Differentially Private Empirical Risk Minimization

Kamalika Chaudhuri[*], Claire Monteleoni[†], Anand D. Sarwate[‡]

June 1, 2010

Abstract

Privacy-preserving machine learning algorithms are crucial for the increasingly common setting in which personal data, such as medical or financial records, are analyzed. We provide general techniques to produce privacy-preserving approximations of classifiers learned via (regularized) empirical risk minimization (ERM). These algorithms are private under the ϵ-*differential privacy* definition due to Dwork et al. (2006). First we apply the output perturbation ideas of Dwork et al. (2006), to ERM classification. Then we propose a new method, *objective perturbation*, for privacy-preserving machine learning algorithm design. This method entails perturbing the objective function before optimizing over classifiers. We prove theoretical results showing that our algorithms preserve privacy and give generalization bounds for linear and nonlinear kernels. We further present a privacy-preserving technique for tuning the parameters in general machine learning algorithms, thereby providing end-to-end privacy guarantees for the training process. We apply these results to produce privacy-preserving analogues of regularized logistic regression and support vector machines. We obtain encouraging results from evaluating their performance on real demographic and benchmark data sets. Our results show that both theoretically and empirically, objective perturbation is superior to the previous state-of-the-art, output perturbation, in managing the inherent tradeoff between privacy and learning performance.

1 Introduction

Privacy has become a growing concern, due to the massive increase in personal information stored in electronic databases, such as medical records, financial records, web search histories, and social network data. Machine learning can be employed to discover novel population-wide patterns, however the results of such algorithms may reveal certain individuals' sensitive information, thereby violating their privacy. Thus, an emerging challenge for machine learning is how to learn from datasets that contain sensitive personal information.

At the first glance, it may appear that simple anonymization of private information is enough to preserve privacy. However, this is often not the case; even if obvious identifiers, such as names and addresses, are removed from the data, the remaining fields can still form unique "signatures" that can help re-identify individuals. Such attacks have been demonstrated by various works, and are

[*] Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093, USA. Email: kchaudhuri@ucsd.edu

[†] Center for Computational Learning Systems, Columbia University, New York, NY 10115, USA. Email: cmontel@ccls.columbia.edu

[‡] Information Theory and Applications Center, University of California, San Diego, La Jolla, CA 92093-0447, USA. Email: asarwate@ucsd.edu

possible in many realistic settings, such as when an adversary has side information [Sweeney, 1997, Narayanan and Shmatikov, 2008, Ganta et al., 2008], and when the data has structural properties [Backstrom et al., 2007], among others. Moreover, even releasing statistics on a sensitive dataset may not be sufficient to preserve privacy, as illustrated on genetic data [Homer et al., 2008, Wang et al., 2009]. Thus, there is a great need for designing machine learning algorithms, that also preserve the privacy of individuals on the datasets they train and operate on.

In this paper, we focus on the problem of classification, one of the fundamental problems of machine learning, when the training data consists of sensitive information of individuals. Our work addresses the Empirical risk minimization (ERM) framework for classification, in which a classifier is chosen by minimizing the average over the training data of the prediction loss (with respect to the label) of the classifier in predicting each training data point. In this work, we focus on regularized ERM in which there is an additional term in the optimization, called the regularizer, which penalizes the complexity of the classifier, with respect to some metric. Regularized ERM methods are widely used in practice, for example logistic regression and SVM, and many also have theoretical justification in the form of generalization error bounds with respect to independently, identically distributed (i.i.d.) data (see Vapnik [1998] for further details).

For our privacy measure, we use a definition due to Dwork et al. [2006b], who have proposed a measure of quantifying the privacy-risk associated with computing functions of sensitive data. Their *ϵ-differential privacy* model is a strong, cryptographically-motivated definition of privacy that has recently received a significant amount of research attention for its robustness to known attacks, such as those involving side information [Ganta et al., 2008]. Algorithms satisfying ϵ-differential privacy are randomized; the output is a random variable whose distribution is conditioned on the data set. A statistical procedure satisfies ϵ-differential privacy if changing a single data point does not shift the output distribution by too much. Therefore, from looking at the output the algorithm, it is difficult to infer the value of any particular data point.

In this paper, we develop methods for approximating empirical risk minimization (ERM) while guaranteeing ϵ-differential privacy. We develop methods for *end-to-end privacy*; each step in the learning process can cause additional risk of privacy violation, and we provide algorithms with quantifiable privacy guarantees for training as well as parameter tuning. For training, we provide two privacy-preserving approximations to ERM. The first is *output perturbation*, based on the *sensitivity method* proposed by Dwork et al. [2006b]. In this method noise is added to the output of the standard ERM algorithm. The second method is novel, and involves adding noise to the regularized ERM objective function prior to minimizing. We call this second method *objective perturbation*. We show theoretical bounds for both procedures; the theoretical performance of objective perturbation is superior to that of output perturbation for most problems. We validate these theoretical results on data sets from the UCI repository.

In practice, parameters in learning algorithms are chosen via a holdout data set. In the context of privacy, we must guarantee the privacy of the holdout data as well. We exploit results from the theory of differential privacy to develop a privacy-preserving parameter tuning algorithm, and demonstrate its use in practice. Together with our training algorithms, this parameter tuning algorithm guarantees privacy to all data used in the learning process.

Guaranteeing privacy incurs a cost in performance; because the algorithms must cause some uncertainty in the output, they increase the loss of the output predictor. Because the ϵ-differential privacy model requires robustness against all data sets, we make no assumptions on the underlying data for the purposes of making privacy guarantees. However, to prove the impact of privacy constraints on the generalization error, we assume the data is i.i.d. according to a fixed but unknown distribution. Although many of our results hold for ERM in general, we provide specific results for classification using logistic regression and support vector machines. Some of the former results were reported in Chaudhuri and Monteleoni [2008]; here we generalize them to ERM and extend the results to kernel methods, and provide experiments on real datasets.

More specifically, the contributions of this paper are as follows:

- We derive a computationally efficient algorithm for ERM classification, based on the sensitivity method due to Dwork et al. [2006b]. We analyze the accuracy of this algorithm, and provide an upper bound on the number of training samples required by this algorithm to achieve a fixed generalization error.

- We provide a general technique, *objective perturbation*, for providing computationally efficient, differentially private approximations to regularized ERM algorithms. This extends the work of Chaudhuri and Monteleoni [2008], which follows as a special case. We specialize the results on the sensitivity method and objective perturbation to logistic regression and support vector machine classifiers. In addition to privacy guarantees, we also provide generalization bounds for this algorithm.

- For kernel methods with nonlinear kernel functions, the optimal classifier is a linear combination of kernel functions centered at the training points. This form is inherently non-private because it reveals the training data. We adapt a random projection method due to Rahimi and Recht Rahimi and Recht [2007, 2008b], to develop privacy-preserving kernel-ERM algorithms. We provide theoretical results on generalization performance.

- Because the holdout data is used in the process of training and releasing a classifier, we provide a privacy-preserving parameter tuning algorithm based on a randomized selection procedure [McSherry and Talwar, 2007] applicable to general machine learning algorithms. This guarantees end-to-end privacy during the learning procedure.

- We validate our results using experiments on two datasets from the UCI Machine Learning repositories (Asuncion and Newman [2007]) and KDDCup (Hettich and Bay [1999]). Our results show that objective perturbation is generally superior to output perturbation. We also demonstrate the impact of end-to-end privacy on generalization error.

1.1 Related Work

There has been a significant amount of literature on the ineffectiveness of simple anonymization procedures. For example, Narayanan and Shmatikov [2008] show that a small amount of auxiliary information (knowledge of a few movie-ratings, and approximate dates) is sufficient for an adversary to re-identify an individual in the Netflix dataset, which consists of anonymized data about Netflix users and their movie ratings. The same phenomenon has been observed in other kinds of data, such as social network graphs (Backstrom et al. [2007]), search query logs (Jones et al. [2007]) and others. Releasing statistics computed on sensitive data can also be problematic; for example, Wang et al. [2009] show that releasing R^2-values computed on high-dimensional genetic data can lead to privacy breaches by an adversary who is armed with a small amount of auxiliary information.

There has also been a significant amount of work on privacy-preserving data mining [Agrawal and Srikant, 2000, Evfimievski et al., 2003, Sweeney, 2002, Machanavajjhala et al., 2006], spanning several communities, that uses privacy models other than differential privacy. Many of the models used have been shown to be susceptible to *composition attacks*, attacks in which the adversary has some reasonable amount of prior knowledge [Ganta et al., 2008]. Other work [Mangasarian et al., 2008] considers the problem of privacy-preserving SVM classification when separate agents have to share private data, and provides a solution that uses random kernels, but does provide any formal privacy guarantee.

An alternative line of privacy work is in the Secure Multiparty Computation setting due to Yao [1982], where the sensitive data is split across multiple hostile databases, and the goal is to compute a function on the union of these databases. Zhan and Matwin [2007] and Laur et al. [2006] consider computing privacy-preserving SVMs in this setting, and their goal is to design a distributed protocol to learn a classifier. This is in contrast with our work, which deals with a setting where the algorithm has access to the entire dataset.

Differential privacy, the formal privacy definition used in our paper, was proposed by the seminal work of Dwork et al. [2006b], and has been used since in numerous works on privacy for example, Chaudhuri and Mishra [2006], McSherry and Talwar [2007], Nissim et al. [2007], Barak et al. [2007], Chaudhuri and Monteleoni [2008], Machanavajjhala et al. [2008]. Unlike many other privacy definitions, such as those mentioned above, differential privacy has been shown to be resistant to composition attacks (attacks involving side-information) [Ganta et al., 2008]. Some follow-up work on differential privacy includes work on differentially-private combinatorial optimization, due to Gupta et al. [2010], and differentially-private contingency tables, due to Barak et al. [2007] and Kasivishwanathan et al. [2010]. Wasserman and Zhou [2010] provide a more statistical view of differential privacy, and Zhou et al. [2009b] provide a technique of generating synthetic data using compression via random linear or affine transformations.

Previous literature has also considered learning with differential privacy. One of the first such works is Kasiviswanathan et al. [2008], which presents a general, although computationally inefficient, method for PAC-learning finite concept classes. Blum et al. [2008] presents a method for releasing a database in a differentially-private manner, so that certain fixed classes of queries can be answered accurately, provided the class of queries has a bounded VC-dimension. Their methods can also be used to learn classifiers with a fixed VC-dimension – see Kasiviswanathan et al. [2008]; however the resulting algorithm is also computationally inefficient. Some sample complexity lower bounds in this setting have been provided by Beimel et al. [2010]. In addition, Dwork and Lei [2009] explore a connection between differential privacy and robust statistics, and provide an algorithm for privacy-preserving regression using ideas from robust statistics. However, their algorithm also requires a running time which is exponential in the data dimension, and is hence computationally inefficient.

This work builds on our preliminary work in Chaudhuri and Monteleoni [2008]. We first show how to extend the sensitivity method, a form of *output perturbation*, due to Dwork et al. [2006b], to classification algorithms. In general, output perturbation methods alter the output of the function computed on the database, before releasing it; in particular the sensitivity method makes an algorithm differentially private by adding noise to its output. In the classification setting, the noise protects the privacy of the training data, but increases the prediction error of the classifier.[1] In

[1] Recently, independent work by Rubinstein et al. [2009] has reported an extension of the sensitivity method to linear and kernel SVMs. Their utility analysis differs from ours, and thus the analogous generalization bounds are not comparable. Because Rubinstein et al. use techniques from algorithmic stability, their utility bounds compare the private and non-private classifiers using the same value for the regularization parameter. In contrast, our approach takes into account how the value of the regularization parameter might change due to privacy constraints.

contrast, we propose the *objective perturbation* method, in which noise is added to the *objective function* before optimizing over the space classifiers. Both the sensitivity method and objective perturbation result in computationally efficient algorithms, for our specific case. In general, our theoretical bounds on sample requirement are incomparable with the bounds of Kasiviswanathan et al. [2008] because of the difference between their setting and ours.

