section 7.6, we empirically study how far our intervals are from optimality. The length of our confidence intervals depends on four factors: (A) the randomness in the data, (B) the randomness in the private regression algorithms, (C) the randomness and estimation error of the Hessian and covariance matrix, (D) the approximation error from our derivation. Combining real datasets with simulations, we measure the variability of the regression coefficients that is due to points (A) and (B). We call those intervals *Variability Intervals* (see Section 7.2 for more information) and they can be interpreted as the true variability of private regression algorithms. Hence they are a lower bound on any possible differentially private confidence interval length – this is not necessarily a tight lower bound, as variability intervals are not differentially private themselves. By comparing confidence intervals to variability intervals, we are isolating the overhead due to our approach – the loss due to points (C) and (D) in Section 7.6. In Section 7.7, we study the relationship between length of the private confidence intervals and other parameters. The run time of our algorithms is reported in Section 7.8.

Sections 7.3 through 7.7 contain a representative sample of the experiments (as many results are qualitatively similar). For complete experimental results, see Supplemental Appendix B.

7.1. **Datasets.** We run experiments on several real datasets: Adult and KDDCUP99 data sets from Lichman [2013], the Banking data set [Moro et al., 2014], the IPUMS-US [ipu, 2017b] dataset and the IPUMS-BR [ipu, 2017a] dataset. Adult [Lichman, 2013] is a dataset extracted from the 1994 Census database and contains 30,162 records on demographic information. A common task based on it is predicting whether annual income exceeds $\$50K$. KDDCUP99 [Lichman, 2013] is the dataset used for the Third International Knowledge Discovery and Data Mining Tools Competition which contains 4,898,431 records. It contains network traffic data simulated in a military network environment and the goal is to distinguish network attacks. Banking [Moro et al., 2014] contains 45,211 records on the direct marketing phone calls of a Portuguese banking institution, and is used for predicting whether the client will subscribe a term deposit. US [ipu, 2017b] and BR [ipu, 2017a] are from IPUMS that provides census and survey data from around the world integrated across time and space. Users can freely choose the data samples and the variables to be used to create data extracts. In the paper, we use the versions from Zhang et al. [2013] where US has 39,928 records and BR has 38,000 records. The targets for both of them are predicting whether personal income exceeds some thresholds.

All the datasets contain both numerical and categorical attributes. As was done in Chaudhuri et al. [2011], Zhang et al. [2013], following common practice for regression problems, we binarize each categorical attribute so that an attribute with cardinality k becomes k binary attributes. Following Chaudhuri et al. [2011], Zhang et al. [2013], we then standardize each attribute so its maximum attribute value becomes 1. As for the target column, it is mapped to either -1 or 1. After pre-processing, the dimensionality of each dataset is given in Table 1. To experiment with the sample size and dimensionality, we may extract sub-datasets from those datasets by first randomly permuting the dataset and then taking the first n_1 samples and/or the first d_1 features from it. We

Table 1: Real Datasets Summary

Dataset	n	d (after binarizing categorical attributes)
Adult	30,162	37
KDDCUP99	4,898,431	90
Banking	45,211	34
IPUMS-US	39,928	57
IPUMS-BR	38,000	53

also add a column of ones as the constant feature to each dataset, and normalize each record so that it lies in the unit L_2 ball. The dimensionality d reported with the experimental results is the one before adding the constant feature.

7.2. **Measures and Methodology.** The quality of confidence intervals is evaluated using two complementary measures, coverage percentage and length. Coverage percentage is the fraction of times they cover the true parameters, so a putative 95% confidence interval should cover the true parameter at least 95% of the time. However, an infinitely long confidence interval can also cover the true parameter at least 95% of the time, so we must also evaluate how short the confidence intervals are. We next explain how we measure coverage and then we introduce a non-private baseline called the *variability interval*, which is a lower bound on any differentially private confidence interval (that is, no differentially private confidence interval can be shorter than the variability interval).

Coverage percentage. For each dataset $\mathcal{D}$, we treat its empirical distribution as the true distribution and the non-private model parameters learned on the data as θ_0. To simulate the effects of sampling, we create multiple "sampled" datasets $\mathcal{D}_1, \ldots, \mathcal{D}_k$ by sampling with replacement from the original dataset $\mathcal{D}$. Each such dataset $\mathcal{D}_i$ is called a *bootstrap replicate*. To estimate coverage, for each $\mathcal{D}_i$ we use our algorithm to compute the privacy-preserving confidence interval $\left(\theta_L^{(i)}[j], \theta_R^{(i)}[j]\right)$ for each coordinate j of θ_0. The coverage percentage for a parameter $\theta_0[j]$ is then the fraction of the privacy-preserving confidence intervals that contain $\theta_0[j]$. The overall coverage is then the average coverage over all parameters:

$$\frac{\sum_{i=1}^{k}\sum_{j=1}^{d'} \mathbf{1}_{\theta_L^{(i)}[j] \le \theta_0[j] \le \theta_R^{(i)}[j]}}{kd'},$$

where $\mathbf{1}$ is the indicator function and d' is the dimensionality of θ_0 after adding the constant feature.

Variability Intervals VI. The variability interval is a non-private baseline that directly measures the actual variation in parameter estimate due to sampling noise and due to the algorithm that estimates the parameters (e.g., output perturbation or objective perturbation). This is possible to obtain in controlled experiments. On the other hand, confidence intervals are an estimate (not a direct measurement) of this variability. We obtain variability intervals as follows:

For each dataset $\mathcal{D}$, we treat its empirical distribution as the true distribution and the non-private model parameters learned on the data as θ_0. To simulate the effects of sampling, we create multiple "sampled" datasets $\mathcal{D}_1, \ldots, \mathcal{D}_m$ by sampling with replacement from the original dataset $\mathcal{D}$. Each such dataset $\mathcal{D}_i$ is called a *bootstrap replicate* and there are $m = 10,000$ of them. This simulates variability due to sampling. On each $\mathcal{D}_i$ we run the privacy-preserving ERM algorithm (either output or objective perturbation) to get the estimate $\tilde{\theta}^{(i)}$. The variability in these $\tilde{\theta}^{(i)}$ is thus solely due to sampling and privacy noise used to create the parameter estimates (with privacy budget ϕ_1) – in other words, it is not affected by the ϕ_2 and ϕ_3 that are used in our confidence interval algorithms.

For $1 \le j \le d'$, let $\tilde{\theta}_L[j]$ and $\tilde{\theta}_R[j]$ be the $\alpha/2$-quantile and $(1-\alpha/2)$-quantile of $\tilde{\theta}^{(1)}[j], \cdots, \tilde{\theta}^{(m)}[j]$, respectively. Then the $1-\alpha$ variability interval for coordinate j is $(\tilde{\theta}_L[j], \tilde{\theta}_R[j])$.

Note that the variability intervals are the true quantiles of our simulation. Any valid $1-\alpha$ differentially private confidence interval must therefore be at least as long as the $1-\alpha$ variability interval and so the quality of a confidence interval is measured as how long it is compared to the variability interval. In experiments we plot average length of confidence intervals vs. average length of variability intervals.

Throughout our experiments, we use $\alpha = 0.05$, $m = 10,000$ (recall m is also the number of post-processing samples used in Algorithms 3 and 5 to estimate confidence intervals). For simplicity, in the figure legends, we use **DP** for differential privacy, **zCDP** for zero-concentrated differential privacy, **CI** for confidence interval, **VI** for variability interval, **obj** for ERM with objective perturbation, **output** for ERM with output perturbation, **LR** for logistic regression and **SVM** for support vector machines.

In our experimental results, we report the privacy parameters ϵ for differential privacy and ρ for zCDP. To compare differentially private and zCDP algorithms on the same plot, we set $\rho = \frac{\epsilon^2}{2}$. This is the closest possible apples-to-apples comparison, as any algorithm satisfying ϵ-differential privacy satisfies $\rho = \frac{\epsilon^2}{2}$ zCDP [Bun and Steinke, 2016].

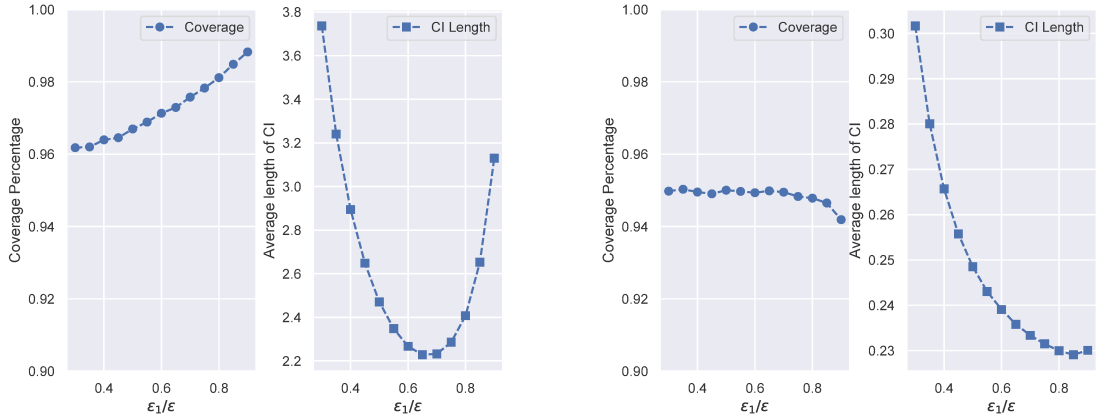
7.3. **Allocation for the Privacy Budget.** Algorithms 3 and 5 each take three privacy budgets ϕ_1, ϕ_2 and ϕ_3 as parameters, used to compute the coefficients, Hessian, and covariance matrix, respectively. In this section, we empirically study the allocation for the three parameters given the total privacy budget. For each method, we experiment with various allocations under different settings and select the one that produces the shortest private confidence intervals with at least 95% coverage.

In the first group of experiments (Section 7.3.1), we test how much of the total privacy budget should be allocated to ϕ_1, which is used by the ERM algorithms for parameter estimation. While varying the percentage of total privacy budget allocated to ϕ_1, we split the remainder equally between ϕ_2 and ϕ_3. For each setting, we use $k = 100,000$ bootstrap replicates to compute the coverage percentage and length of the privacy-preserving confidence intervals. For pure ϵ-differentially private algorithms, we vary the proportion of the total privacy budget assigned to ϕ_1 from 0.3 to 0.9. For ρ-zCDP algorithms, we vary the proportion of the budget allocated to ϕ_1 from 0.1 to 0.98.

Once we settle on an allocation for ϕ_1, we experiment (in Section 7.3.2) with how to divide the remaining privacy budget between ϕ_2 and ϕ_3 . We again use $k = 100,000$ bootstrap replicates to compute the coverage percentage and length of the privacy-preserving confidence intervals.

Ideally, we would find some allocation for the three budgets that consistently works well no matter the dataset, the sample size, and the dimensionality. Therefore, for each method, we experiment under various settings by using different datasets, sample sizes and dimensionality. The results for logistic regression and SVM are similar, so we show representative results in this section and the rest in Supplemental Appendix B.1.