2 Model

We will use $\|\mathbf{x}\|$, $\|\mathbf{x}\|_\infty$, and $\|\mathbf{x}\|_\mathcal{H}$ to denote the ℓ_2-norm, ℓ_∞-norm, and norm in a Hilbert space $\mathcal{H}$, respectively. For an integer n we will use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. Vectors will typically be written in boldface and sets in calligraphic type. For a matrix A, we will use the notation $\|A\|_2$ to denote the L_2 norm of A.

2.1 Empirical Risk Minimization

In this paper we develop privacy-preserving algorithms for *regularized empirical risk minimization*, a special case of which is learning a classifier from labeled examples. We will phrase our problem in terms of classification and indicate when more general results hold. Our algorithms take as input *training data* $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, 2, \ldots, n\}$ of n data-label pairs. In the case of binary classification the data space $\mathcal{X} = \mathbb{R}^d$ and the label set $\mathcal{Y} = \{-1, +1\}$. We will assume throughout that $\mathcal{X}$ is the unit ball so that $\|\mathbf{x}_i\|_2 \leq 1$.

We would like to produce a *predictor* $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$. We measure the quality of our predictor on the training data via a nonnegative *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

In regularized empirical risk minimization (ERM), we choose a predictor $\mathbf{f}$ that minimizes the regularized empirical loss:

$$J(\mathbf{f}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i) + \Lambda N(\mathbf{f}). \tag{1}$$

This minimization is performed over $\mathbf{f}$ in an hypothesis class $\mathcal{H}$. The regularizer $N(\cdot)$ prevents over-fitting. For the first part of this paper we will restrict our attention to linear predictors and with some abuse of notation we will write $\mathbf{f}(\mathbf{x}) = \mathbf{f}^T\mathbf{x}$.

2.2 Assumptions on loss and regularizer

The conditions under which we can prove results on privacy and generalization error depend on analytic properties of the loss and regularizer. In particular, we will require certain forms of convexity (see Rockafellar and Wets [1998]).

Definition 1. *A function $H(\mathbf{f})$ over $\mathbf{f} \in \mathbb{R}^d$ is said to be* strictly convex *if for all $\alpha \in (0, 1)$, $\mathbf{f}$, and $\mathbf{g}$,*

$$H(\alpha\mathbf{f} + (1 - \alpha)\mathbf{g}) < \alpha H(\mathbf{f}) + (1 - \alpha)H(\mathbf{g}). \tag{2}$$

It is said to be λ-strongly convex *if for all $\alpha \in (0, 1)$, $\mathbf{f}$, and $\mathbf{g}$,*

$$H(\alpha\mathbf{f} + (1 - \alpha)\mathbf{g}) \leq \alpha \frac{H(\mathbf{f}) + H(\mathbf{g})}{2} - \lambda \frac{1}{2} \alpha(1 - \alpha) \|\mathbf{f} - \mathbf{g}\|_2^2 . \tag{3}$$

A strictly convex function has a unique minimum – see Boyd and Vandenberghe [2004]. Strong convexity plays a role in guaranteeing our privacy and generalization requirements. For our privacy results to hold we will also require that the regularizer $N(\cdot)$ and loss function $\ell(\cdot, \cdot)$ be differentiable functions of $\mathbf{f}$. This excludes certain classes of regularizers, such as the ℓ_1-norm regularizer $N(\mathbf{f}) = \|\mathbf{f}\|_1$, and classes of loss functions such as the hinge loss $\ell_{\mathrm{SVM}}(\mathbf{f}^T\mathbf{x}, y) = (1 - y\mathbf{f}^T\mathbf{x})^+$. In some cases we can prove privacy guarantees for approximations to these non-differentiable functions.

2.3 Privacy model

We are interested in producing a classifier in a manner that preserves the privacy of individual entries of the dataset $\mathcal{D}$ that is used in training the classifier. The notion of privacy we use is the *ϵ-differential privacy model*, developed by Dwork et al. [2006b], Dwork [2006], which defines a notion of privacy for a randomized algorithm $\mathcal{A}(\mathcal{D})$. Suppose $\mathcal{A}(\mathcal{D})$ produces a classifier, and let $\mathcal{D}'$ be another dataset that differs from $\mathcal{D}$ in one entry (which we assume is the private value of one person). That is, $\mathcal{D}'$ and $\mathcal{D}$ have $n - 1$ points $(\mathbf{x}_i, y_i)$ in common. The algorithm $\mathcal{A}$ provides differential privacy if for any set $\mathcal{S}$, the likelihood that $\mathcal{A}(\mathcal{D}) \in \mathcal{S}$ is close to the likelihood $\mathcal{A}(\mathcal{D}') \in \mathcal{S}$, (where the likelihood is over the randomness in the algorithm). That is, any single entry of the dataset does not affect the output distribution of the algorithm by much; dually, this means that an adversary, who knows all but one entry of the dataset, cannot gain much additional information about the last entry by observing the output of the algorithm.

The following definition of differential privacy is due to Dwork et al. [2006b], paraphrased from Wasserman and Zhou [2010].

Definition 2. *An algorithm $\mathcal{A}(\mathcal{B})$ taking values in a set $\mathcal{T}$ provides ϵ_p-differential privacy if*

$$\sup_{\mathcal{S}} \sup_{\mathcal{D}, \mathcal{D}'} \frac{\mu\left(\mathcal{S} \mid \mathcal{B} = \mathcal{D}\right)}{\mu\left(\mathcal{S} \mid \mathcal{B} = \mathcal{D}'\right)} \leq e^{\epsilon_p}, \tag{4}$$

where the first supremum is over all measurable $\mathcal{S} \subseteq \mathcal{T}$, the second is over all datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in a single entry, and $\mu(\cdot|\mathcal{B})$ is the conditional distribution (measure) on $\mathcal{T}$ induced by the output $\mathcal{A}(\mathcal{B})$ given a dataset $\mathcal{B}$. The ratio is interpreted to be 1 whenever the numerator and denominator are both 0.

Note that if $\mathcal{S}$ is a set of measure 0 under the conditional measures induced by $\mathcal{D}$ and $\mathcal{D}'$, the ratio is automatically 1. A more measure-theoretic definition is given in Zhou et al. [2009a]. An illustration of the definition is given in Figure 1.

The following form of the definition is due to Dwork et al. [2006a].

Definition 3. *An algorithm $\mathcal{A}$ provides ϵ_p-differential privacy if for any two datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ in a single entry and for any set $\mathcal{S}$,*

$$\exp(-\epsilon_p)\mathbb{P}(\mathcal{A}(\mathcal{D}') \in \mathcal{S}) \leq \mathbb{P}(\mathcal{A}(\mathcal{D}) \in \mathcal{S}) \leq \exp(\epsilon_p)\mathbb{P}(\mathcal{A}(\mathcal{D}') \in \mathcal{S}), \tag{5}$$

where $\mathcal{A}(\mathcal{D})$ (resp. $\mathcal{A}(\mathcal{D}')$) is the output of $\mathcal{A}$ on input $\mathcal{D}$ (resp. $\mathcal{D}'$).

We observe that an algorithm $\mathcal{A}$ that satisfies Equation 4 also satisfies Equation 5, and as a result, Definition 2 is stronger than Definition 3.

From this definition, it is clear that the $\mathcal{A}(\mathcal{D})$ that outputs the minimizer of the ERM objective (1) does not provide ϵ_p-differential privacy for any ϵ_p. This is because an ERM solution is a linear

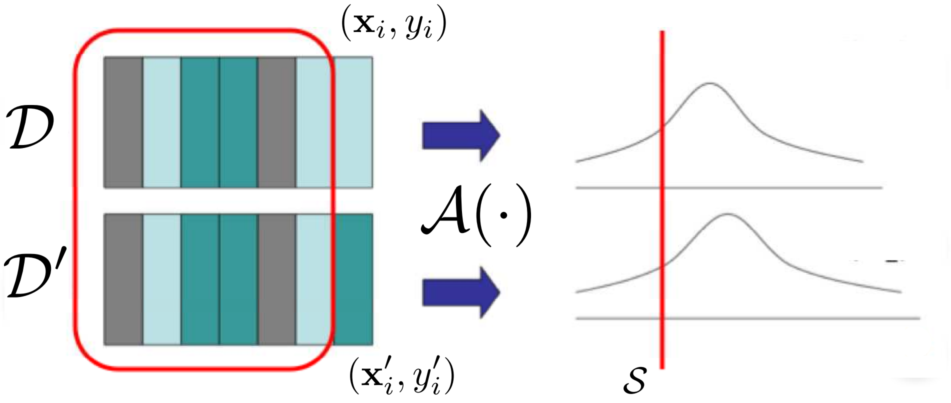


Figure 1: An algorithm which is differentially private. When datasets which are identical except for a single entry are input to the algorithm $\mathcal{A}$, the two distributions on the algorithm's output are close. For a fixed measurable $\mathcal{S}$ the ratio of the measures (or densities) should be bounded.

combination of some selected training samples "near" the decision boundary. If $\mathcal{D}$ and $\mathcal{D}'$ differ in one of these samples, then the classifier will change completely, making the likelihood ratio in (5) infinite. Regularization helps by bounding the L_2 norm of the change, but does not completely solve the problem.

Dwork et al. [2006b] also provide a standard recipe for computing privacy-preserving approximations to functions by adding noise with a particular distribution to the output of the function. We call this recipe the *sensitivity method.* Let $g : (\mathbb{R}^m)^n \rightarrow \mathbb{R}$ be a scalar function of $z_1, \ldots, z_n$, where $z_i \in \mathbb{R}^m$ corresponds to the private value of individual i; then the sensitivity of g is defined as follows.

Definition 4. *The sensitivity of a function $g : (\mathbb{R}^m)^n \rightarrow \mathbb{R}$ is maximum difference between the values of the function when one input changes. More formally, the sensitivity $S(g)$ of g is defined as:*

$$S(g) = \max_{i} \max_{z_1,\ldots,z_n,z_i'} \left| g(z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_n) - g(z_1, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_n) \right|. \quad (6)$$

To compute a function g on a dataset $\mathcal{D} = \{z_1, \ldots, z_n\}$, the sensitivity method outputs $g(z_1, \ldots, z_n) + \eta$, where η is a random variable drawn according to the Laplace distribution, with mean 0 and standard deviation $\frac{S(g)}{\epsilon_p}$. It is shown in Dwork et al. [2006b] that such a procedure is ϵ_p-differentially private.

3 Privacy-preserving ERM

Here we describe two approaches for creating privacy-preserving algorithms from (1).

3.1 Output perturbation : the sensitivity method

Algorithm 1 is derived from the *sensitivity method* of Dwork et al. [2006b], a general method for generating a privacy-preserving approximation to any function $A(\cdot)$. In this section the norm $\|\cdot\|$

is the L_2-norm unless otherwise specified. For the function $A(\mathcal{D}) = \operatorname{argmin} J(\mathbf{f}, \mathcal{D})$, Algorithm 1 outputs a vector $A(\mathcal{D}) + \mathbf{b}$, where $\mathbf{b}$ is random noise with density

$$\nu(\mathbf{b}) = \frac{1}{\alpha} e^{-\beta \|\mathbf{b}\|} \ , \tag{7}$$

where α is a normalizing constant. The parameter β is a function of ϵ_p, and the L_2-*sensitivity* of $A(\cdot)$, which is defined as follows.