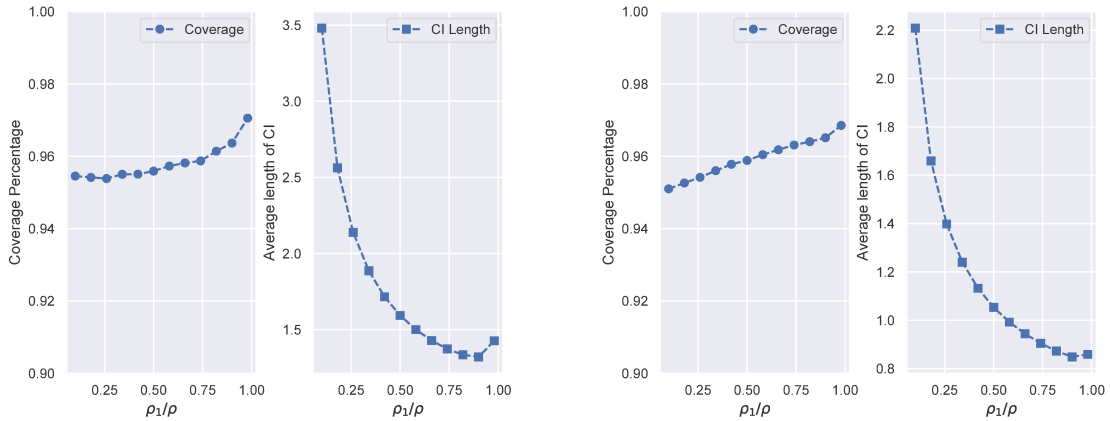
7.3.1. *Allocation for ϕ_1.* We first consider the combination of pure differential privacy and objective perturbation. Out of a total budget of ϵ, ϵ_1 out of ϵ is used by objective perturbation to obtain logistic regression coefficients. The results are shown in Figure 1. It consists of two pairs of plots. In Figure 1a, we consider the Adult dataset and dimensionality $d = 5$. The left part of the figure shows coverage as a function of ϵ_1/ϵ and the right side shows the corresponding confidence interval length. Figure 1b uses the IPUMS-US dataset with $d = 1$ (i.e. we consider one feature aside from the constant feature). As the ratio ϵ_1/ϵ increases, coverage percentage also tends to increase in Figure 1a. The coverage is greater than 95% for all values of ϵ_1/ϵ that we tested in Figure 1a. In Figure 1b, we observe that the coverage stays around 95% until ϵ_1/ϵ increases to about 0.8. With $\epsilon_1/\epsilon > 0.8$, coverage slightly decreases. We also notice such patterns in other results where $d = 1$. On the other hand, the length of the privacy-preserving confidence intervals from the plots shows a



(A) Adult, $n = 5,000$, $d = 5$ (B) US, $n = 5,000$, $d = 1$

Figure 1: [ϵ-**DP, objective perturbation, logistic regression**] Coverage percentage and average length of confidence intervals vs. ϵ_1/ϵ for objective perturbation based ϵ-DP confidence intervals with a total privacy budget of $\epsilon = 1.0$. $\epsilon_2 = \epsilon_3 = (\epsilon - \epsilon_1)/2$, $c = 0.001$.

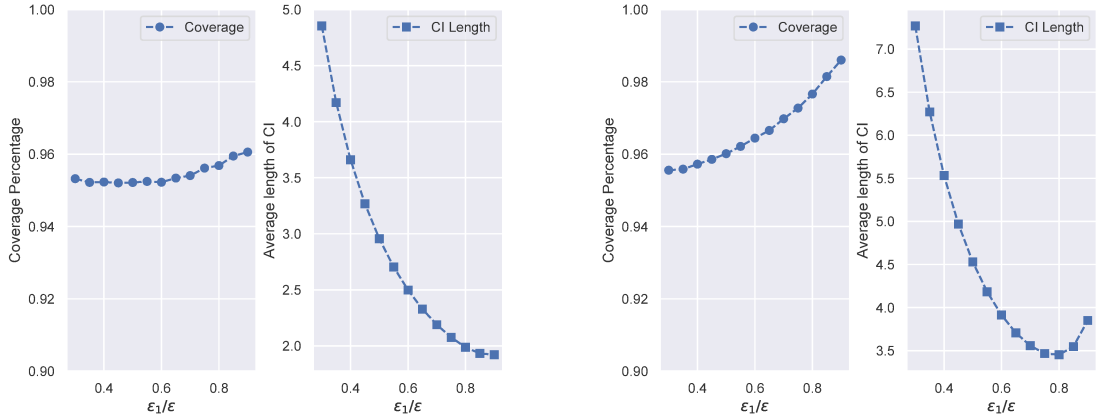
parabola shape with first decreasing and then increasing length for increasing values of ϵ_1/ϵ. This is expected behavior – when ϵ_1/ϵ is small, the coefficients are very noisy, resulting in large intervals. When ϵ_1/ϵ is close to 1, very little budget remains for estimating the intervals hence they become very noisy and large. In the plots, the confidence interval length is small when ϵ_1/ϵ is around 0.65 and at the same time the coverage is at least 95%. Thus we set ϵ_1/ϵ to be 0.65 for objective perturbation when using pure differential privacy.



(A) Banking, $n = 5,000$, $d = 5$ (B) BR, $n = 10,000$, $d = 10$

Figure 2: **[zCDP, objective perturbation, logistic regression]** Coverage percentage and average length of confidence intervals vs. ρ_1/ρ for objective perturbation based zCDP confidence intervals with a total privacy budget of $\rho = 0.5$. $\rho_2 = \rho_3 = (\rho - \rho_1)/2$, $c = 0.001$.

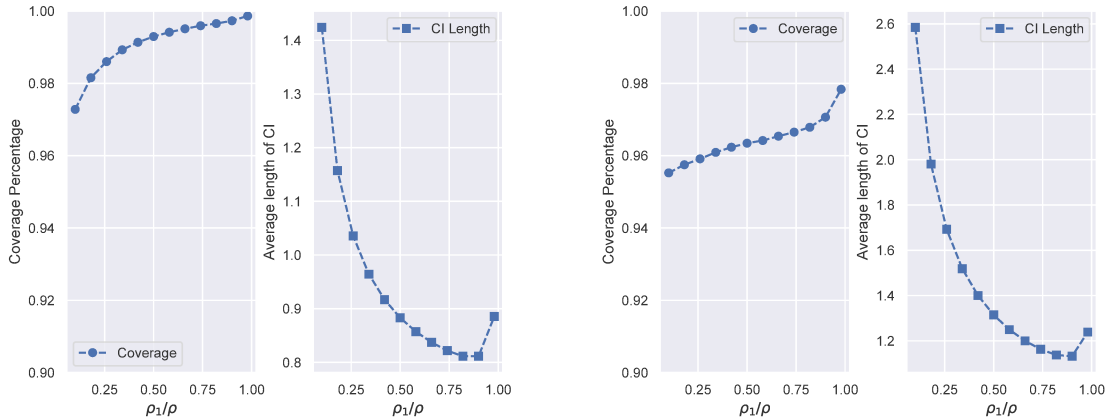
We next consider objective perturbation with ρ-zCDP in Figure 2. Here ρ is the total privacy budget and ρ_1 out of ρ is used for estimating coefficients. In both parts of the figure, the coverage is consistently over 95%. As with the case of pure differential privacy, the length of the privacy-preserving confidence intervals starts large, decreases rapidly and then starts to increase. Setting $\rho_1/\rho = 0.9$ seems to provide short confidence intervals at the desired level of coverage.



(A) Banking, $n = 5,000$, $d = 1$ (B) Banking, $n = 5,000$, $d = 5$

Figure 3: [**ϵ-DP, output perturbation, logistic regression**] Coverage percentage and average length of confidence intervals vs. ϵ_1/ϵ for output perturbation based ϵ-DP confidence intervals with a total privacy budget of $\epsilon = 1.0$. $\epsilon_2 = \epsilon_3 = (\epsilon - \epsilon_1)/2$, $c = 0.001$.

In Figure 3, we consider output perturbation and pure differential privacy. Again, coverage stays at the desired level. Setting ϵ_1/ϵ to be 0.8 seems to result in good coverage and short confidence interval length.

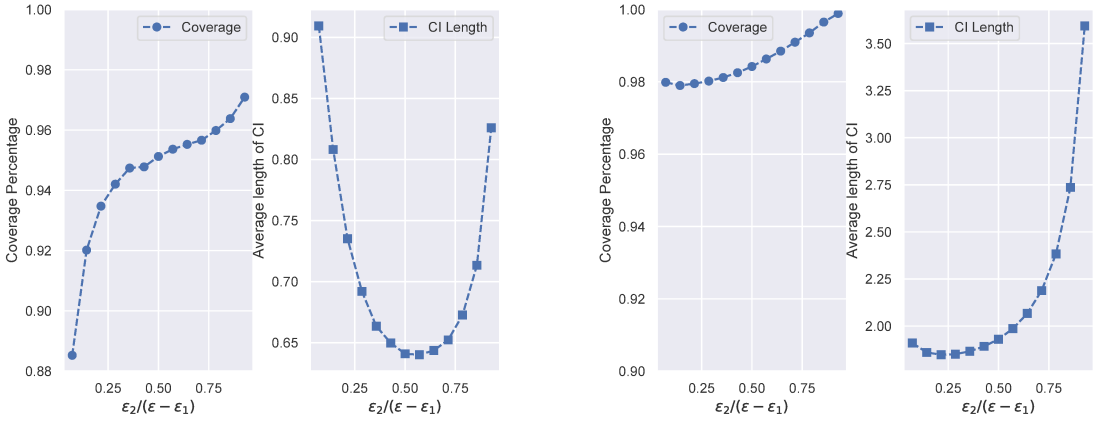


(A) KDDCUP99, $n = 10,000$, $d = 10$ (B) Banking, $n = 5,000$, $d = 5$

Figure 4: **[zCDP, output perturbation, logistic regression]** Coverage percentage and average length of confidence intervals vs. ρ_1/ρ for output perturbation based zCDP confidence intervals with a total privacy budget of $\rho = 0.5$. $\rho_2 = \rho_3 = (\rho - \rho_1)/2$, $c = 0.001$.

In Figure 4, we consider output perturbation and ρ-zCDP. The coverage generally stays above the desired level of 95%. As for length of the privacy-preserving confidence intervals, the pattern is the same as before, i.e., length starts large, decreases rapidly as we increase the value for ρ_1/ρ, and starts to increase again. Setting $\rho_1/\rho = 0.9$ seems to provide a good combination of short intervals and the desired coverage.

7.3.2. *Allocation for the Remaining Budget.* Having set an allocation for ϕ_1 for each method, we next experiment with the allocation for ϕ_2 and ϕ_3 from the remaining privacy budget.