Definition 5. *The L_2-sensitivity of a vector-valued function is defined as the maximum change in the L_2 norm of the value of the function when one input changes. More formally,*

$$S(A) = \max_i \max_{z_1, \ldots, z_n, z_i'} \left\| A(z_1, \ldots, z_i, \ldots) - A(z_1, \ldots, z_i', \ldots) \right\| . \tag{8}$$

The interested reader is referred to Dwork et al. [2006b] for further details. Adding noise to the output of $A(\cdot)$ has the effect of masking the effect of any particular data point. However, in some applications the sensitivity of the minimizer $\operatorname{argmin} J(\mathbf{f}, \mathcal{D})$ may be quite high, which would require the sensitivity method to add noise with high variance.

Algorithm 1 ERM with output perturbation (sensitivity)

```
Inputs: Data $\mathcal{D} = \{z_i\}$, parameters $\epsilon_p$, $\Lambda$.
Output: Approximate minimizer $\mathbf{f}_{\text{priv}}$.
Draw a vector $\mathbf{b}$ according to (7) with $\beta = \frac{n \Lambda \epsilon_p}{2}$.
Compute $\mathbf{f}_{\text{priv}} = \operatorname{argmin} J(\mathbf{f}, \mathcal{D}) + \mathbf{b}$.
```

3.2 Objective perturbation

A different approach, first proposed by Chaudhuri and Monteleoni [2008], is to add noise to the objective function itself and then produce the minimizer of the perturbed objective. That is, we can minimize

$$J_{\text{priv}}(\mathbf{f}, \mathcal{D}) = J(\mathbf{f}, \mathcal{D}) + \frac{1}{n} \mathbf{b}^T \mathbf{f}, \tag{9}$$

where $\mathbf{b}$ has density given by (7), with $\beta = \epsilon_p$. Note that the privacy parameter here does not depend on the sensitivity of the of the classification algorithm.

Algorithm 2 ERM with objective perturbation

```
Inputs: Data $\mathcal{D} = \{z_i\}$, parameters $\epsilon_p$, $\Lambda$, $c$.
Output: Approximate minimizer $\mathbf{f}_{\text{priv}}$.
Let $\epsilon_p' = \epsilon_p - \log(1 + \frac{2c}{n\Lambda} + \frac{c^2}{n^2 \Lambda^2})$.
If $\epsilon_p' > 0$, then $\Delta = 0$, else $\Delta = \frac{c}{n(e^{\epsilon_p/4} - 1)} - \Lambda$, and $\epsilon_p' = \epsilon_p / 2$.
Draw a vector $\mathbf{b}$ according to (7) with $\beta = \epsilon_p'/2$.
Compute $\mathbf{f}_{\text{priv}} = \operatorname{argmin} J_{\text{priv}}(\mathbf{f}, \mathcal{D}) + \frac{1}{2} \Delta ||\mathbf{f}||^2$.
```

The algorithm requires a certain slack, $\log(1 + \frac{2c}{n\Lambda} + \frac{c^2}{n^2 \Lambda^2})$, in the privacy parameter. This is due to additional factors in bounding the ratio of the densities.

3.3 Privacy guarantees

In this section, we establish the conditions under which Algorithms 1 and 2 provide ϵ_p-differential privacy. First, we establish guarantees for Algorithm 1.

3.3.1 Privacy Guarantees for Output Perturbation

Theorem 1. *If $N(\cdot)$ is differentiable, and 1-strongly convex, and ℓ is convex and differentiable, with $|\ell'(z)| \leq 1$ for all z, then, Algorithm 1 provides ϵ_p-differential privacy.*

The proof of Theorem 1 follows from Corollary 1, and Dwork et al. [2006b]. The proof is provided here for completeness.

Proof. From Corollary 1, if the conditions on $N(\cdot)$ and ℓ hold, then the L_2-sensivity of ERM with regularization parameter Λ is at most $\frac{2}{n\Lambda}$. We observe that when we pick $||\mathbf{b}||$ from the distribution in Algorithm 1, for a specific vector $\mathbf{b_0} \in \mathbb{R}^d$, the density at $\mathbf{b_0}$ is proportional to $e^{-\frac{n\Lambda\epsilon_p}{2}||\mathbf{b_0}||}$. Let $\mathcal{D}$ and $\mathcal{D}'$ be any two datasets that differ in the value of one individual. Then, for any $\mathbf{f}$,

$$\frac{g(\mathbf{f}|\mathcal{D})}{g(\mathbf{f}|\mathcal{D}')} = \frac{\nu(\mathbf{b}_1)}{\nu(\mathbf{b}_2)} = e^{-\frac{n\Lambda\epsilon_p}{2}(||\mathbf{b}_1||-||\mathbf{b}_2||)}, \tag{10}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the corresponding noise vectors chosen in Step 1 of Algorithm 1, and $g(\mathbf{f}|\mathcal{D})$ ($g(\mathbf{f}|\mathcal{D}')$ resp.) is the density of the output of Algorithm 1 at $\mathbf{f}$, when the input is $\mathcal{D}$ ($\mathcal{D}'$ resp.). If $\mathbf{f}_1$ and $\mathbf{f}_2$ are the solutions respectively to non-private regularized ERM when the input is $\mathcal{D}$ and $\mathcal{D}'$, then, $\mathbf{b}_2 - \mathbf{b}_1 = \mathbf{f}_2 - \mathbf{f}_1$. From Corollary 1, and using a triangle inequality,

$$||\mathbf{b}_1|| - ||\mathbf{b}_2|| \leq ||\mathbf{b}_1 - \mathbf{b}_2|| = ||\mathbf{f}_2 - \mathbf{f}_1|| \leq \frac{2}{n\Lambda}. \tag{11}$$

Moreover, by symmetry, the density of the directions of $\mathbf{b}_1$ and $\mathbf{b}_2$ are uniform. Therefore, by construction, $\frac{\nu(\mathbf{b}_1)}{\nu(\mathbf{b}_2)} \leq e^{\epsilon_p}$. The theorem follows. □

The main ingredient of the proof of Theorem 1 is a result about the sensitivity of regularized ERM, which is provided below.

Lemma 1. *Let $G(\mathbf{f})$ and $g(\mathbf{f})$ be two vector-valued functions, which are continuous, and differentiable at all points. Moreover, let $G(\mathbf{f})$ and $G(\mathbf{f}) + g(\mathbf{f})$ be λ-strongly convex. If $\mathbf{f}_1 = \operatorname{argmin}_{\mathbf{f}} G(\mathbf{f})$ and $\mathbf{f}_2 = \operatorname{argmin}_{\mathbf{f}} G(\mathbf{f}) + g(\mathbf{f})$, then*

$$\|\mathbf{f}_1 - \mathbf{f}_2\| \leq \frac{1}{\lambda} \max_{\mathbf{f}} \|\nabla g(\mathbf{f})\| . \tag{12}$$

Proof. Using the definition of $\mathbf{f}_1$ and $\mathbf{f}_2$, and the fact that G and g are continuous and differentiable everywhere,

$$\nabla G(\mathbf{f}_1) = \nabla G(\mathbf{f}_2) + \nabla g(\mathbf{f}_2) = \mathbf{0}. \tag{13}$$

As $G(\mathbf{f})$ is λ-strongly convex, it follows from Lemma 14 of (Shalev-Shwartz [2007]) that:

$$(\nabla G(\mathbf{f}_1) - \nabla G(\mathbf{f}_2))^T(\mathbf{f}_1 - \mathbf{f}_2) \geq \lambda \|\mathbf{f}_1 - \mathbf{f}_2\|^2 . \tag{14}$$

Combining this with (13) and the Cauchy-Schwartz inequality, we get that

$$\|\mathbf{f}_1 - \mathbf{f}_2\| \cdot \|\nabla g(\mathbf{f}_2)\| \geq (\mathbf{f}_1 - \mathbf{f}_2)^T \nabla g(\mathbf{f}_2) = (\nabla G(\mathbf{f}_1) - \nabla G(\mathbf{f}_2))^T(\mathbf{f}_1 - \mathbf{f}_2) \geq \lambda \|\mathbf{f}_1 - \mathbf{f}_2\|^2 . \tag{15}$$

The conclusion follows from dividing both sides by $\lambda \|\mathbf{f}_1 - \mathbf{f}_2\|$. □

Corollary 1. *If $N(\cdot)$ is differentiable and 1-strongly convex, and ℓ is convex and differentiable with $|\ell'(z)| \leq 1$ for all z, then, the L_2-sensitivity of $J(\mathbf{f}, \mathcal{D})$ is at most $\frac{2}{n\Lambda}$.*

Proof. Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and $\mathcal{D}' = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}'_n, y'_n)\}$ be two datasets that differ in the value of the n-th individual. Moreover, we let $G(\mathbf{f}) = J(\mathbf{f}, \mathcal{D})$, $g(\mathbf{f}) = J(\mathbf{f}, \mathcal{D}') - J(\mathbf{f}, \mathcal{D})$, $\mathbf{f}_1 = \mathrm{argmin}_{\mathbf{f}}\, J(\mathbf{f}, \mathcal{D})$, and $\mathbf{f}_2 = \mathrm{argmin}_{\mathbf{f}}\, J(\mathbf{f}, \mathcal{D}')$. Finally, we set $g(\mathbf{f}) = \frac{1}{n}(\ell(y'_n \mathbf{f}^T \mathbf{x}'_n) - \ell(y_n \mathbf{f}^T \mathbf{x}_n))$.

We observe that due to the convexity of ℓ, and 1-strong convexity of $N(\cdot)$, $G(\mathbf{f}) = J(\mathbf{f}, \mathcal{D})$ is Λ-strongly convex. Moreover, $G(\mathbf{f}) + g(\mathbf{f}) = J(\mathbf{f}, \mathcal{D}')$ is also Λ-strongly convex. Finally, due to the differentiability of $N(\cdot)$ and ℓ, $G(\mathbf{f})$ and $g(\mathbf{f})$ are also differentiable at all points. We have:

$$\nabla g(\mathbf{f}) = \frac{1}{n}(y_n \ell'(y_n \mathbf{f}^T \mathbf{x}_n)\mathbf{x}_n - y'_n \ell'(y'_n \mathbf{f}^T \mathbf{x}'_n)\mathbf{x}'_n). \tag{16}$$

As $y_i \in [-1, 1]$, $|\ell'(z)| \leq 1$, for all z, and $||\mathbf{x}_i|| \leq 1$, for any $\mathbf{f}$, $||\nabla g(\mathbf{f})|| \leq \frac{1}{n}(||\mathbf{x}_n - \mathbf{x}'_n||) \leq \frac{1}{n}(||\mathbf{x}_n|| + ||\mathbf{x}'_n||) \leq \frac{2}{n}$. The proof now follows by an application of Lemma 1. □

3.3.2 Privacy Guarantees for Objective Perturbation

In this section, we show that Algorithm 2 is ϵ_p-differentially private.