(A) Adult, $n = 5,000$, $d = 1$ (B) Adult, $n = 10,000$, $d = 10$

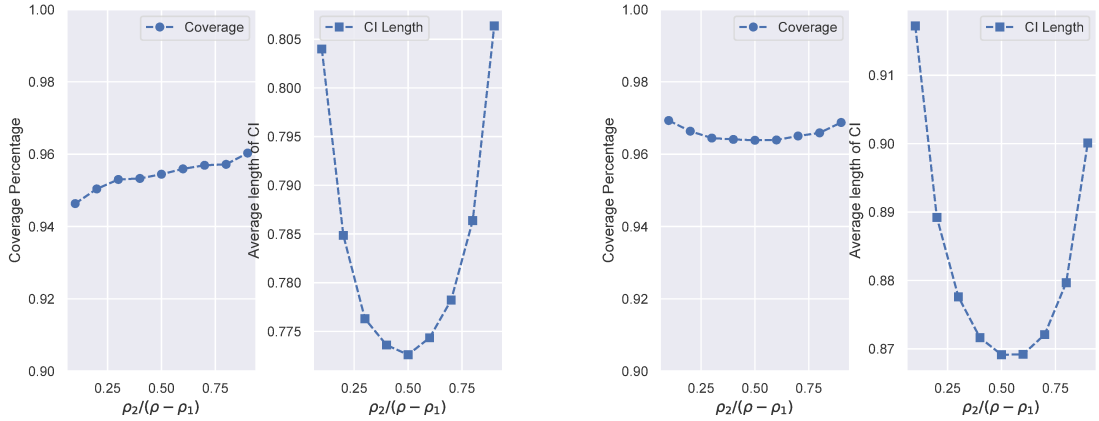
Figure 5: [**ϵ-DP, objective perturbation, logistic regression**] Coverage percentage and average length of confidence intervals vs. $\epsilon_2/(\epsilon - \epsilon_1)$ for objective perturbation based ϵ-DP confidence intervals with a total privacy budget of $\epsilon = 1.0$. $\epsilon_1 = 0.65$, $c = 0.001$.

We first consider the combination of pure differential privacy and objective perturbation. Given $\epsilon_1/\epsilon = 0.65$ from Section 7.3.1, out of the remaining budget 0.35ϵ, ϵ_2 out of 0.35ϵ is used to compute the pure differentially private estimation for the Hessian and ϵ_3 out of 0.35ϵ is used to compute the pure differentially private estimation for the covariance matrix. The results are shown in Figure 5. It consists of two pairs of plots. In Figure 5a, we consider the Adult dataset and dimensionality $d = 1$. Again, the left part of the figure shows coverage as a function of $\epsilon_2/(\epsilon - \epsilon_1) = \epsilon_2/(0.35\epsilon)$ and the right side shows the corresponding confidence interval length. Figure 5b uses the Adult dataset with $d = 10$. As the ratio $\epsilon_2/(\epsilon - \epsilon_1)$ increases, coverage percentage also tends to increase in Figure 5a. The coverage is greater than 95% when $\epsilon_2/(\epsilon - \epsilon_1)$ increases to about 0.5. In Figure 5b, the coverage is greater than 95% for all values of $\epsilon_2/(\epsilon - \epsilon_1)$ that we tested. On the other hand, the length of the privacy-preserving confidence intervals from the plots shows a parabola shape with first decreasing and then increasing length for increasing values of $\epsilon_2/(\epsilon - \epsilon_1)$. In the plots, the confidence interval length is small when $\epsilon_2/(\epsilon - \epsilon_1)$ is around 0.5 and at the same time the coverage is at least 95%. Thus we set $\epsilon_2/(\epsilon - \epsilon_1)$ to be 0.5 for the combination of pure differential privacy and objective perturbation.

We next consider objective perturbation with ρ-zCDP in Figure 6. Given $\rho_1/\rho = 0.8$ from Section 7.3.1, ρ_2 out of the remaining budget $\rho - \rho_1 = 0.2\rho$ is used for the private estimate for the Hessian with zCDP and ρ_3 out of $\rho - \rho_1$ is used for the private estimate for the covariance matrix with zCDP. In both parts of the figure, the coverage is over 95% except for the first point (i.e., $\rho_2/(\rho - \rho_1) = 0.125$) in Figure 6a. As with the case of pure differential privacy, the length of the privacy-preserving confidence intervals starts large, decreases rapidly and then starts to increase. In the plots, the confidence interval length is small when $\rho_2/(\rho - \rho_1)$ is around 0.5 and at the same time the coverage is at least 95%. Thus we set $\rho_2/(\rho - \rho_1)$ to be 0.5.

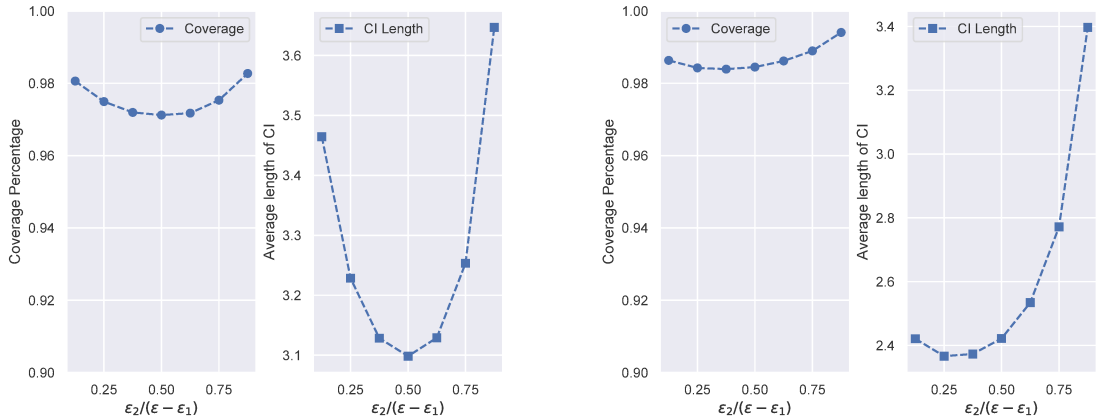
In Figure 7, we consider output perturbation and pure differential privacy. Again, coverage stays at the desired level. Setting $\epsilon_2/(\epsilon - \epsilon_1)$ to be 0.5 seems to result in good coverage and short confidence interval length.

In Figure 8, we consider output perturbation and ρ-zCDP. The coverage is greater than 95% for all values of $\rho_2/(\rho - \rho_1)$ that we tested in both of the plots. As for length of the privacy-preserving confidence intervals, the pattern is slightly different from other methods. The confidence interval is short even when only 25% of the remaining privacy budget is allocated to ρ_2. However, as the ratio $\rho_2/(\rho - \rho_1)$ increases from this point, the confidence interval length also gets larger in Figure 8. The



(A) Banking, $n = 5,000$, $d = 1$ (B) Banking, $n = 10,000$, $d = 10$

Figure 6: **[zCDP, objective perturbation, logistic regression]** Coverage percentage and average length of confidence intervals vs. $\rho_2/(\rho - \rho_1)$ for objective perturbation based zCDP confidence intervals with a total privacy budget of $\rho = 0.5$. $\rho_1 = 0.45$, $c = 0.001$.



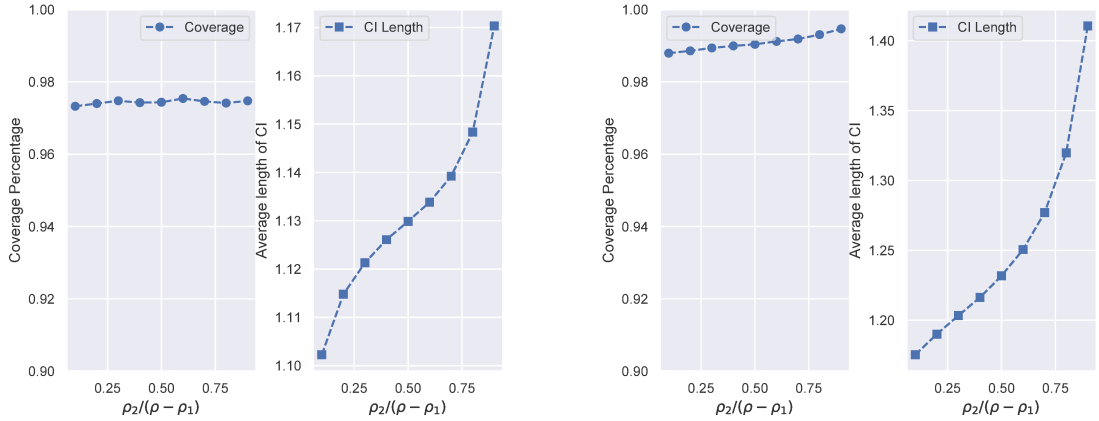
(A) BR, $n = 5,000$, $d = 5$ (B) BR, $n = 10,000$, $d = 10$

Figure 7: **[ϵ-DP, output perturbation, logistic regression]** Coverage percentage and average length of confidence intervals vs. $\epsilon_2/(\epsilon-\epsilon_1)$ for output perturbation based ϵ-DP confidence intervals with a total privacy budget of $\epsilon = 1.0$. $\epsilon_1 = 0.8$, $c = 0.001$.

increase is modest. For example, in Figure 8a, the confidence interval length grows from 1.12 to 1.13 as the allocation increases from 0.25 of the remaining privacy budget to 0.5. Hence, for all methods, a good general rule is to split the remaining privacy budget equally between the covariance matrix and the Hessian.

In Table 2, we summarize the allocations of the privacy budgets that work well empirically for each method. We use these allocations throughout the rest of the experiments.

7.4. **Empirical Sample Complexity of Private Confidence Intervals.** A general trend in statistics is that a theoretical analysis of finite-sample length of confidence intervals is too conservative – the estimated sample complexity needed to achieve a specified level of coverage would result in confidence intervals that are much longer than necessary. Thus here we evaluate the empirical sample complexity – how many data points are needed to achieve 95% confidence intervals with the



(A) KDDCUP99, $n = 5,000$, $d = 1$ (B) KDDCUP99, $n = 5,000$, $d = 5$

Figure 8: **[zCDP, output perturbation, logistic regression]** Coverage percentage and average length of confidence intervals vs. $\rho_2/(\rho - \rho_1)$ for output perturbation based zCDP confidence intervals with a total privacy budget of $\rho = 0.5$. $\rho_1 = 0.45$, $c = 0.001$.

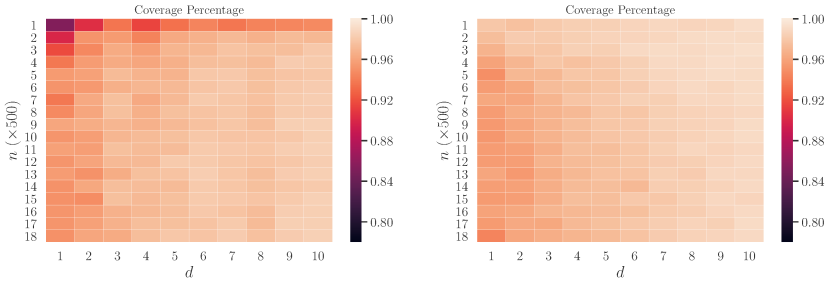
Table 2: Empirical Allocation for Privacy Budgets given the Total Budget $\phi = \epsilon$ for pure DP or $\phi = \rho$ for zCDP

	pure DP		zCDP	
	objective perturbation	output perturbation	objective perturbation	output perturbation
ϕ_1	0.65ϵ	0.8ϵ	0.9ρ	0.9ρ
ϕ_2	0.175ϵ	0.1ϵ	0.05ρ	0.05ρ
ϕ_3	0.175ϵ	0.1ϵ	0.05ρ	0.05ρ

methods we proposed. The sample complexity is affected by two other factors: the dimensionality d and the total privacy budget ϕ ($\phi = \epsilon$ for pure-DP and $\phi = \rho$ for ρ-zCDP). In each case we plot a heatmap where the x axis is the experimental variable we modify, the y axis is sample size, and the color indicates coverage percentage (we also provide a corresponding graph where the color indicates confidence interval length). Each coverage and length estimate is the average of $k = 1,000$ repetitions. We show the heatmaps for logistic regression on the Adult dataset in Figures 9 and 10, the KDDCUP99 dataset in Figures 11 and 12. See Supplemental Appendix B.2 for complete results including those for SVM as well as other datasets.