Theorem 2. *If $N(\cdot)$ is 1-strongly convex and doubly differentiable, and $\ell(\cdot)$ is convex and doubly differentiable, with $|\ell'(z)| \leq 1$ and $|\ell''(z)| \leq c$ for all z, then Algorithm 2 is ϵ_p-differentially private.*

Proof. Consider an $\mathbf{f}_{\mathrm{priv}}$ output by Algorithm 2. We observe that given *any* fixed $\mathbf{f}_{\mathrm{priv}}$ and a fixed dataset $\mathcal{D}$, there always exists a $\mathbf{b}$ such that Algorithm 2 outputs $\mathbf{f}_{\mathrm{priv}}$ on input $\mathcal{D}$. Because ℓ is differentiable and convex, and $N(\cdot)$ is differentiable, we can take the gradient of the objective function and set it to $\mathbf{0}$ at $\mathbf{f}_{\mathrm{priv}}$. Therefore,

$$\mathbf{b} = -n\Lambda \nabla N(\mathbf{f}_{\mathrm{priv}}) - \sum_{i=1}^{n} y_i \ell'(y_i \mathbf{f}_{\mathrm{priv}}^T \mathbf{x}_i)\mathbf{x}_i - n\Delta \mathbf{f}_{\mathrm{priv}}. \tag{17}$$

Note that (17) holds because for any $\mathbf{f}$, $\nabla \ell(\mathbf{f}^T \mathbf{x}) = \ell'(\mathbf{f}^T \mathbf{x})\mathbf{x}$.

We claim that as ℓ is differentiable and $J(\mathbf{f}, \mathcal{D}) + \frac{\Delta}{2}||\mathbf{f}||^2$ is strongly convex, given a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, there is a bijection between $\mathbf{b}$ and $\mathbf{f}_{\mathrm{priv}}$. The relation (17) shows that two different $\mathbf{b}$ values cannot result in the same $\mathbf{f}_{\mathrm{priv}}$. Furthermore, since the objective is strictly convex, for a fixed $\mathbf{b}$ and $\mathcal{D}$, there is a unique $\mathbf{f}_{\mathrm{priv}}$; therefore the map from $\mathbf{b}$ to $\mathbf{f}_{\mathrm{priv}}$ is injective. The relation (17) also shows that for any $\mathbf{f}_{\mathrm{priv}}$ there exists a $\mathbf{b}$ for which $\mathbf{f}_{\mathrm{priv}}$ is the minimizer, so the map from $\mathbf{b}$ to $\mathbf{f}_{\mathrm{priv}}$ is surjective.

To show ϵ_p-differential privacy, we need to compute the ratio $g(\mathbf{f}_{\mathrm{priv}}|\mathcal{D})/g(\mathbf{f}_{\mathrm{priv}}|\mathcal{D}')$ of the densities of $\mathbf{f}_{\mathrm{priv}}$ under the two datasets $\mathcal{D}$ and $\mathcal{D}'$. This ratio can be written as [Billingsley, 1995]

$$\frac{g(\mathbf{f}_{\mathrm{priv}}|\mathcal{D})}{g(\mathbf{f}_{\mathrm{priv}}|\mathcal{D}')} = \frac{\mu(\mathbf{b}|\mathcal{D})}{\mu(\mathbf{b}'|\mathcal{D}')} \cdot \frac{|\det(\mathbf{J}(\mathbf{f}_{\mathrm{priv}} \to \mathbf{b}|\mathcal{D}))|^{-1}}{|\det(\mathbf{J}(\mathbf{f}_{\mathrm{priv}} \to \mathbf{b}'|\mathcal{D}'))|^{-1}}, \tag{18}$$

where $\mathbf{J}(\mathbf{f}_{\mathrm{priv}} \to \mathbf{b}|\mathcal{D})$, $\mathbf{J}(\mathbf{f}_{\mathrm{priv}} \to \mathbf{b}|\mathcal{D}')$ are the Jacobian matrices of the mappings from $\mathbf{f}_{\mathrm{priv}}$ to $\mathbf{b}$, and $\mu(\mathbf{b}|\mathcal{D})$ and $\mu(\mathbf{b}|\mathcal{D}')$ are the densities of the unique $\mathbf{b}$s given output $\mathbf{f}_{\mathrm{priv}}$, when the datasets are $\mathcal{D}$ and $\mathcal{D}'$ respectively.

First, we bound the ratio of the Jacobian determinants. Let $\mathbf{b}^{(j)}$ denote the j-th coordinate of $\mathbf{b}$. From (17) we have

$$\mathbf{b}^{(j)} = -n\Lambda \nabla N(\mathbf{f}_{\text{priv}})^{(j)} - \sum_{i=1}^{n} \ell'(y_i \mathbf{f}_{\text{priv}}^T \mathbf{x}_i)\mathbf{x}_i^{(j)} - n\Delta \mathbf{f}_{\text{priv}}^{(j)}. \tag{19}$$

Given a dataset $\mathcal{D}$, the (j,k)-th entry of the Jacobian matrix $\mathbf{J}(\mathbf{f} \to \mathbf{b}|\mathcal{D})$ is

$$\frac{\partial \mathbf{b}^{(j)}}{\partial \mathbf{f}_{\text{priv}}^{(k)}} = -n\Lambda \nabla^2 N(\mathbf{f}_{\text{priv}})^{(j,k)} - \sum_i y_i^2 \ell''(y_i \mathbf{f}_{\text{priv}}^T \mathbf{x}_i)\mathbf{x}_i^{(j)}\mathbf{x}_i^{(k)} - n\Delta 1(j=k), \tag{20}$$

where $1(\cdot)$ is the indicator function. We note that the Jacobian is defined for all $\mathbf{f}_{\text{priv}}$ because $N(\cdot)$ and ℓ are globally doubly differentiable.

Let $\mathcal{D}$ and $\mathcal{D}'$ be two datasets which differ in the value of the n-th item such that $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1}), (\mathbf{x}_n, y_n)\}$ and $\mathcal{D}' = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1}), (\mathbf{x}'_n, y'_n)\}$. Moreover, we define matrices A and E as follows:

$$\begin{aligned} A &= n\Lambda \nabla^2 N(\mathbf{f}_{\text{priv}}) + \sum_{i=1}^{n} y_i^2 \ell''(y_i \mathbf{f}_{\text{priv}}^T \mathbf{x}_i)\mathbf{x}_i \mathbf{x}_i^T + n\Delta I_d \\ E &= -y_n^2 \ell''(y_n \mathbf{f}_{\text{priv}}^T \mathbf{x}_n)\mathbf{x}_n \mathbf{x}_n^T + (y'_n)^2 \ell''(y'_n \mathbf{f}_{\text{priv}}^T \mathbf{x}'_n)\mathbf{x}'_n \mathbf{x}'^T_n. \end{aligned}$$

Then, $\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}|\mathcal{D}) = -A$, and $\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}|\mathcal{D}') = -(A+E)$.

As E has rank at most 2, from Lemma 2,

$$\begin{aligned} \frac{|\det(\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}|\mathcal{D}'))|}{|\det(\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}|\mathcal{D}))|} &= \frac{|\det(A+E)|}{|\det A|} \\ &= |1 + \lambda_1(A^{-1}E) + \lambda_2(A^{-1}E) + \lambda_1(A^{-1}E)\lambda_2(A^{-1}E)|. \end{aligned} \tag{21}$$

For a 1-strongly convex function N, the Hessian $\nabla^2 N(\mathbf{f}_{\text{priv}})$ has eigenvalues greater than 1 (Boyd and Vandenberghe [2004]). Since we have assumed ℓ is doubly differentiable and convex, any eigenvalue of A is therefore at least $n\Lambda + n\Delta$; therefore, for $j = 1, 2$, $|\lambda_j(A^{-1}E)| \leq \frac{|\lambda_j(E)|}{n(\Lambda+\Delta)}$. Applying Lemma 3, with $w_1 = y_n^2 \ell''(y_n \mathbf{f}_{\text{priv}}^T \mathbf{x}_n)$, $\mathbf{x}_1 = \mathbf{x}_n$, and $w_2 = -(y'_n)^2 \ell''(y'_n \mathbf{f}_{\text{priv}}^T \mathbf{x}'_n)$, $\mathbf{x}_2 = \mathbf{x}'_n$, and upper bounds on $|y_i|$ and $||\mathbf{x}_i||$ and $|\ell''(z)|$,

$$|\lambda_1(E)| + |\lambda_2(E)| \leq 2c. \tag{22}$$

Therefore, $|\lambda_1(E)| \cdot |\lambda_2(E)| \leq c^2$, and

$$\frac{|\det(A+E)|}{|\det(A)|} \leq 1 + \frac{2c}{n(\Lambda+\Delta)} + \frac{c^2}{n^2(\Lambda+\Delta)^2} = \left(1 + \frac{c}{n(\Lambda+\Delta)}\right)^2. \tag{23}$$

We now consider two cases. In the first case, $\Delta = 0$, and by definition, in that case, $1 + \frac{2c}{n\Lambda} + \frac{c^2}{n^2\Lambda^2} \leq e^{\epsilon_p - \epsilon'_p}$. In the second case, $\Delta > 0$, and in this case, by definition of Δ, $(1 + \frac{c}{n(\Lambda+\Delta)})^2 = e^{\epsilon_p/2} = e^{\epsilon_p - \epsilon'_p}$.

Next, we bound the ratio of the densities of $\mathbf{b}$. We observe that as $|\ell'(z)| \leq 1$, for any z and $|y_i|, ||\mathbf{x}_i|| \leq 1$, for datasets $\mathcal{D}$ and $\mathcal{D}'$ which differ by one value,

$$\mathbf{b}' - \mathbf{b} = y_n \ell'(y_n \mathbf{f}_{\text{priv}}^T \mathbf{x}_n)\mathbf{x}_n - y'_n \ell'(y_n \mathbf{f}_{\text{priv}}^T \mathbf{x}'_n)\mathbf{x}'_n. \tag{24}$$

This implies that:

$$\|\mathbf{b}\| - \left\|\mathbf{b}'\right\| \leq \left\|\mathbf{b} - \mathbf{b}'\right\| \leq 2. \tag{25}$$

We can write:

$$\frac{\mu(\mathbf{b}|\mathcal{D})}{\mu(\mathbf{b}'|\mathcal{D}')} = \frac{||\mathbf{b}||^{d-1} e^{-\epsilon_p' ||\mathbf{b}||/2} \cdot \frac{1}{\text{surf}(||\mathbf{b}||)}}{||\mathbf{b}'||^{d-1} e^{-\epsilon_p' ||\mathbf{b}'||/2} \cdot \frac{1}{\text{surf}(||\mathbf{b}'||)}} \leq e^{\epsilon_p' (||\mathbf{b}|| - ||\mathbf{b}'||)/2} \leq e^{\epsilon_p'}, \tag{26}$$

where surf(x) denotes the surface area of the sphere in d dimensions with radius x. Here the last step follows from the fact that surf$(x) = s(1)x^{d-1}$, where $s(1)$ is the surface area of the unit sphere in $\mathbb{R}^d$.

Finally, we are ready to bound the ratio of densities:

$$\begin{aligned}
\frac{g(\mathbf{f}_{\text{priv}}|\mathcal{D})}{g(\mathbf{f}_{\text{priv}}|\mathcal{D}')} &= \frac{\mu(\mathbf{b}|\mathcal{D})}{\mu(\mathbf{b}'|\mathcal{D}')} \cdot \frac{|\det(\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}|\mathcal{D}'))|}{|\det(\mathbf{J}(\mathbf{f}_{\text{priv}} \to \mathbf{b}'|\mathcal{D}))|} && (27) \\
&= \frac{\mu(\mathbf{b}|\mathcal{D})}{\mu(\mathbf{b}'|\mathcal{D}')} \cdot \frac{|\det(A+E)|}{|\det A|} && (28) \\
&\leq e^{\epsilon_p'} \cdot e^{\epsilon_p - \epsilon_p'} && (29) \\
&\leq e^{\epsilon_p}. && (30)
\end{aligned}$$

Thus, Algorithm 2 satisfies Definition 2. □

Lemma 2. *If A is full rank, and if E has rank at most 2, then,*

$$\frac{\det(A+E) - \det(A)}{\det(A)} = \lambda_1(A^{-1}E) + \lambda_2(A^{-1}E) + \lambda_1(A^{-1}E)\lambda_2(A^{-1}E), \tag{31}$$

where $\lambda_j(Z)$ is the j-th eigenvalue of matrix Z.