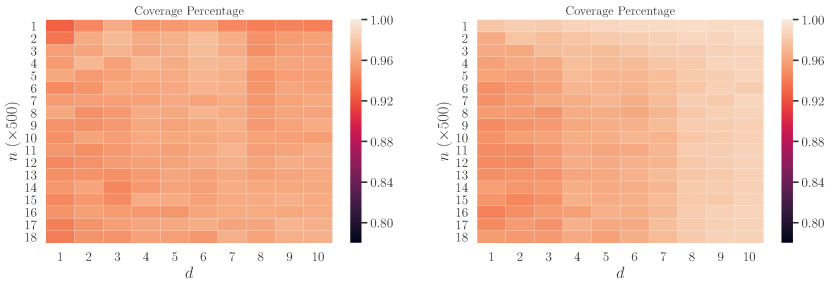
7.4.1. *Varying Dimensionality.* In Figures 9 and 10, we vary the parameter d while fixing the total privacy budget.

Figure 9 shows a heatmap of coverage percentage for various combinations of n and d for each of the four private interval estimation methods. To compare them all together, we use the same scale to show in the subplots. Note that each cell is an average of $k = 1,000$ repetitions, as a result the heat map is not perfectly smooth. For objective perturbation with pure-DP, the coverage starts low at very small values for n, and then grows as we increase n. Coverage reaches 95% when n is around $4,500$ with $d = 1$. When we increase d, coverage is generally increasing, so the sample complexity needed to achieve the desired coverage decreases. For output perturbation with pure-DP and ρ-zCDP, a similar trend on coverage happens for d, i.e., coverage is increasing with d. The coverage percentages generally exceed 95%, even for small n around 500. For objective perturbation with zCDP, the coverage percentage is about 95% once $n \geq 2,000$. From the plots, we see that



(A) ϵ-DP, objective perturbation

(B) ϵ-DP, output perturbation



(C) zCDP, objective perturbation

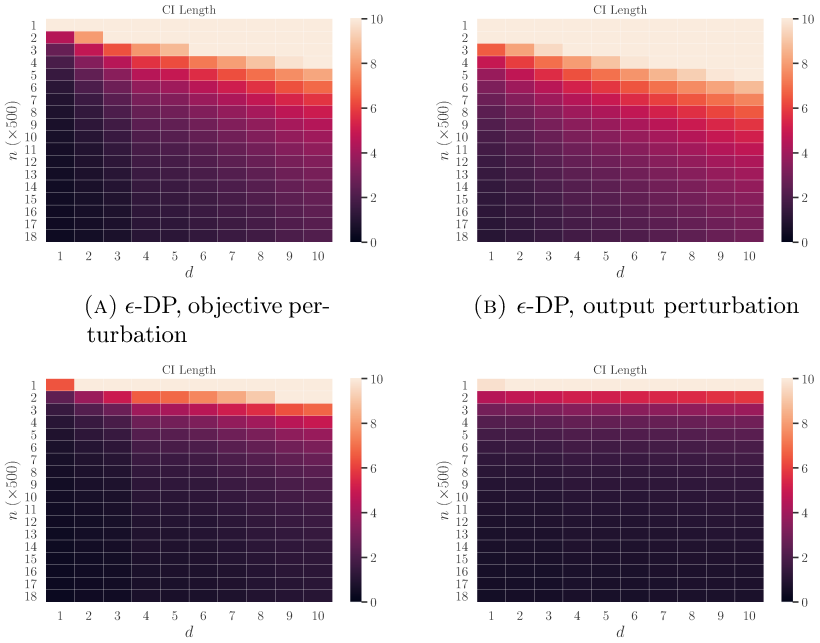
(D) zCDP, output perturbation

Figure 9: Coverage percentage from 1000 confidence intervals as a function of the sample size n and the dimensionality d on Adult dataset for logistic regression. $\epsilon = 1.0$, $\rho = \epsilon^2/2 = 0.5$, $c = 0.001$.

output perturbation based methods outperform objective perturbation based methods in sample complexity to achieve desired coverage.

In Figure 10, the corresponding lengths for the privacy-preserving confidence intervals under the same settings as Figure 9 are reported. Again, we use the same scale to show the results for the four methods. The general trend is that confidence intervals under zCDP are shorter than the corresponding intervals for pure DP. This is expected as zCDP is a relaxation of pure DP. Also, as expected, the length increases as dimensionality increases. We note that zCDP is more robust to changes in dimensionality. We stop at $d = 10$ as this results in a Hessian and covariance matrix with 121 entries ($d' = 11$ due to the constant feature). We expect interval length (and also running time of the parameter estimation algorithms) to start degrading rapidly with further increases in dimensionality.

7.4.2. *Varying Total Privacy Budget.* In Figures 11 and 12, we vary the total privacy budget while fixing $d = 10$. We show the coverage results in Figure 11 and the length results in Figure 12. Note that for the purpose of comparison, we set $\rho = \epsilon^2/2$ as every pure ϵ-differentially private algorithm satisfies ρ-zCDP for $\rho = \epsilon^2/2$ [Bun and Steinke, 2016]. In all cases, coverage percentage increases with n and total privacy budget. It is worth noting that the coverage of output perturbation methods remains more stable than for objective perturbation as the output perturbation methods achieve the desired coverage percentage for all settings we tried but objective perturbation methods had more difficulty with small n and ϵ or ρ. Also, the intervals under zCDP are shorter than those of pure-DP (more detailed experiments on confidence interval length appear in the next sections). In general, zCDP with output perturbation seems to provide the best confidence intervals.



(A) ϵ-DP, objective perturbation

(B) ϵ-DP, output perturbation

(C) zCDP, objective perturbation

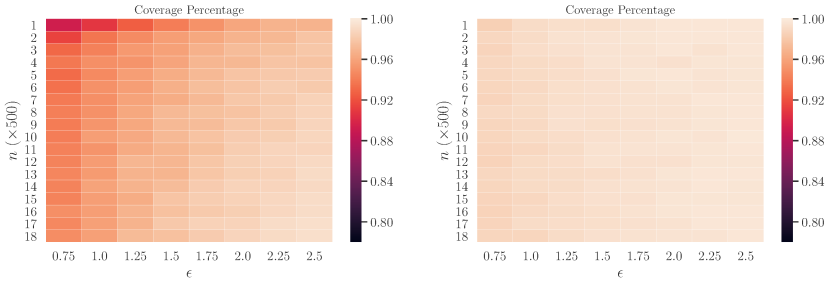
(D) zCDP, output perturbation

Figure 10: Average length from 1000 confidence intervals as a function of the sample size n and the dimensionality d on Adult dataset for logistic regression. $\epsilon = 1.0$, $\rho = \epsilon^2/2 = 0.5$, $c = 0.001$.

7.5. **The Overhead on Sample Complexity for Differential Privacy.** In the section, we experiment with the overhead on sample complexity due to these differentially private methods for computing the confidence intervals. The privacy noise results in longer confidence intervals. Thus, given a sample size, we compute the length for the non-private intervals, and determine the sample size we need so that the differentially private confidence intervals have that same length. To obtain this mapping for sample size in the privacy-preserving case, we start with $n = 2,500$, compute the average length from $k = 1,000$ private intervals, and compare it to the non-private interval. Then in the next round, we add an increment of $2,500$ to the current private sample size n and repeat the computation of the private intervals until the average length is approximately equal to or less than that of the non-private interval. We show the results in Figure 13 for logistic regression. For SVM results, see the complete version in Supplemental Appendix B.3.

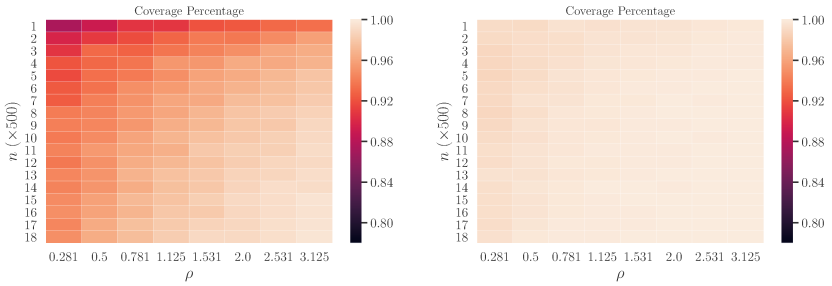
In Figure 13, the three sub-plots each corresponds to a different dataset. In each of them, the sample size for the non-private intervals ranges from $1,000$ to $10,000$. The corresponding sample size for achieving the same private interval length is shown as the vertical coordinate. In general, the three sub-plots show similar patterns. We see that for pure differential privacy, objective perturbation clearly yields shorter confidence intervals. Combined with analysis of coverage from the previous sections, we see that output perturbation should be preferred in the high privacy/low n regime as it achieves the necessary coverage, but for larger n or lower privacy regimes, objective perturbation should be preferred due to shorter confidence intervals. We also see that zCDP methods produce shorter confidence intervals than pure DP methods with zCDP + output perturbation producing slightly better results than objective perturbation.

7.6. **Comparison among the Private Confidence Intervals and the Variability Intervals.** In this section, we compare the average length of the privacy-preserving confidence intervals to



(A) ϵ-DP, objective perturbation

(B) ϵ-DP, output perturbation



(C) zCDP, objective perturbation

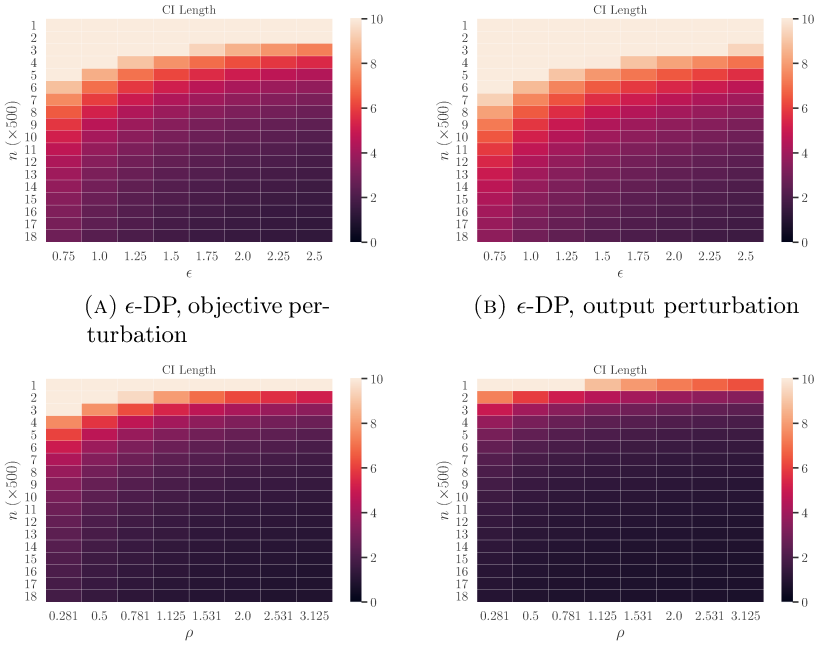
(D) zCDP, output perturbation

Figure 11: Coverage percentage from 1000 confidence intervals as a function of the sample size n and the total privacy budget ϵ (or ρ where $\rho = \epsilon^2/2$) on KDDCUP99 dataset for logistic regression. $d = 10$, $c = 0.001$.

their corresponding variability intervals (defined in Section 7.2). Recall that variability intervals are non-private lower bounds on differentially private confidence intervals. The difference between them can be attributed to the algorithms that compute the private confidence intervals after the coefficients have already been produced. The comparison between private confidence intervals and variability intervals thus monitors the overhead from the private estimation of the Hessian and covariance matrix in generating the confidence intervals.