Proof. To show this we use some recent results of Ipsen and Rehman [2008]. Theorem 2.13 in their paper gives an expansion

$$\frac{\det(A+E) - \det(A)}{\det A} = \det(A^{-1}E) + \sum_{k=1}^{n-1} \sum_{1 \leq i_i < i_2 < \cdots < i_k \leq n} \det\left((A^{-1}E)_{i_1,\ldots,i_k}\right), \tag{32}$$

where $(B)_{i_1,\ldots,i_k}$ is the submatrix of matrix B formed by removing rows $i_1, \ldots, i_k$ and columns $i_1, \ldots, i_k$ from B.

Turning to the other terms, note that the computation should not depend on the basis in which the matrices are expressed. In particular, we can write out the expansion as if $A^{-1}E$ has been diagonalized. Since the matrices are positive semidefinite, diagonalizing $A^{-1}E$ puts the eigenvalues on the diagonal. Therefore $\det((A^{-1}E)_{i_1,\ldots,i_k})$ is the product of the eigenvalues of $A^{-1}E$ for indices not equal to $i_1, \ldots, i_k$. Thus

$$\begin{aligned}
\sum_{1 \leq i_i < i_2 < \cdots < i_k \leq n} \det\left((A^{-1}E)_{i_1,\ldots,i_k}\right) &= \sum_{1 \leq i_i < i_2 < \cdots < i_k \leq n} \prod_{j=1}^{k} \lambda_{i_j}(A^{-1}E) && (33) \\
&= \sum_{S \subset \{1,2,\ldots,n\} : |S| = k} \prod_{j \in S} \lambda_j(A^{-1}E). && (34)
\end{aligned}$$

This is just the k-th symmetric polynomial in the eigenvalues of $A^{-1}E$, which is the sum of all products of size-k subsets of the eigenvalues. But since E has rank at most 2, the matrix $A^{-1}E$ has rank at most 2, so at most 2 of the eigenvalues are non-zero. Thus for $k \geq 2$ all of those terms are 0 since they are products of 3 or more eigenvalues. Thus we have

$$\frac{\det(A+E) - \det(A)}{\det A} = \lambda_1(A^{-1}E) + \lambda_2(A^{-1}E) + \lambda_1(A^{-1}E)\lambda_2(A^{-1}E). \tag{35}$$

□

Lemma 3. *Let $A = \sum_j w_j \mathbf{x}_j \mathbf{x}_j^T$, for $d \times 1$ vectors $\mathbf{x}_j$. Then,*

$$\sum_{i=1}^{d} |\lambda_i(A)| \leq \sum_j |w_j| \cdot ||\mathbf{x}_j||^2. \tag{36}$$

Proof. Let $\mathbf{u}_1, \ldots, \mathbf{u}_d$ be the singular vectors of A. Recall that the $\mathbf{u}_i$s form an orthonormal basis of $\mathbb{R}^d$. Now,

$$\begin{aligned}\sum_{i=1}^{d} |\lambda_i(A)| &= \sum_{i=1}^{d} |\mathbf{u}_i^T A \mathbf{u}_i| = \sum_{i=1}^{d} |\mathbf{u}_i^T (\sum_j w_j \mathbf{x}_j \mathbf{x}_j^T) \mathbf{u}_i| = \sum_{i=1}^{d} |\sum_j w_j (\mathbf{u}_i^T \mathbf{x}_j)^2| \\ &\leq \sum_{i=1}^{d} \sum_j |w_j| (\mathbf{u}_i^T \mathbf{x}_j)^2 = \sum_j |w_j| \sum_{i=1}^{d} (\mathbf{u}_i^T \mathbf{x}_j)^2 = \sum_j |w_j| \cdot ||\mathbf{x}_j||^2,\end{aligned}$$

where the last step follows because $\mathbf{u}_i$s form an orthonormal basis of $\mathbb{R}^d$. □

3.4 Application to classification

In this section, we show how to use our results to provide privacy-preserving versions of logistic regression and support vector machines.

3.4.1 Logistic Regression

One popular ERM classification algorithm is regularized logistic regression. In this case, $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, and the loss function is $\ell_{\mathrm{LR}}(z) = \log(1 + e^{-z})$. Taking derivatives and double derivatives,

$$\begin{aligned}\ell'_{\mathrm{LR}}(z) &= \frac{-1}{(1+e^z)} \\ \ell''_{\mathrm{LR}}(z) &= \frac{1}{(1+e^{-z})(1+e^z)}.\end{aligned}$$

Note that ℓ_{LR} is continuous, differentiable and doubly differentiable, with $c \leq \frac{1}{4}$. Therefore, we can plug in logistic loss directly to Theorems 1 and 2 to get the following result.

Corollary 2. *The output of Algorithm 1 with $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, $\ell = \ell_{\mathrm{LR}}$ is an ϵ_p-differentially private approximation to logistic regression. The output of Algorithm 2 with $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, $c = \frac{1}{4}$, and $\ell = \ell_{\mathrm{LR}}$, is an ϵ_p-differentially private approximation to logistic regression.*

We quantify how well the outputs of Algorithms 1 and 2 approximate (non-private) logistic regression in Section 4.

3.4.2 Support Vector Machines

Another very commonly used classifier is L_2-regularized support vector machines. In this case, again, $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, and

$$\ell_{\mathrm{SVM}}(z) = \max(0, 1 - z). \tag{37}$$

Notice that this loss function is continuous, but not differentiable, and thus it does not satisfy conditions in Theorems 1 and 2.

There are two alternative solutions to this. First, we can approximate ℓ_{SVM} by a different loss function, which is doubly differentiable, as follows. This is motivated by Chapelle [2007].

$$\ell_s(z) = \begin{cases} 0 & \text{if} \quad z > 1 + h \\ -\frac{(1-z)^4}{16h^3} + \frac{3(1-z)^2}{8h} + \frac{1-z}{2} + \frac{3h}{16} & \text{if} \quad |1 - z| \leq h \\ 1 - z & \text{if} \quad z < 1 - h \end{cases} \tag{38}$$

As $h \rightarrow 0$, this loss approaches the hinge loss. Taking derivatives, we observe that:

$$\ell_s'(z) = \begin{cases} 0 & \text{if} \quad z > 1 + h \\ \frac{(1-z)^3}{4h^3} - \frac{3(1-z)}{4h} - \frac{1}{2} & \text{if} \quad |1 - z| \leq h \\ -1 & \text{if} \quad z < 1 - h \end{cases} \tag{39}$$

Moreover,

$$\ell_s''(z) = \begin{cases} 0 & \text{if} \quad z > 1 + h \\ -\frac{3(1-z)^2}{4h^3} + \frac{3}{4h} & \text{if} \quad |1 - z| \leq h \\ 0 & \text{if} \quad z < 1 - h \end{cases} \tag{40}$$

Observe that this implies that $|\ell_s''(z)| \leq \frac{3}{4h}$ for all h and z. Moreover, ℓ_s is convex, as $\ell_s''(z) \geq 0$ for all z. Therefore, ℓ_s can be used in Theorems 1 and 2, which gives us privacy-preserving approximations to regularized support vector machines.

Corollary 3. *The output of Algorithm 1 with $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, and $\ell = \ell_s$ is an ϵ_p-differentially private approximation to support vector machines. The output of Algorithm 2 with $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, $c = \frac{3}{4h}$, and $\ell = \ell_s$ is an ϵ_p-differentially private approximation to support vector machines.*

The second solution is to use Huber Loss, as suggested by Chapelle [2007], which is defined as follows:

$$\ell_{\mathrm{Huber}}(z) = \begin{cases} 0 & \text{if} \quad z > 1 + h \\ \frac{1}{4h}(1 + h - z)^2 & \text{if} \quad |1 - z| \leq h \\ 1 - z & \text{if} \quad z < 1 - h \end{cases} \tag{41}$$

Observe that Huber loss is convex and differentiable, and piecewise doubly-differentiable, with $c = \frac{1}{2h}$. However, it is not globally doubly differentiable, and hence the Jacobian in the proof of Theorem 2 is undefined for certain values of $\mathbf{f}$. However, we can show that in this case, Algorithm 2, when run with $c = \frac{1}{2h}$ satisfies Definition 3.

Let G denote the map from $\mathbf{f}_{\mathrm{priv}}$ to $\mathbf{b}$ in (17) under $\mathcal{B} = \mathcal{D}$, and H denote the map under $\mathcal{B} = \mathcal{D}'$. By definition, the probability $\mathbb{P}(\mathbf{f}_{\mathrm{priv}} \in \mathcal{S} \mid \mathcal{B} = \mathcal{D}) = \mathbb{P}_{\mathbf{b}}(\mathbf{b} \in G(\mathcal{S}))$.

Corollary 4. *Let $\mathbf{f}_{\text{priv}}$ be the output of Algorithm 2 with $\ell = \ell_{\text{Huber}}$, $c = \frac{1}{2h}$, and $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||_2^2$. For any set $\mathcal{S}$ of possible values of $\mathbf{f}_{\text{priv}}$, and any pair of datasets $\mathcal{D}$, $\mathcal{D}'$ which differ in the private value of one person $(\mathbf{x}_n, y_n)$,*

$$e^{-\epsilon_p}\mathbb{P}(\mathcal{S} \mid \mathcal{B} = \mathcal{D}') \leq \mathbb{P}(\mathcal{S} \mid \mathcal{B} = \mathcal{D}) \leq e^{\epsilon_p}\mathbb{P}(\mathcal{S} \mid \mathcal{B} = \mathcal{D}'). \tag{42}$$

Proof. Consider the event $\mathbf{f}_{\text{priv}} \in \mathcal{S}$. Let $\mathcal{T} = G(\mathcal{S})$ and $\mathcal{T}' = H(\mathcal{S})$. Because G is a bijection, we have

$$\mathbb{P}(\mathbf{f}_{\text{priv}} \in \mathcal{S} \mid \mathcal{B} = \mathcal{D}) = \mathbb{P}_{\mathbf{b}}(\mathbf{b} \in \mathcal{T} \mid \mathcal{B} = \mathcal{D}), \tag{43}$$

and a similar expression when $\mathcal{B} = \mathcal{D}'$.

Now note that $\ell''_{\text{Huber}}(z)$ is only non-differentiable for a finite number of values of z. Let $\mathcal{Z}$ be the set of these values of z.

$$\mathcal{C} = \{\mathbf{f} : y\mathbf{f}^T\mathbf{x} = z \in \mathcal{Z},\ (\mathbf{x}, y) \in \mathcal{D} \cup \mathcal{D}'\}. \tag{44}$$

Pick a tuple $(z, (\mathbf{x}, y)) \in \mathcal{Z} \times (\mathcal{D} \cup \mathcal{D}')$. The set of $\mathbf{f}$ such that $y\mathbf{f}^T\mathbf{x} = z$ is a hyperplane in $\mathbb{R}^d$. Since $\nabla N(\mathbf{f}) = \mathbf{f}/2$ and ℓ' is piecewise linear, from (17) we see that the set of corresponding $\mathbf{b}$'s is also piecewise linear, and hence has Lebesgue measure 0. Since the measure corresponding to $\mathbf{b}$ is absolutely continuous with respect to the Lebesgue measure, this hyperplane has probability 0 under $\mathbf{b}$ as well. Since $\mathcal{C}$ is a finite union of such hyperplanes, we have $\mathbb{P}(\mathbf{b} \in G(\mathcal{C})) = 0$.