We conduct the experiments with varying values for n, d and the total privacy budget ϕ (using the allocations determined in Table 2 in Section 7.3). Recall that those allocations shifted most of the privacy budget to the ERM algorithms so we expect an adverse effect on the overhead from confidence interval construction. The results are in Figures 14 to 16 for the application of logistic regression. For complete results including the application of SVM, see Supplemental Appendix B.4. The reported length for the confidence intervals is averaged from $k = 1,000$ independent runs. In each of the figures, there are three sub-plots. The first one compares pure-DP with zCDP methods for objective perturbation. The second one compares pure-DP with zCDP methods for output perturbation. The last one compares the winner (i.e., the method that produces shorter intervals) from the previous two comparisons (to avoid cluttering the images).

In Figure 14, we vary the sample size n. In all plots, the length for confidence intervals, variability intervals, and the gap between them decrease with increasing n. As a sanity check, note that the confidence intervals are indeed longer than their corresponding variability intervals in the plots. The difference in length of confidence intervals and variability intervals shrinks slowly with increasing n. In each case, using zCDP results in shorter confidence intervals, shorter variability intervals, and smaller overhead (even though a larger share of the privacy budget is spent on coefficient estimation rather than interval estimation). Output perturbation with zCDP clearly performs the best.



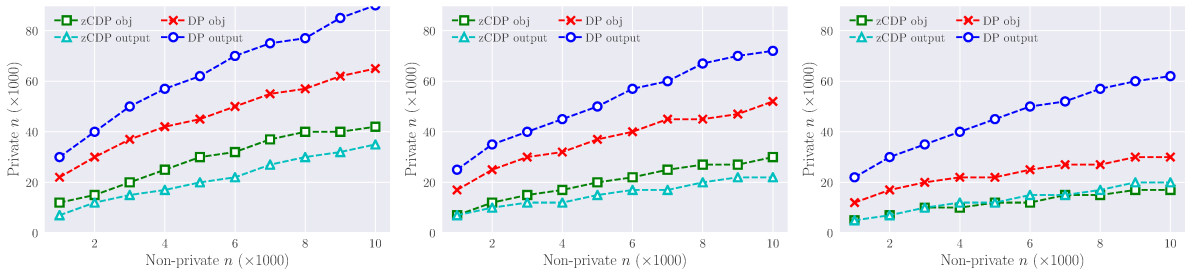
(A) ϵ-DP, objective perturbation

(B) ϵ-DP, output perturbation

(C) zCDP, objective perturbation

(D) zCDP, output perturbation

Figure 12: Average length from 1000 confidence intervals as a function of the sample size n and the total privacy budget ϵ (or ρ where $\rho = \epsilon^2/2$) on KDDCUP99 dataset for logistic regression. $d = 10$, $c = 0.001$.



(A) Adult

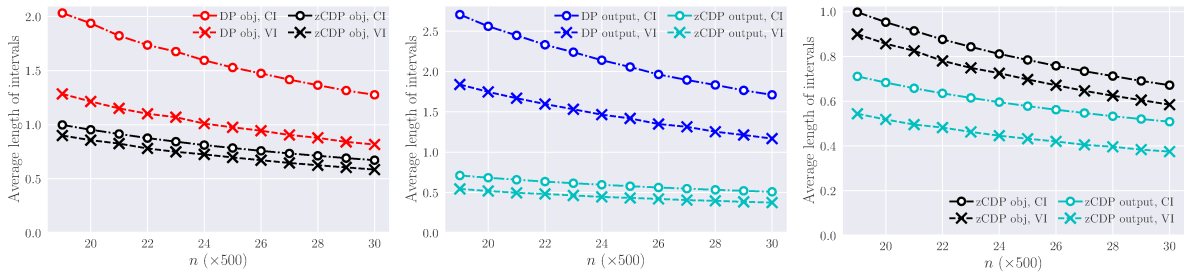
(B) Banking

(C) US

Figure 13: The mapping between sample complexities such that the average length of the non-private confidence intervals is equivalent to that of the private confidence intervals for logistic regression. $\epsilon = 1.0$ (corresponds to $\rho = 0.5$), $d = 10$, $c = 0.001$.

In Figure 15, we vary the dimensionality d. We note that in all cases, the confidence interval length, the variability interval length, and the overhead increase with dimensionality. However, zCDP with output perturbation is much more stable than the other methods.

In Figure 16, we vary the total privacy budget. Since we may compare pure-DP and zCDP methods in the same plot, we only report the total privacy budget in terms of ϵ for ϵ-differential privacy in the x-coordinate. The corresponding values for ρ of ρ-zCDP can be obtained by $\rho = \epsilon^2/2$. We note that confidence interval length decreases when the total budget increases and variability interval length also decreases. However, the variability intervals shrink much faster than the confidence intervals. This could be an effect of having most of the privacy budget allocated to estimating the coefficients (which affects variability intervals) rather than to estimating the Hessian

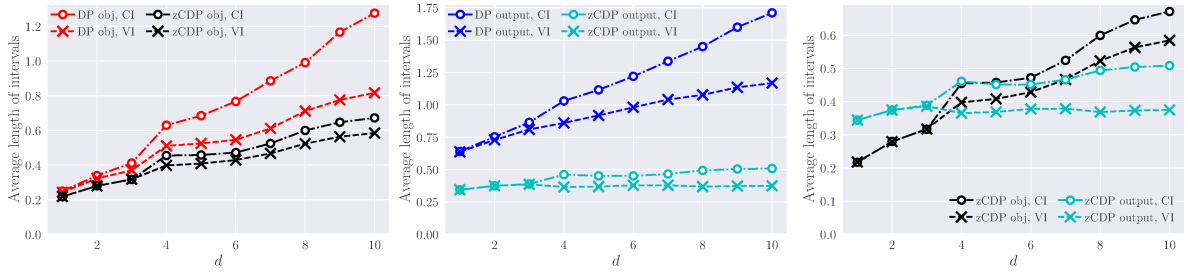


(A) ϵ-DP vs. zCDP, obj. perturbation

(B) ϵ-DP vs. zCDP, output perturbation

(C) obj. vs. output perturbation, zCDP

Figure 14: Comparison among length of intervals with varying n for logistic regression on Adult dataset. $d = 10$, $\epsilon = 1.0$ (corresponds to $\rho = 0.5$), $c = 0.001$.

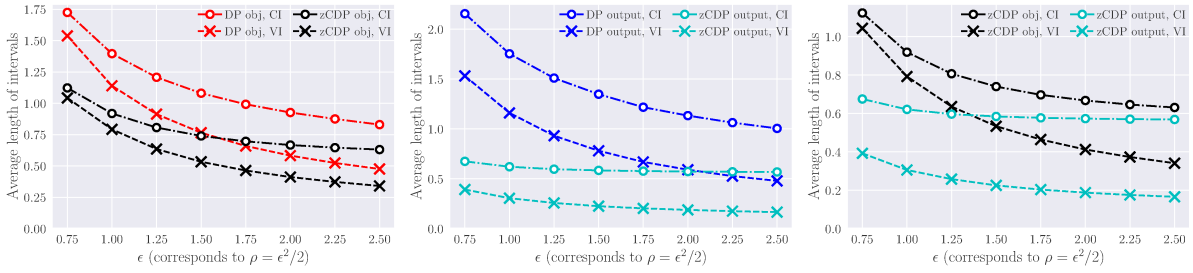


(A) ϵ-DP vs. zCDP, obj. perturbation

(B) ϵ-DP vs. zCDP, output perturbation

(C) obj. vs. output perturbation, zCDP

Figure 15: Comparison among length of intervals with varying d for logistic regression on Adult dataset. $n = 15,000$, $\epsilon = 1.0$ (corresponds to $\rho = 0.5$), $c = 0.001$.



(A) ϵ-DP vs. zCDP, obj. perturbation

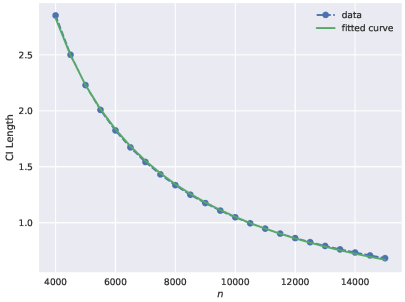
(B) ϵ-DP vs. zCDP, output perturbation

(C) obj. vs. output perturbation, zCDP

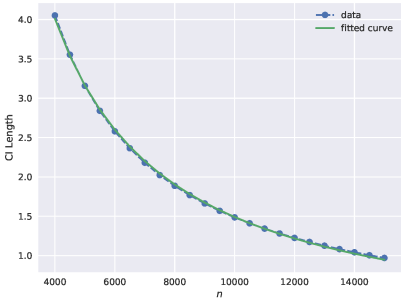
Figure 16: Comparison among length of intervals with varying ϵ (or ρ with $\rho = \epsilon^2/2$) for logistic regression on KDDCUP99 dataset. $n = 15,000$, $d = 10$, $c = 0.001$.

and covariance matrices (which affects the gap). We do not see much change with zCDP + output perturbation because those intervals are short to begin with. Note that interval length (confidence or variability) is not expected to go to 0 because they are still affected by sampling error.

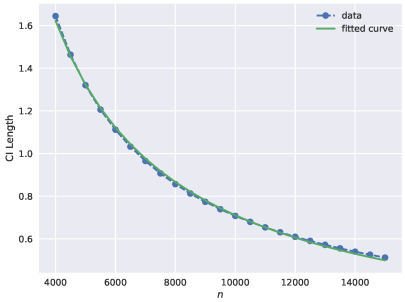
7.7. **Modeling the Relationship between Length of the Intervals and Other Parameters.** In this section we try to fit curves to the empirical lengths of the confidence intervals to give a rough idea of how the lengths of the intervals scale with n and privacy budget. We expect non-private interval length to be $O(1/\sqrt{n})$ with an additional $O(1/n)$ privacy overhead. (Roughly because in



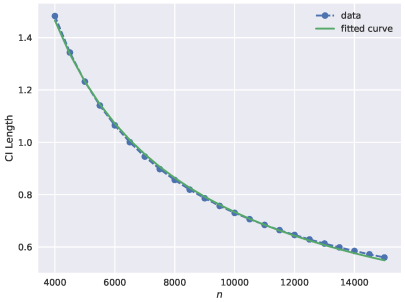
(A) ϵ-DP, objective perturbation, Adult



(B) ϵ-DP, output perturbation, US



(C) zCDP, objective perturbation, Banking



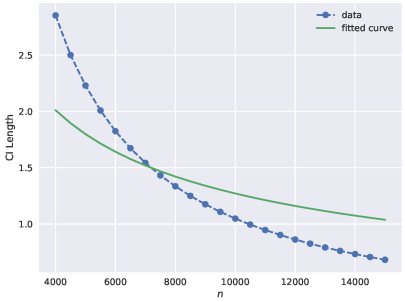
(D) zCDP, output perturbation, KDDCUP99

Figure 17: Relationship between average length of the confidence intervals and the sample size n for logistic regression. $d = 5$, $c = 0.001$, $\epsilon = 1.0$ (corresponds to $\rho = 0.5$). The fitted curve is $\frac{c_0}{n} + \frac{c_1}{\sqrt{n}}$.

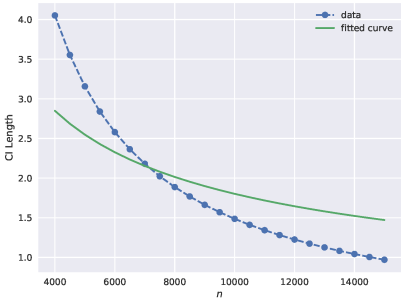
tail bounds, when a random variable X follows the Laplace or Gaussian distribution, if we fix the upper bound for $P(\frac{|X-\mu|}{n} \geq t)$, we see that n and t are inversely proportional). Hence we try to fit the curve $\frac{c_0}{n} + \frac{c_1}{\sqrt{n}}$ to the confidence interval length. The result is a relatively good fit shown in Figure 17. This extra term of $O(1/n)$ appears to be a cost of privacy. One way of supporting this claim is to simply fit $\frac{c}{\sqrt{n}}$ to the data. The results, shown in Figure 18 show a very bad fit to the data and suggest that the lower order term $O(1/n)$ is important.

To see the effect of privacy budget on the interval length, we note that in the nonprivate case, confidence intervals are approximately proportional to the standard deviation in the data. Combining standard deviation from the privacy with that of the data, we expect $\sqrt{\frac{c_0}{\epsilon^2} + c_1}$ to provide a good fit for pure differential privacy and $\sqrt{\frac{c_0}{\rho} + c_1}$ to be a good fit for zCDP. The result is a relatively good fit shown in Figure 19.

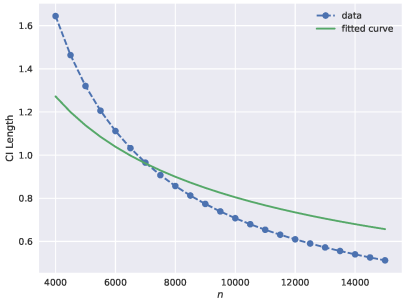
7.8. **Run Time.** We report the runtime to get one confidence interval from each of the methods on a laptop with a 2.7GHz processor and 8GB memory using Python. The ERM minimization was solved using L-BFGS. Times are averaged over 100 runs. The results show that SVM generally takes longer than logistic regression (LR) and all algorithms are reasonably fast for practical use.



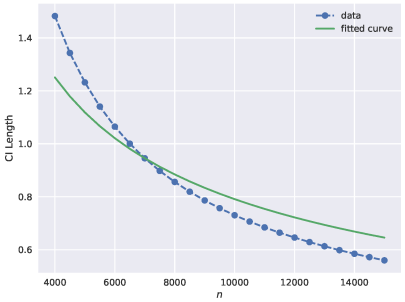
(A) ϵ-DP, objective perturbation, Adult



(B) ϵ-DP, output perturbation, US



(C) zCDP, objective perturbation, Banking



(D) zCDP, output perturbation, KDDCUP99

Figure 18: Relationship between average length of the confidence intervals and the sample size n for logistic regression. $d = 5$, $c = 0.001$, $\epsilon = 1.0$ (corresponds to $\rho = 0.5$). The fitted curve is $\frac{c}{\sqrt{n}}$, that shows the length for the privacy-preserving confidence intervals is not proportional to $\frac{1}{\sqrt{n}}$ as in the non-private case.

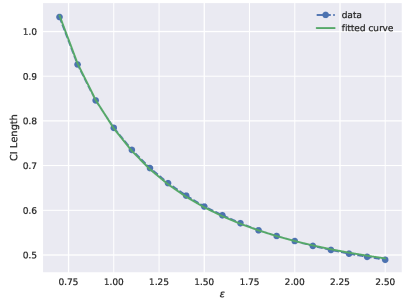
Table 3: Average runtime in seconds for one confidence interval

	pure-DP				zCDP			
	objective		output		objective		output	
	LR	SVM	LR	SVM	LR	SVM	LR	SVM
Adult, n =30,162 and $d = 10$	0.68	3.03	0.72	2.77	0.72	3.26	0.45	2.89
KDDCUP99, n =450,000 and $d = 10$	7.68	38.03	7.39	39.94	6.83	35.31	6.60	34.48

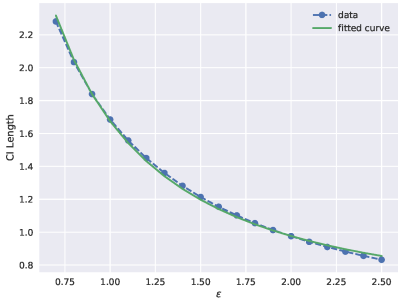
8. Conclusions

In this paper, we proposed algorithms for generating private confidence intervals for the coefficients of several private ERM algorithms (objective and output perturbation). Using the concept of variability intervals, we were able to empirically study how much these intervals increased as a result of privately estimating them.

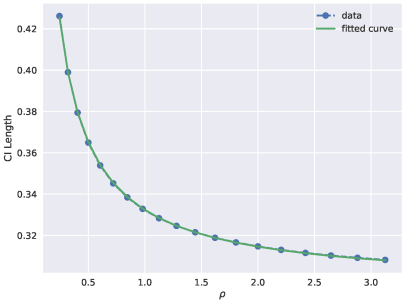
There are several interesting related open problems that are left for future work. These include a theoretical study on the length of these intervals along with lower bounds on their length at a desired coverage level. The algorithms we studied required convergence to optimality of the ERM subproblems they generated (i.e. optimal solution of the perturbed objective for the objective perturbation approach or, in the case of output perturbation, optimal solution of the non-private problem before noise is added to it). An interesting direction is to provide private confidence intervals for algorithms that stop early or use stochastic gradient descent.



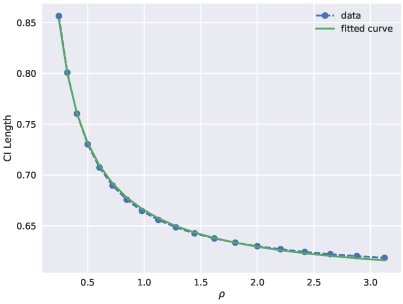
(A) ϵ-DP, objective perturbation, BR



(B) ϵ-DP, output perturbation, Adult



(C) zCDP, objective perturbation, US



(D) zCDP, output perturbation, KDDCUP99

Figure 19: Relationship between average length of the confidence intervals and the total privacy budget ϵ (or $\rho = \epsilon^2/2$) for logistic regression. $n = 10,000$, $d = 5$, $c = 0.001$. The fitted curve is $\sqrt{\frac{c_0}{\epsilon^2} + c_1}$ for ϵ-DP, $\sqrt{\frac{c_0}{\rho} + c_1}$ for zCDP.

References

Minnesota population center. integrated public use microdata series, international: Version 6.5 brazil, 2017a. URL http://doi.org/10.18128/D020.V6.5.

Ipums-usa, 2017b. URL www.ipums.org.

J. Acharya, Z. Sun, and H. Zhang. Differentially private testing of identity and closeness of discrete distributions. *arXiv preprint arXiv:1707.05128*, 2017.

A. F. Barrientos, J. P. Reiter, A. Machanavajjhala, and Y. Chen. Differentially private significance tests for regression coefficients. *arXiv preprint arXiv:1705.09561*, 2017.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 464–473. IEEE, 2014.

A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005.

M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

B. Cai, C. Daskalakis, and G. Kamath. Privit: Private and sample efficient identity testing. In *International Conference on Machine Learning*, pages 635–644, 2017.

O. Chapelle. Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178, 2007.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

K. Chaudhuri, A. Sarwate, and K. Sinha. Near-optimal differentially private principal components. In *Advances in Neural Information Processing Systems*, pages 989–997, 2012.

Y. Chen, A. Machanavajjhala, J. P. Reiter, and A. F. Barrientos. Differentially private regression diagnostics. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 81–90. IEEE, 2016.

V. D'Orazio, J. Honaker, and G. King. Differential privacy for social science inference. 2015.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006a.

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

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

A. Friedman and A. Schuster. Data mining with differential privacy. In *KDD*, 2010.

M. Gaboardi, H.-W. Lim, R. M. Rogers, and S. P. Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*. JMLR, 2016.

P. Jain and A. Thakurta. Differentially private learning with kernels. In *International Conference on Machine Learning*, pages 118–126, 2013.

P. Jain and A. G. Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484, 2014.

W. Jiang, C. Xie, and Z. Zhang. Wishart mechanism for differentially private principal components analysis. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

K. Kakizaki, K. Fukuchi, and J. Sakuma. Differentially private chi-squared test by unit circle mechanism. In *International Conference on Machine Learning*, pages 1761–1770, 2017.

V. Karwa and S. Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

S. P. Kasiviswanathan and H. Jin. Efficient private empirical risk minimization for high-dimensional learning. In *International Conference on Machine Learning*, pages 488–497, 2016.

S. P. Kasiviswanathan, K. Nissim, and H. Jin. Private incremental regression. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 167–182. ACM, 2017.

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

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

K. Ligett, S. Neel, A. Roth, B. Waggoner, and S. Z. Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. In *Advances in Neural Information Processing Systems*, pages 2563–2573, 2017.

S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

A. Rényi. On measures of entropy and information. Technical report, HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, 1961.

R. Rogers and D. Kifer. A new class of private chi-square hypothesis tests. In *Artificial Intelligence and Statistics*, pages 991–1000, 2017.

B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *arXiv preprint arXiv:0911.5708*, 2009.

O. Sheffet. Private approximations of the 2nd-moment matrix using existing techniques in linear regression. *arXiv preprint arXiv:1507.00056*, 2015.

O. Sheffet. Differentially private ordinary least squares. In *International Conference on Machine Learning*, pages 3105–3114, 2017.

K. Talwar, A. Thakurta, and L. Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

K. Talwar, A. G. Thakurta, and L. Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3025–3033, 2015.

C. Uhler, A. Slavković, and S. E. Fienberg. Privacy-preserving data sharing for genome-wide association studies. *The Journal of privacy and confidentiality*, 5(1):137, 2013.

D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2719–2728, 2017.