Thus we have $\mathbb{P}_{\mathbf{b}}(\mathcal{T} \mid \mathcal{B} = \mathcal{D}) = \mathbb{P}_{\mathbf{b}}(G(\mathcal{S} \setminus \mathcal{C}) \mid \mathcal{B} = \mathcal{D})$, and similarly for $\mathcal{D}'$. From the definition of G and H, for $\mathbf{f} \in \mathcal{S} \setminus \mathcal{C}$,

$$H(\mathbf{f}) = G(\mathbf{f}) + y_n\ell'(y_n\mathbf{f}^T\mathbf{x}_n)\mathbf{x}_n - y'_n\ell'(y'_n\mathbf{f}^T\mathbf{x}'_n)\mathbf{x}'_n. \tag{45}$$

since $\mathbf{f} \notin \mathcal{C}$, this mapping shows that if $\mathbb{P}_{\mathbf{b}}(G(\mathcal{S} \setminus \mathcal{C}) \mid \mathcal{B} = \mathcal{D}) = 0$ then we must have $\mathbb{P}_{\mathbf{b}}(H(\mathcal{S} \setminus \mathcal{C}) \mid \mathcal{B} = \mathcal{D}) = 0$. Thus the result holds for sets of measure 0. If $\mathcal{S} \setminus \mathcal{C}$ has positive measure we can calculate the ratio of the probabilities for $\mathbf{f}_{\text{priv}}$ for which the loss is twice-differentiable. For such $\mathbf{f}_{\text{priv}}$ the Jacobian is also defined, and we can use a method similar to Theorem 2 to prove the result. □

Remark: Because the privacy proof for Algorithm 1 does not require the analytic properties of 2, we can also use Huber loss in Algorithm 1 to get an ϵ_g-differentially private approximation to the SVM. We quantify how well the outputs of Algorithms 1 and 2 approximate private support vector machines in Section 4.

4 Generalization performance

In this section, we provide guarantees on the performance of privacy-preserving ERM algorithms in Section 3. We provide these bounds for L_2-regularization. To quantify this performance, we will assume that the n entries in the dataset $\mathcal{D}$ are drawn i.i.d. according to a fixed distribution $P(\mathbf{x}, y)$. We measure the performance of these algorithms by the number of samples n required to acheive error $L^* + \epsilon_g$, where L^* is the loss of the best ERM predictor.

Given a distribution P the expected loss $L(\mathbf{f})$ for a classifier $\mathbf{f}$ is

$$L(\mathbf{f}) = \mathbb{E}_{(\mathbf{x},y)\sim P}\left[\ell(\mathbf{f}^T\mathbf{x}, y)\right]. \tag{46}$$

The sample complexity for generalization error ϵ_g against a classifier $\mathbf{f}_0$ is number of samples n required to achieve error $L(\mathbf{f}_0) + \epsilon_g$ under any data distribution P. We would like the sample complexity to be low.

For a fixed P we define the following function, which will be useful in our analysis:

$$\bar{J}(\mathbf{f}) = L(\mathbf{f}) + \frac{\Lambda}{2}\|\mathbf{f}\|^2. \tag{47}$$

The function $\bar{J}(\mathbf{f})$ is the expectation (over P) of the non-private L_2-regularized ERM objective evaluated at $\mathbf{f}$.

For non-private ERM, Shalev-Shwartz and Srebro [2008] show that for a given $\mathbf{f}_0$ with loss $L(\mathbf{f}_0) = L^*$, if the number of data points

$$n > C\frac{||\mathbf{f}_0||^2\log(\frac{1}{\delta})}{\epsilon_g^2} \tag{48}$$

for some constant C, then the excess loss of the SVM solution $\mathbf{f}_{svm}$ satisfies $L(\mathbf{f}_{svm}) \leq L(\mathbf{f}_0) + \epsilon_g$. This order growth will hold for our results as well. It also serves as a reference against which we can compare the additional burden on the sample complexity imposed by the privacy constraints.

For most learning problems, we require the generalization error $\epsilon_g < 1$. Moreover, it is also typically the case that for more difficult learning problems, $||\mathbf{f}_0||$ is higher. For example, for regularized SVM, $\frac{1}{||\mathbf{f}_0||}$ is the margin of classification, and as a result, $||\mathbf{f}_0||$ is higher for learning problems with smaller margin. From the bounds provided in this section, we note that the dominating term in the sample requirement for objective perturbation has a better dependence on $||\mathbf{f}_0||$ as well as $\frac{1}{\epsilon_g}$; as a result, for more difficult learning problems, we expect objective perturbation to perform better than output perturbation.

4.1 Output perturbation

First, we provide performance guarantees for Algorithm 1, by providing a bound on the number of samples required for Algorithm 1 to produce a classifier with low error.

Definition 6. *A function $g(z) : \mathbb{R} \to \mathbb{R}$ is c-Lipschitz if $|g(z_1) - g(z_2)| \leq c|z_1 - z_2|$, for all pairs z_1, z_2.*

Recall that if a function $g(z)$ is differentiable, with $|g'(z)| \leq r$ for all z, then $g(z)$ is also r-Lipschitz.

Theorem 3. *Let $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, and let $\mathbf{f}_0$ be a classifier such that $L(\mathbf{f}_0) = L^*$. If ℓ is differentiable and continuous, the derivative ℓ' is c-Lipschitz, the data $\mathcal{D}$ is drawn i.i.d. according to P, and the number of training samples satisfies*

$$n > C\max\left(\frac{||\mathbf{f}_0||^2\log(\frac{1}{\delta})}{\epsilon_g^2}, \frac{d\log(\frac{d}{\delta})||\mathbf{f}_0||}{\epsilon_g\epsilon_p}, \frac{d\log(\frac{d}{\delta})c^{1/2}||\mathbf{f}_0||^2}{\epsilon_g^{3/2}\epsilon_p}\right), \tag{49}$$

for some constant C, then the output $\mathbf{f}_{\mathrm{priv}}$ of Algorithm 1 satisfies

$$\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) \leq L^* + \epsilon_g\right) \geq 1 - 2\delta. \tag{50}$$

Proof. Let

$$\mathbf{f}_{\mathrm{rtr}} = \operatorname*{argmin}_{\mathbf{f}} \bar{J}(\mathbf{f})$$
$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}} J(\mathbf{f}, \mathcal{D}),$$

and $\mathbf{f}_{\mathrm{priv}}$ denote the output of Algorithm 1. Using the analysis method of Shalev-Shwartz and Srebro [2008] shows

$$L(\mathbf{f}_{\mathrm{priv}}) = L(\mathbf{f}_0) + (\bar{J}(\mathbf{f}_{\mathrm{priv}}) - \bar{J}(\mathbf{f}_{\mathrm{rtr}})) + (\bar{J}(\mathbf{f}_{\mathrm{rtr}}) - \bar{J}(\mathbf{f}_0)) + \frac{\Lambda}{2}||\mathbf{f}_0||^2 - \frac{\Lambda}{2}||\mathbf{f}_{\mathrm{priv}}||^2. \tag{51}$$

We will bound the terms on the right-hand side of (51).

For a regularizer $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$ the Hessian satisfies $||\nabla^2 N(\mathbf{f})||_2 \leq 1$. Therefore, from Lemma 4, with probability $1 - \delta$ over the privacy mechanism,

$$J(\mathbf{f}_{\mathrm{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D}) \leq \frac{8d^2 \log^2(d/\delta)(c + \Lambda)}{\Lambda^2 n^2 \epsilon_p^2}. \tag{52}$$

Furthermore, the results of Sridharan et al. [2008] show that with probability $1 - \delta$ over the choice of the data distribution,

$$\bar{J}(\mathbf{f}_{\mathrm{priv}}) - \bar{J}(\mathbf{f}_{\mathrm{rtr}}) \leq 2(J(\mathbf{f}_{\mathrm{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D})) + O\left(\frac{\log(1/\delta)}{\Lambda n}\right). \tag{53}$$

Combining the preceding two statements, with probability $1 - 2\delta$ over the noise in the privacy mechanism and the data distribution, the second term in the right-hand-side of (51) is at most:

$$\bar{J}(\mathbf{f}_{\mathrm{priv}}) - \bar{J}(\mathbf{f}_{\mathrm{rtr}}) \leq \frac{16d^2 \log^2(d/\delta)(c + \Lambda)}{\Lambda^2 n^2 \epsilon_p^2} + O\left(\frac{\log(1/\delta)}{\Lambda n}\right). \tag{54}$$

By definition of $\mathbf{f}_{\mathrm{rtr}}$, the difference $(\bar{J}(\mathbf{f}_{\mathrm{rtr}}) - \bar{J}(\mathbf{f}_0)) \leq 0$. Setting $\Lambda = \frac{\epsilon_g}{||\mathbf{f}_0||^2}$ in (51) and using (54), we obtain

$$L(\mathbf{f}_{\mathrm{priv}}) \leq L(\mathbf{f}_0) + \frac{16||\mathbf{f}_0||^4 d^2 \log^2(d/\delta)(c + \epsilon_g/||\mathbf{f}_0||^2)}{n^2 \epsilon_g^2 \epsilon_p^2} + O\left(||\mathbf{f}_0||^2 \frac{\log(1/\delta)}{n\epsilon_g}\right) + \frac{\epsilon_g}{2}. \tag{55}$$

Solving for n to make the total excess error equal to ϵ_g yields (49). □

Lemma 4. *Suppose $N(\cdot)$ is doubly differentiable with $||\nabla^2 N(\mathbf{f})||_2 \leq \eta$ for all $\mathbf{f}$, and suppose that ℓ is differentiable and has continuous and c-Lipschitz derivatives. Given training data $\mathcal{D}$, let $\mathbf{f}^*$ be a classifier that minimizes $J(\mathbf{f}, \mathcal{D})$ and let $\mathbf{f}_{\mathrm{priv}}$ be the classifier output by Algorithm 1. Then*

$$\mathbb{P}_{\mathbf{b}}\left(J(\mathbf{f}_{\mathrm{priv}}, \mathcal{D}) \leq J(\mathbf{f}^*, \mathcal{D}) + \frac{2d^2(c + \Lambda\eta)\log^2(d/\delta)}{\Lambda^2 n^2 \epsilon_p^2}\right) \geq 1 - \delta, \tag{56}$$

where the probability is taken over the randomness in the noise $\mathbf{b}$ *of Algorithm 1.*

Note that when ℓ is doubly differentiable, c is an upper bound on the double derivative of ℓ, and is the same as the constant c in Theorem 2.