D. Wang, M. Gaboardi, and J. Xu. Efficient empirical risk minimization with smooth loss functions in non-interactive local differential privacy. *arXiv preprint arXiv:1802.04085*, 2018.

Y. Wang, J. Lee, and D. Kifer. Differentially private hypothesis testing, revisited. *ArXiv e-prints*, 2015.

X. Wu, M. Fredrikson, W. Wu, S. Jha, and J. F. Naughton. Revisiting differentially private regression: Lessons from learning theory and their consequences. *arXiv preprint arXiv:1512.06388*, 2015.

F. Yu, S. E. Fienberg, A. B. Slavković, and C. Uhler. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of biomedical informatics*, 50:133–141, 2014a.

F. Yu, M. Rybar, C. Uhler, and S. E. Fienberg. Differentially-private logistic regression for detecting multiple-snp association in gwas databases. In *International Conference on Privacy in Statistical Databases*, pages 170–184. Springer, 2014b.

J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. Privgene: differentially private model fitting using genetic algorithms. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 665–676. ACM, 2013.

J. Zhang, K. Zheng, W. Mou, and L. Wang. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3922–3928. AAAI Press, 2017.

Appendix A. Proofs

A.1. **Proof of Theorem 1.**

Theorem 1 . *If the loss function $f(\cdot)$ is convex and doubly differentiable, with $|f'(\cdot)| \leq 1$ and $|f''(\cdot)| \leq t$, then Algorithm 1 satisfies ϵ-differential privacy whenever all the feature vectors $\vec{x}_i$ have $||\vec{x}_i||_2 \leq 1$.*

Proof. The loss function $f(\vec{x}, y, \theta) = f(y\theta^T\vec{x})$. Because $f(\cdot)$ and $\|\theta\|_2^2$ are convex and differentiable, given any data set $\mathcal{D}$, the gradient of the objective function equals to 0 at the empirical minimizer $\tilde{\theta}$:

$$\beta = -2nc\tilde{\theta} - \sum_{i=1}^{n} y_i f'(y_i \tilde{\theta}^T \vec{x}_i)\vec{x}_i.$$

To show ϵ-differential privacy, we compute the ratio of the densities of $\tilde{\theta}$ under the two neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$.

$$\frac{g(\tilde{\theta}|\mathcal{D})}{g(\tilde{\theta}|\mathcal{D}')} = \frac{\mathtt{v}(\beta|\mathcal{D})}{\mathtt{v}(\beta'|\mathcal{D}')} \cdot \frac{|\det(\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D}))|^{-1}}{|\det(\mathbf{J}(\tilde{\theta} \to \beta'|\mathcal{D}'))|^{-1}},$$

where $\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D})$ is the Jacobian matrix of the mapping from $\tilde{\theta}$ to β.

Given $\mathcal{D}$, the (j,k)-th entry of $\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D})$ is

$$\frac{\partial \beta^{(j)}}{\partial \tilde{\theta}^{(k)}} = -2nc\mathbf{1}_{j=k} - \sum_{i=1}^{n} y_i^2 f''(y_i \tilde{\theta}^T \vec{x}_i) \vec{x}_i^{(j)} \vec{x}_i^{(k)},$$

where $\mathbf{1}$ is the indicator function. The Jacobian is well defined since $f(\cdot)$ is doubly differentiable.

Given $\mathcal{D}$ and $\mathcal{D}'$, define

$$\begin{aligned} A &= 2nc\mathbf{I}_d + \sum_{i=1}^{n-1} y_i^2 f''(y_i \tilde{\theta}^T \vec{x}_i) \vec{x}_i \vec{x}_i^T, \\ \vec{u}\vec{u}^T &= y_n^2 f''(y_n \tilde{\theta}^T \vec{x}_n) \vec{x}_n \vec{x}_n^T, \\ \vec{v}\vec{v}^T &= y_z^2 f''(y_z \tilde{\theta}^T \vec{x}_z) \vec{x}_z \vec{x}_z^T. \end{aligned}$$

Then, $\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D}) = -(A + \vec{u}\vec{u}^T)$, $\mathbf{J}(\tilde{\theta} \to \beta'|\mathcal{D}') = -(A + \vec{v}\vec{v}^T)$.

Therefore,

$$\begin{aligned} \frac{|\det(\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D}))|^{-1}}{|\det(\mathbf{J}(\tilde{\theta} \to \beta'|\mathcal{D}'))|^{-1}} &= \frac{|\det(A + \vec{v}\vec{v}^T)|}{|\det A + \vec{u}\vec{u}^T|} \\ &= \frac{|(1 + \vec{v}^T A^{-1} \vec{v}) \det(A)|}{|(1 + \vec{u}^T A^{-1} \vec{u}) \det(A)|} \\ &= \frac{|1 + \vec{v}^T A^{-1} \vec{v}|}{|1 + \vec{u}^T A^{-1} \vec{u}|}. \end{aligned}$$

We can see A is a symmetric positive definite matrix and its eigenvalues are at least $2nc$. So the eigenvalues of A^{-1} are at most $\frac{1}{2nc}$. Since $y \in \{-1, 1\}$, $\|\vec{x}\|_2 \leq 1$ and $f''(\cdot) \leq t$,

$$\begin{aligned} \frac{|1 + \vec{v}^T A^{-1} \vec{v}|}{|1 + \vec{u}^T A^{-1} \vec{u}|} &\leq \frac{1 + \|\vec{v}^T A^{-1} \vec{v}\|_2}{1} \\ &\qquad \text{(by the triangle inequality and } \vec{u}^T A^{-1} \vec{u} \geq 0) \\ &\leq 1 + \|v\|_2 \|A^{-1} \vec{v}\|_2 \leq 1 + \|v\|_2^2 \|A^{-1}\|_2 \\ &= 1 + y_z^2 f''(y_z \tilde{\theta}^T \vec{x}_z) \|\vec{x}_z\|_2^2 \|A^{-1}\|_2 \\ &\leq 1 + t\|A^{-1}\|_2 \leq 1 + \frac{t}{2nc}. \end{aligned}$$

Then by the definition of ϵ', $1 + \frac{t}{2nc} = e^{\epsilon - \epsilon'}$.

Next, we bound the ratio of the densities of the noise vectors:

$$\beta' - \beta = y_n f'(y_n \tilde{\theta}^T \vec{x}_n) \vec{x}_n - y_z f'(y_z \tilde{\theta}^T \vec{x}_z) \vec{x}_z.$$

Since $y \in \{-1, 1\}$, $\|\vec{x}\|_2 \leq 1$ and $f'(\cdot) \leq 1$,

$$\|\beta'\|_2 - \|\beta\|_2 \leq \|\beta' - \beta\|_2 \leq 2,$$

so

$$\frac{\mathtt{v}(\beta|\mathcal{D})}{\mathtt{v}(\beta'|\mathcal{D}')} = \frac{e^{-\epsilon'\|\beta\|_2/2}}{e^{-\epsilon'\|\beta'\|_2/2}} = e^{\epsilon'(\|\beta'\|_2 - \|\beta\|_2)/2} \leq e^{\epsilon'}.$$

Therefore,

$$\frac{g(\tilde{\theta}|\mathcal{D})}{g(\tilde{\theta}|\mathcal{D}')} = \frac{\mathrm{v}(\beta|\mathcal{D})}{\mathrm{v}(\beta'|\mathcal{D}')} \cdot \frac{|\det(\mathbf{J}(\tilde{\theta} \to \beta|\mathcal{D}))|^{-1}}{|\det(\mathbf{J}(\tilde{\theta} \to \beta'|\mathcal{D}'))|^{-1}} \leq e^{\epsilon'} \cdot e^{\epsilon-\epsilon'} = e^{\epsilon}.$$

□

A.2. **Proof of Lemma 1.**

Lemma 1. *Algorithm 2 satisfies ϕ-differential privacy and ϕ-zCDP.*

Proof. In Algorithm 2, only Line 8 touches the matrix M since Lines 1 through 7 are sampling from the noise distribution and all other lines are just post-processing on the perturbed matrix $\tilde{M}$. So we just need to prove getting $\tilde{M}$ through Line 8 satisfies differential privacy and zCDP.

(1) When protecting differential privacy:

To simplify, let $\mathrm{vec}\,(M)$ be the vector representation of M by stacking its rows. Let M' be the neighbor of M which differs in only one entry. Then given the density of the noise in Equation 3.2 and the L_2 sensitivity $Sens(M)$,

$$\begin{aligned}\frac{\mathrm{vec}\,(\tilde{M})|M}{\mathrm{vec}\,(\tilde{M})|M'} =& \frac{\mathrm{v}[\mathrm{vec}\,(\tilde{M}-M)]}{\mathrm{v}[\mathrm{vec}\,(\tilde{M}-M')]} \\ =& \frac{e^{-\phi/Sens(M)\cdot\|\,\mathrm{vec}\,(\tilde{M}-M)\|_2}}{e^{-\phi/Sens(M)\cdot\|\,\mathrm{vec}\,(\tilde{M}-M')\|_2}} \\ =& e^{\phi/Sens(M)\cdot(\|\,\mathrm{vec}\,(\tilde{M}-M')\|_2-\|\,\mathrm{vec}\,(\tilde{M}-M')\|_2)} \\ \leq& e^{\phi}.\end{aligned}$$

Therefore, Line 8 satisfies ϕ-differential privacy.

(2) When protecting zCDP:

By Proposition 3, Line 8 satisfies ϕ-zCDP.

The rest of the algorithm is just post-processing on the perturbed matrix. By the post-processing property of differential privacy and zCDP, Algorithm 2 satisfies ϕ-differential privacy and ϕ-zCDP. □

A.3. **Proof of Theorem 2.**

Theorem 2 . *Under the conditions of Theorem 1, Algorithm 3 satisfies $(\phi_1+\phi_2+\phi_3)$-differential privacy and $(\phi_1^2/2+\phi_2+\phi_3)$-zCDP.*

Proof. In Algorithm 3, there are three parts that touch the true data.

First, the computation of the minimizer $\tilde{\theta}$ to the objective function. Based on Theorem 1, the computation is ϕ_1-differentially private as long as the loss function $f(\cdot)$ is convex and doubly differentiable with $|f'(\cdot)| \leq 1$ and $|f''(\cdot)| \leq t$ for some finite t. By Proposition 1, the computation also satisfies $(\phi_1^2/2)$-zCDP.

The next two parts are the computations of the Hessian and the covariance matrix. By Lemma 1, the computation of the Hessian satisfies ϕ_2-differential privacy and ϕ_2-zCDP, and the computation of the covariance matrix satisfies ϕ_3-differential privacy and ϕ_3-zCDP.

All other computations are post-processing and therefore do not violate differential privacy or zCDP. By the composition theorem of differential privacy and zCDP, Algorithm 3 satisfies $(\phi_1+\phi_2+\phi_3)$-differential privacy and $(\phi_1^2/2+\phi_2+\phi_3)$-zCDP. □

A.4. **Proof of Theorem 4.**

Theorem 4 . *Under the same conditions as Theorem 3, Algorithm 5 satisfies* $(\phi_1+\phi_2+\phi_3)$*-differential privacy and* $(\phi_1+\phi_2+\phi_3)$*-zCDP.*

Proof. In Algorithm 5, there are three parts that touch the true data.

First, the computation of the minimizer to the objective function. By Theorem 3, this process satisfies ϕ_1-differential privacy and ϕ_1-zCDP as long as the loss function $f(\cdot)$ is convex and differentiable with $|f'(\cdot)| \leq 1$.

The next two parts are the computations of the Hessian and the covariance matrix. By Lemma 1, the computation of the Hessian satisfies ϕ_2-differential privacy and ϕ_2-zCDP, and the computation of the covariance matrix satisfies ϕ_3-differential privacy and ϕ_3-zCDP.

All other computations are post-processing and therefore do not violate differential privacy or zCDP. By the composition theorem of differential privacy and zCDP, Algorithm 5 satisfies $(\phi_1+\phi_2+\phi_3)$-differential privacy and $(\phi_1+\phi_2+\phi_3)$-zCDP. □

A.5. **Proof of Lemma 2.**

Lemma 2. *The* L_2*-sensitivity of the covariance matrix* Σ *(defined in Equation 4.3) for logistic regression is at most* $2S(\|\theta_0\|_2)^2/n$*.*

Proof. We compute the L_2 sensitivity for Σ as

$$\begin{aligned}
&\max_{\mathcal{D},\mathcal{D}'} \Big\| \operatorname{vec}\left(\Sigma_{\mathcal{D}} - \Sigma_{\mathcal{D}'}\right)\Big\|_2 \\
&= \max_{\vec{x}_n,y_n,\vec{x}_z,y_z} \frac{1}{n}\Big\| \operatorname{vec}\left[\nabla(f(\vec{x}_n,y_n,\theta_0))[\nabla f(\vec{x}_n,y_n,\theta_0)]^T\right] - \operatorname{vec}\left[\nabla(f(\vec{x}_z,y_z,\theta_0))[\nabla f(\vec{x}_z,y_z,\theta_0)]^T\right]\Big\|_2 \\
&= \max_{\vec{x}_n,y_n,\vec{x}_z,y_z} \frac{1}{n}\Big\| \left[S(-y_n\theta_0^T\vec{x}_n)^2 \operatorname{vec}\left(\vec{x}_n\vec{x}_n^T\right) - S(-y_z\theta_0^T\vec{x}_z)^2 \operatorname{vec}\left(\vec{x}_z\vec{x}_z^T\right)\right]\Big\|_2 \\
&\leq \max_{y,\vec{x}} \frac{2}{n} S(-y\theta_0^T\vec{x})^2 \| \operatorname{vec}\left(\vec{x}\vec{x}^T\right)\|_2.
\end{aligned}$$

Since $\|\vec{x}\|_2 \leq 1$, we get $\| \operatorname{vec}\left(\vec{x}\vec{x}^T\right)\|_2 = \sqrt{\sum_{1\leq j,k\leq d} x[j]^2x[k]^2} = \sqrt{(\sum_{j=1}^d x[j]^2)^2} \leq 1$. From the Cauchy-Schwarz inequality, we get $\|\theta^T\vec{x}\|_2 \leq \|\theta\|_2\|\vec{x}\|_2 \leq \|\theta\|_2$. We know either $\theta^T\vec{x} = \|\theta^T\vec{x}\|_2$ or $\theta^T\vec{x} = -\|\theta^T\vec{x}\|_2$, then $-\|\theta\|_2 \leq \theta^T\vec{x} \leq \|\theta\|_2$. Based on the fact that the sigmoid function $S(t)$ is monotonically increasing in t and $y \in \{-1,1\}$,

$$\max_{y,\vec{x}} \frac{2}{n} S(-y\theta_0^T\vec{x})^2 \| \operatorname{vec}\left(\vec{x}\vec{x}^T\right)\|_2 \leq \frac{2}{n} S(\|\theta_0\|_2)^2.$$

□

A.6. **Proof of Lemma 3.**

Lemma 3. *The* L_2*-sensitivity of the Hessian* $H[J_n(\tilde{\theta})]$ *(defined in Equation 4.2) for logistic regression is at most* $1/(2n)$*.*

Proof. We compute the L_2 sensitivity for $H[J_n(\tilde{\theta})]$ as:

$$\begin{aligned}
&\max_{\mathcal{D},\mathcal{D}'} \Big\| \operatorname{vec}\left(H[J_n(\mathcal{D},\tilde{\theta})] - H[J_n(\mathcal{D}',\tilde{\theta})]\right)\Big\|_2 \\
&= \max_{\vec{x}_n,y_n\vec{x}_z,y_z} \frac{1}{n}\Big\| \operatorname{vec}\Big[H[f(\vec{x}_n,y_n,\tilde{\theta})] - H[f(\vec{x}_z,y_z,\tilde{\theta})]\Big]\Big\|_2
\end{aligned}$$

$$
\begin{aligned}
&= \max_{\vec{x}_n, y_n \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\left[S(-y_n \tilde{\theta}^T \vec{x}_n) S(y_n \tilde{\theta}^T \vec{x}_n) \vec{x}_n \vec{x}_n^T \right] - \operatorname{vec}\left[S(-y_z \tilde{\theta}^T \vec{x}_z) S(y_z \tilde{\theta}^T \vec{x}_z) \vec{x}_z \vec{x}_z^T \right] \Big\|_2 \\
&\leq \max_{\vec{x}, y} \frac{2}{n} S(-y \tilde{\theta}^T \vec{x}) S(y \tilde{\theta}^T \vec{x}) \| \operatorname{vec}(\vec{x} \vec{x}^T) \|_2 \\
&\leq \max_{\vec{x}} \frac{2}{n} S(\tilde{\theta}^T \vec{x}) S(-\tilde{\theta}^T \vec{x}).
\end{aligned}
$$

In Appendix A.5, we have shown that $-\|\theta\|_2 \leq \theta^T \vec{x} \leq \|\theta\|_2$. The function $S(\tilde{\theta}^T \vec{x}) S(-\tilde{\theta}^T \vec{x})$ achieves the maximum $1/4$ at $\tilde{\theta}^T \vec{x} = 0$. So,

$$
\max_{\vec{x}} \frac{2}{n} S(\tilde{\theta}^T \vec{x}) S(-\tilde{\theta}^T \vec{x}) \leq \frac{1}{2n}.
$$

□

A.7. **Proof of Lemma 4.**

Lemma 4. *The L_2-sensitivity of the covariance matrix Σ (defined in Equation 4.3) for SVM is at most $2/n$.*

Proof. We compute the L_2 sensitivity for Σ as:

$$
\begin{aligned}
&\max_{\mathcal{D}, \mathcal{D}'} \Big\| \operatorname{vec}\left(\Sigma'_{\mathcal{D}} - \Sigma'_{\mathcal{D}'} \right) \Big\|_2 \\
&= \max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}_n, y_n, \theta_0) [\nabla f(\vec{x}_n, y_n, \theta_0)]^T \right] - \operatorname{vec}\left[\nabla f(\vec{x}_z, y_z, \theta_0) [\nabla f(\vec{x}_z, y_z, \theta_0)]^T \right] \Big\|_2 \\
&\leq \max_{y, \vec{x}} \frac{2}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}, y, \theta_0) [\nabla f(\vec{x}, y, \theta_0)]^T \right] \Big\|_2.
\end{aligned}
$$

There are three cases:

(1) If $y \theta^T \vec{x} > 1 + h$,

$$
\max_{y, \vec{x}} \frac{2}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}, y, \theta_0) [\nabla f(\vec{x}, y, \theta_0)]^T \right] \Big\|_2 = 0.
$$

(2) If $|1 - y \theta^T \vec{x}| \leq h$,

$$
\begin{aligned}
&\max_{y, \vec{x}} \frac{2}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}, y, \theta_0) [\nabla f(\vec{x}, y, \theta_0)]^T \right] \Big\|_2 \\
=&\max_{y, \vec{x}} \frac{2}{n} \left[\frac{y}{2h} (y \theta_0^T \vec{x} - 1 - h) \right]^2 \| \operatorname{vec}(\vec{x} \vec{x}^T) \|_2 \\
\leq&\max_{y, \vec{x}} \frac{1}{2nh^2} (y \theta_0^T \vec{x} - 1 - h)^2 \\
\leq&\frac{1}{2nh^2} \cdot 4h^2 \quad \text{since } y \theta_0^T \vec{x} - 1 \in [-h, h] \\
=&2/n.
\end{aligned}
$$

(3) If $y \theta^T \vec{x} < 1 - h$,

$$
\begin{aligned}
&\max_{y, \vec{x}} \frac{2}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}, y, \theta_0) [\nabla f(\vec{x}, y, \theta_0)]^T \right] \Big\|_2 \\
=&\max_{y, \vec{x}} \frac{2}{n} y^2 \| \operatorname{vec}(\vec{x} \vec{x}^T) \|_2 \\
\leq&2/n.
\end{aligned}
$$

Therefore, in all cases,

$$\max_{y,\vec{x}} \frac{2}{n} \Big\| \operatorname{vec}\left[\nabla f(\vec{x}, y, \theta_0)[\nabla f(\vec{x}, y, \theta_0)]^T\right] \Big\|_2 \leq 2/n.$$

□

A.8. **Proof of Lemma 5.**

Lemma 5. *The L_2-sensitivity of the Hessian $H[J_n(\tilde{\theta})]$ (defined in Equation 4.2) for SVM is at most $1/(nh)$.*

Proof. The L_2 sensitivity for $H[J_n(\tilde{\theta})]$ can be computed as:

$$\begin{aligned}
&\max_{\mathcal{D},\mathcal{D}'} \Big\| \operatorname{vec}\{H[J_n(\mathcal{D}, \tilde{\theta})] - H[J_n(\mathcal{D}', \tilde{\theta})]\} \Big\|_2 \\
=&\max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\{H[f(y_n \tilde{\theta} \vec{x}_n)] - H[f(y_z \tilde{\theta} \vec{x}_z)]\} \Big\|_2.
\end{aligned}$$

There are two cases:

(1) If $|1 - y\theta^T \vec{x}| \leq h$,

$$\begin{aligned}
&\max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\{H[f(y_n \tilde{\theta} \vec{x}_n)] - H[f(y_z \tilde{\theta} \vec{x}_z)]\} \Big\|_2 \\
=&\max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\left\{ \frac{y_n^2}{2h} \vec{x}_n \vec{x}_n^T - \frac{y_z^2}{2h} \vec{x}_z \vec{x}_z^T \right\} \Big\|_2 \\
\leq&\max_{\vec{x}, y} \frac{2}{n} \cdot \frac{1}{2h} \| \operatorname{vec}(\vec{x}\vec{x}^T) \|_2 \\
\leq&1/(nh).
\end{aligned}$$

(2) Otherwise,

$$\max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\{H[f(y_n \tilde{\theta} \vec{x}_n)] - H[f(y_z \tilde{\theta} \vec{x}_z)]\} \Big\|_2 = 0.$$

Therefore, in all cases,

$$\max_{\vec{x}_n, y_n, \vec{x}_z, y_z} \frac{1}{n} \Big\| \operatorname{vec}\{H[f(y_n \tilde{\theta} \vec{x}_n)] - H[f(y_z \tilde{\theta} \vec{x}_z)]\} \Big\|_2 \leq 1/(nh).$$

□