Proof. Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, and recall that $||\mathbf{x}_i|| \leq 1$, and $|y_i| \leq 1$. As $N(\cdot)$ and ℓ are differentiable, we use the Mean Value Theorem to show that for some t between 0 and 1,

$$\begin{aligned} J(\mathbf{f}_{\text{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D}) &= (\mathbf{f}_{\text{priv}} - \mathbf{f}^*)^T \nabla J(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}}) &(57)\\ &\leq ||\mathbf{f}_{\text{priv}} - \mathbf{f}^*|| \cdot ||\nabla J(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})||, &(58)\end{aligned}$$

where the second step follows by an application of the Cauchy-Schwartz inequality. Recall that

$$\nabla J(\mathbf{f}, \mathcal{D}) = \Lambda \nabla N(\mathbf{f}) + \frac{1}{n} \sum_i y_i \ell'(y_i \mathbf{f}^T \mathbf{x}_i) \mathbf{x}_i. \tag{59}$$

Moreover, recall that $\nabla J(\mathbf{f}^*, \mathcal{D}) = 0$, from the optimality of $\mathbf{f}^*$. Therefore,

$$\begin{aligned} \nabla J(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}}, \mathcal{D}) = \nabla J(\mathbf{f}^*, \mathcal{D}) &- \Lambda(\nabla N(\mathbf{f}^*) - \nabla N(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})) \\ &- \frac{1}{n} \sum_i y_i \left(\ell'(y_i (\mathbf{f}^*)^T \mathbf{x}_i) - \ell'(y_i (t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})^T \mathbf{x}_i) \right) \mathbf{x}_i. \qquad (60)\end{aligned}$$

Now, from the Lipschitz condition on ℓ, for each i we can upper bound each term in the summation above:

$$\begin{aligned} &\left\| y_i \left(\ell'(y_i (\mathbf{f}^*)^T \mathbf{x}_i) - \ell'(y_i (t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})^T \mathbf{x}_i) \right) \mathbf{x}_i \right\| \\ &\qquad \leq |y_i| \cdot ||\mathbf{x}_i|| \cdot |\ell'(y_i (\mathbf{f}^*)^T \mathbf{x}_i) - \ell'(y_i (t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})^T \mathbf{x}_i)| \\ &\qquad \leq |y_i| \cdot ||\mathbf{x}_i|| \cdot c \cdot |y_i (1-t)(\mathbf{f}^* - \mathbf{f}_{\text{priv}})^T \mathbf{x}_i| \\ &\qquad \leq c(1-t)|y_i|^2 \cdot ||\mathbf{x}_i||^2 \cdot ||\mathbf{f}^* - \mathbf{f}_{\text{priv}}|| \\ &\qquad \leq c(1-t)||\mathbf{f}^* - \mathbf{f}_{\text{priv}}||. \qquad (61)\end{aligned}$$

The third step follows because ℓ' is c-Lipschitz and the last step follows from the bounds on $|y_i|$ and $||\mathbf{x}_i||$. Because N is doubly differentiable, we can apply the Mean Value Theorem again to conclude that

$$||\nabla N(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}}) - \nabla N(\mathbf{f}^*)|| \leq (1-t)||\mathbf{f}_{\text{priv}} - \mathbf{f}^*|| \cdot ||\nabla^2 N(\mathbf{f}'')||_2 \tag{62}$$

for some $\mathbf{f}'' \in \mathbb{R}^d$.

As $0 \leq t \leq 1$, we can combine (60), (61), and (62) to obtain

$$\begin{aligned} \|\nabla J(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}}, \mathcal{D})\| &\leq \|\Lambda(\nabla N(\mathbf{f}^*) - \nabla N(t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}}))\| \\ &\qquad + \left\| \frac{1}{n} \sum_i y_i (\ell'(y_i (\mathbf{f}^*)^T \mathbf{x}_i) - \ell'(y_i (t\mathbf{f}^* + (1-t)\mathbf{f}_{\text{priv}})^T \mathbf{x}_i)) \mathbf{x}_i \right\| \\ &\leq (1-t)\,\|\mathbf{f}_{\text{priv}} - \mathbf{f}^*\| \cdot \left(\Lambda\eta + \frac{1}{n} \cdot n \cdot c \right) \\ &\leq \|\mathbf{f}_{\text{priv}} - \mathbf{f}^*\|\,(\Lambda\eta + c). \qquad (63)\end{aligned}$$

From the definition of Algorithm 1, $\mathbf{f}_{\text{priv}} - \mathbf{f}^* = \mathbf{b}$, where $\mathbf{b}$ is the noise vector. Now we can apply Lemma 5 to $||\mathbf{f}_{\text{priv}} - \mathbf{f}^*||$, with parameters $k = d$, and $\theta = \frac{2}{\Lambda n \epsilon_p}$. From Lemma 5, with probability $1 - \delta$, $||\mathbf{f}_{\text{priv}} - \mathbf{f}^*|| \leq \frac{2d \log(\frac{d}{\delta})}{\Lambda n \epsilon_p}$. The Lemma follows by combining this with Equations 63 and 58.

□

Lemma 5. *Let X be a random variable drawn from the distribution $\Gamma(k, \theta)$, where k is an integer. Then,*

$$\mathbb{P}\left(X < k\theta \log\left(\frac{k}{\delta}\right)\right) \geq 1 - \delta. \tag{64}$$

Proof. Since k is an integer, we can decompose X distributed according to $\Gamma(k, \theta)$ as a summation

$$X = X_1 + \ldots + X_k, \tag{65}$$

where $X_1, X_2, \ldots, X_k$ are independent exponential random variables with mean θ. For each i we have $\mathbb{P}(X_i \geq \theta \log(k/\delta)) = \delta/k$. Now,

$$\begin{aligned} \mathbb{P}(X < k\theta \log(k/\delta)) &\geq \mathbb{P}(X_i < \theta \log(k/\delta)\ i = 1, 2, \ldots, k) && (66) \\ &= (1 - \delta/k)^k && (67) \\ &\geq 1 - \delta. && (68) \end{aligned}$$

□

4.2 Objective perturbation

We now establish performance bounds on Algorithm 2. The bound can be summarized as follows.

Theorem 4. *Let $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$, and let $\mathbf{f}_0$ be a classifier with expected loss $L(\mathbf{f}_0) = L^*$. Let ℓ be convex, doubly differentiable, and let its derivatives satisfy $|\ell'(z)| \leq 1$ and $|\ell''(z)| \leq c$ for all z. If the n training samples in $\mathcal{D}$ are drawn i.i.d. accoding to P, and if*

$$n > C \max\left(\frac{||\mathbf{f}_0||^2 \log(1/\delta)}{\epsilon_g^2}, \frac{c||\mathbf{f}_0||^2}{\epsilon_g \epsilon_p}, \frac{d \log(\frac{d}{\delta})||\mathbf{f}_0||}{\epsilon_g \epsilon_p}\right) \tag{69}$$

for some constant C, then, the output $\mathbf{f}_{\text{priv}}$ of Algorithm 2 satisfies

$$\mathbb{P}\left(L(\mathbf{f}_{\text{priv}}) \leq L^* + \epsilon_g\right) \geq 1 - 2\delta. \tag{70}$$

Proof. Let

$$\begin{aligned} \mathbf{f}_{\text{rtr}} &= \operatorname*{argmin}_{\mathbf{f}} \bar{J}(\mathbf{f}) \\ \mathbf{f}^* &= \operatorname*{argmin}_{\mathbf{f}} J(\mathbf{f}, \mathcal{D}), \end{aligned}$$

and $\mathbf{f}_{\text{priv}}$ denote the output of Algorithm 1. As in Theorem 3, the analysis of Shalev-Shwartz and Srebro [2008] shows

$$L(\mathbf{f}_{\text{priv}}) = L(\mathbf{f}_0) + (\bar{J}(\mathbf{f}_{\text{priv}}) - \bar{J}(\mathbf{f}_{\text{rtr}})) + (\bar{J}(\mathbf{f}_{\text{rtr}}) - \bar{J}(\mathbf{f}_0)) + \frac{\Lambda}{2}||\mathbf{f}_0||^2 - \frac{\Lambda}{2}||\mathbf{f}_{\text{priv}}||^2. \tag{71}$$

We will bound each of the terms on the right-hand-side.

If $n > \frac{c||\mathbf{f}_0||^2}{\epsilon_g \epsilon_p}$ and $\Lambda > \frac{\epsilon_g}{4||\mathbf{f}_0||^2}$, then $n\Lambda > \frac{c}{4\epsilon_p}$, so from the definition of ϵ'_p in Algorithm 2,

$$\epsilon'_p = \epsilon_p - 2\log\left(1 + \frac{c}{n\Lambda}\right) = \epsilon_p - 2\log\left(1 + \frac{\epsilon_p}{4}\right) \geq \epsilon_p - \frac{\epsilon_p}{2}, \tag{72}$$

where the last step follows because $\log(1+x) \leq x$ for $x \in [0,1]$. Note that for these values of Λ we have $\epsilon'_p > 0$.

Therefore, we can apply Lemma 6 to conclude that with probability at least $1 - \delta$ over the privacy mechanism,

$$J(\mathbf{f}_{\text{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D}) \leq \frac{4d^2 \log^2(d/\delta)}{\Lambda n^2 \epsilon_p^2}. \tag{73}$$

From Sridharan et al. [2008],

$$\bar{J}(\mathbf{f}_{\text{priv}}) - \bar{J}(\mathbf{f}_{\text{rtr}}) \leq 2(J(\mathbf{f}_{\text{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D})) + O\left(\frac{\log(1/\delta)}{\Lambda n}\right) \tag{74}$$

$$\leq \frac{8d^2 \log^2(d/\delta)}{\Lambda n^2 \epsilon_p^2} + O\left(\frac{\log(1/\delta)}{\Lambda n}\right). \tag{75}$$

By definition of $\mathbf{f}^*$, we have $\bar{J}(\mathbf{f}_{\text{rtr}}) - \bar{J}(\mathbf{f}_0) \leq 0$. If Λ is set to be $\frac{\epsilon_g}{||\mathbf{f}_0||^2}$, then, the fourth quantity in Equation 71 is at most $\frac{\epsilon_g}{2}$. The theorem follows. □

The following lemma is analogous to Lemma 4, and it establishes a bound on the distance between the output of Algorithm 2, and non-private regularized ERM. We note that this bound holds when Algorithm 2 has $\epsilon'_p > 0$, that is, when $\Delta = 0$. Ensuring that $\Delta = 0$ requires an additional condition on n, which is stated in Theorem 4.

Lemma 6. *Let $\epsilon'_p > 0$. Let $\mathbf{f}^* = \operatorname{argmin} J(\mathbf{f}, \mathcal{D})$, and let $\mathbf{f}_{\text{priv}}$ be the classifier output by Algorithm 2. If $N(\cdot)$ is 1-strongly convex and globally differentiable, and if ℓ is convex and differentiable at all points, with $|\ell'(z)| \leq 1$ for all z, then*

$$\mathbb{P}_{\mathbf{b}}\left(J(\mathbf{f}_{\text{priv}}, \mathcal{D}) \leq J(\mathbf{f}^*, \mathcal{D}) + \frac{4d^2 \log^2(d/\delta)}{\Lambda n^2 \epsilon_p^2}\right) \geq 1 - \delta, \tag{76}$$

where the probability is taken over the randomness in the noise $\mathbf{b}$ of Algorithm 2.

Proof. By the assumption $\epsilon'_p > 0$, the classifier $\mathbf{f}_{\text{priv}}$ minimizes the objective function $J(\mathbf{f}, \mathcal{D}) + \frac{1}{n}\mathbf{b}^T\mathbf{f}$, and therefore

$$J(\mathbf{f}_{\text{priv}}, \mathcal{D}) \leq J(\mathbf{f}^*, \mathcal{D}) + \frac{1}{n}\mathbf{b}^T(\mathbf{f}^* - \mathbf{f}_{\text{priv}}). \tag{77}$$

First, we try to bound $||\mathbf{f}^* - \mathbf{f}_{\text{priv}}||$. Recall that $\Lambda N(\cdot)$ is Λ-strongly convex and globally differentiable, and ℓ is convex and differentiable. We can therefore apply Lemma 1 with $G(\mathbf{f}) = J(\mathbf{f}, \mathcal{D})$ and $g(\mathbf{f}) = \frac{1}{n}\mathbf{b}^T\mathbf{f}$ to obtain the bound

$$||\mathbf{f}^* - \mathbf{f}_{\text{priv}}|| \leq \frac{1}{\Lambda}\left\|\nabla(\frac{1}{n}\mathbf{b}^T\mathbf{f})\right\| \leq \frac{||\mathbf{b}||}{n\Lambda}. \tag{78}$$

Therefore by the Cauchy-Schwartz inequality,

$$J(\mathbf{f}_{\text{priv}}, \mathcal{D}) - J(\mathbf{f}^*, \mathcal{D}) \leq \frac{||\mathbf{b}||^2}{n^2 \Lambda}. \tag{79}$$

Since $||\mathbf{b}||$ is drawn from a $\Gamma(d, \frac{2}{\epsilon_p})$ distribution, from Lemma 5, with probability $1 - \delta$, $||\mathbf{b}|| \leq \frac{2d \log(d/\delta)}{\epsilon_p}$. The Lemma follows by plugging this in to the previous equation. □

4.3 Applications

In this section, we examine the sample requirement of privacy-preserving regularized logistic regression and support vector machines. Recall that in both these cases, $N(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2$.

Corollary 5 (Logistic Regression)**.** *Let training data $\mathcal{D}$ be generated i.i.d. accoding to a distribution P and let $\mathbf{f}_0$ be a classifier with expected loss $L(\mathbf{f}_0) = L^*$. Let the loss function $\ell = \ell_{\text{LR}}$.*

1. *There exists a C_1 such that if*

$$n > C_1 \max \left(\frac{||\mathbf{f}_0||^2 \log(\frac{1}{\delta})}{\epsilon_g^2}, \frac{d \log(\frac{d}{\delta}) ||\mathbf{f}_0||}{\epsilon_g \epsilon_p}, \frac{d \log(\frac{d}{\delta}) ||\mathbf{f}_0||^2}{\epsilon_g^{3/2} \epsilon_p} \right) \tag{80}$$

then the output $\mathbf{f}_{\text{priv}}$ of Algorithm 1 satisfies

$$\mathbb{P}\left(L(\mathbf{f}_{\text{priv}}) \leq L^* + \epsilon_g\right) \geq 1 - \delta. \tag{81}$$

2. *There exists a C_2 such that if*

$$n > C \max \left(\frac{||\mathbf{f}_0||^2 \log(1/\delta)}{\epsilon_g^2}, \frac{||\mathbf{f}_0||^2}{\epsilon_g \epsilon_p}, \frac{d \log(\frac{d}{\delta}) ||\mathbf{f}_0||}{\epsilon_g \epsilon_p} \right) \tag{82}$$

then the output $\mathbf{f}_{\text{priv}}$ of Algorithm 2 with $c = \frac{1}{4}$ satisfies

$$\mathbb{P}\left(L(\mathbf{f}_{\text{priv}}) \leq L^* + \epsilon_g\right) \geq 1 - \delta. \tag{83}$$

Proof. Since ℓ_{LR} is convex and doubly differentiable for any z_1, z_2,

$$\ell'_{\text{LR}}(z_1) - \ell'_{\text{LR}}(z_2) \leq \ell''_{\text{LR}}(z^*)(z_1 - z_2) \tag{84}$$

for some $z^* \in [z_1, z_2]$. Moreover, $|\ell''_{\text{LR}}(z^*)| \leq c = \frac{1}{4}$, so ℓ' is $\frac{1}{4}$-Lipschitz. The corollary now follows from Theorems 3 and 4. □

For SVMs we state results with $\ell = \ell_{\text{Huber}}$, but a similar bound can be shown for ℓ_s as well.

Corollary 6 (Huber Support Vector Machines)**.** *Let training data $\mathcal{D}$ be generated i.i.d. accoding to a distribution P and let $\mathbf{f}_0$ be a classifier with expected loss $L(\mathbf{f}_0) = L^*$. Let the loss function $\ell = \ell_{\text{Huber}}$.*

1. *There exists a* C_1 *such that if*

$$n > C_1 \max\left(\frac{||\mathbf{f}_0||^2 \log(\frac{1}{\delta})}{\epsilon_g^2}, \frac{d\log(\frac{d}{\delta})||\mathbf{f}_0||}{\epsilon_g \epsilon_p}, \frac{d\log(\frac{d}{\delta})||\mathbf{f}_0||^2}{h^{1/2}\epsilon_g^{3/2}\epsilon_p}\right) \tag{85}$$

   then the output $\mathbf{f}_{\mathrm{priv}}$ *of Algorithm 1 satisfies*

$$\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) \le L^* + \epsilon_g\right) \ge 1 - \delta. \tag{86}$$

2. *There exists a* C_2 *such that if*

$$n > C \max\left(\frac{||\mathbf{f}_0||^2 \log(1/\delta)}{\epsilon_g^2}, \frac{||\mathbf{f}_0||^2}{h\epsilon_g\epsilon_p}, \frac{d\log(\frac{d}{\delta})||\mathbf{f}_0||}{\epsilon_g\epsilon_p}\right) \tag{87}$$

   then the output $\mathbf{f}_{\mathrm{priv}}$ *of Algorithm 2 with* $c = \frac{1}{4}$ *satisfies*

$$\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) \le L^* + \epsilon_g\right) \ge 1 - \delta. \tag{88}$$

Proof. The Huber loss is convex and differentiable with continuous derivatives. Moreover, since the derivative of the Huber loss is piecewise linear with slope 0 or at most $\frac{1}{2h}$, for any z_1, z_2,

$$|\ell'_{\mathrm{Huber}}(z_1) - \ell'_{\mathrm{Huber}}(z_2)| \le \frac{1}{2h}|z_1 - z_2|, \tag{89}$$

so ℓ'_{Huber} is $\frac{1}{2h}$-Lipschitz. The first part of the corollary follows from Theorem 3.

For the second part of the corollary, we observe that from Corollary 4, we do not need ℓ to be globally double differentiable, and the bound on $|\ell''(z)|$ in Theorem 4 is only needed to ensure that $\epsilon'_p > 0$; since ℓ_{Huber} is double differentiable except in a set of Lebesgue measure 0, with $|\ell''_{\mathrm{Huber}}(z)| \le \frac{1}{2h}$, the corollary follows by an application of Theorem 4. □

5 Kernel methods

A powerful methodology in learning problems is the "kernel trick," which allows the efficient construction of a predictor $\mathbf{f}$ that resides in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated to a positive definite kernel function $k(\cdot, \cdot)$. The representer theorem [Kimeldorf and Wahba, 1970] shows that the regularized empirical risk in (1) is minimized by a function $\mathbf{f}(\mathbf{x})$ that is given by a linear combination of kernel functions centered at the data points:

$$\mathbf{f}^*(\mathbf{x}) = \sum_{i=1}^{n} a_i k(\mathbf{x}(i), \mathbf{x}). \tag{90}$$

This elegant result is appealing from a theoretical standpoint. To use it in practice, one releases the values a_i corresponding to the $\mathbf{f}$ that minimizes the empirical risk, along with the data points $\mathbf{x}(i)$; the user classifies a new $\mathbf{x}$ by evaluating the function in Equation 90.

A crucial difficulty in terms of privacy is that this directly releases the private values $\mathbf{x}(i)$ of some individuals in the training set. Thus, even if the classifier is computed in a privacy-preserving way, any classifier released by this process requires revealing the data. We provide an algorithm that avoids this problem, using an approximation method [Rahimi and Recht, 2007, 2008b] to approximate the kernel function using random projections.

5.1 Mathematical preliminaries

We will assume that the kernel function is translation invariant, so $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$. The key idea in the random projection method is from Bochner's Theorem, which states that a continuous translation invariant kernel is positive definite if and only if it is the Fourier transform of a nonnegative measure. This means that the Fourier transform $K(\theta)$ of translation-invariant kernel function $k(\mathbf{t})$ can be normalized so that $\bar{K}(\theta) = K(\theta)/\left\|K(\theta)\right\|_1$ is a probability measure on the transform space Θ. We will assume $\bar{K}(\theta)$ is uniformly bounded over θ.

In this representation

$$k(\mathbf{x}, \mathbf{x}') = \int_{\Theta} \phi(\mathbf{x}; \theta) \phi(\mathbf{x}'; \theta) \bar{K}(\theta) d\theta, \tag{91}$$

where we will assume the feature functions $\phi(\mathbf{x}; \theta)$ are bounded:

$$|\phi(\mathbf{x}; \theta)| \leq \zeta \qquad \forall \mathbf{x} \in \mathcal{X}, \ \forall \theta \in \Theta. \tag{92}$$

A function $\mathbf{f} \in \mathcal{H}$ can be written as

$$\mathbf{f}(\mathbf{x}) = \int_{\Theta} a(\theta) \phi(\mathbf{x}; \theta) \bar{K}(\theta) d\theta. \tag{93}$$

To prove our generalization bounds we must show that bounded classifiers $\mathbf{f}$ induce bounded functions $a(\theta)$. Writing the evaluation functional as an inner product with $k(\mathbf{x}, \mathbf{x}')$ and (91) shows

$$\mathbf{f}(\mathbf{x}) = \int_{\Theta} \left(\int_{\mathcal{X}} \mathbf{f}(\mathbf{x}') \phi(\mathbf{x}'; \theta) d\mathbf{x}' \right) \phi(\mathbf{x}; \theta) \bar{K}(\theta) d\theta. \tag{94}$$

Thus we have

$$a(\theta) = \int_{\mathcal{X}} \mathbf{f}(\mathbf{x}') \phi(\mathbf{x}'; \theta) d\mathbf{x}' \tag{95}$$

$$|a(\theta)| \leq \mathrm{Vol}(\mathcal{X}) \cdot \zeta \cdot \left\|\mathbf{f}\right\|_{\infty}. \tag{96}$$

This shows that $a(\theta)$ is bounded uniformly over Θ when $\mathbf{f}(\mathbf{x})$ is bounded uniformly over $\mathcal{X}$. The volume of the unit ball is $\mathrm{Vol}(\mathcal{X}) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}$ (see Ball [1997] for more details). For large d this is $(\sqrt{\frac{2\pi e}{d}})^d$ by Stirling's formula. Furthermore, we have

$$\left\|\mathbf{f}\right\|_{\mathcal{H}}^2 = \int_{\Theta} a(\theta)^2 \bar{K}(\theta) d\theta. \tag{97}$$

5.2 A reduction to the linear case

We now describe how to apply Algorithms 1 and 2 for classification with kernels, by transforming to linear classification. Given $\{\theta_j\}$, let $R : \mathcal{X} \to \mathbb{R}^D$ be the map that sends $\mathbf{x}(i)$ to a vector $\mathbf{v}(i) \in \mathbb{R}^D$ where $\mathbf{v}_j(i) = \phi(\mathbf{x}(i); \theta_j)$ for $j \in [D]$. We then use Algorithm 1 or Algorithm 2 to compute a privacy-preserving linear classifier $\mathbf{f}$ in $\mathbb{R}^D$. The algorithm releases R and $\tilde{\mathbf{f}}$. The overall classifier is $\mathbf{f}_{\text{priv}}(\mathbf{x}) = \tilde{\mathbf{f}}(R(\mathbf{x}))$.

```
Algorithm 3 Private ERM for nonlinear kernels
  Inputs: Data $\{(\mathbf{x}_i, y_i) : i \in [n]\}$, positive definite kernel function $k(\cdot, \cdot)$, sampling function $\bar{K}(\theta)$, parameters $\epsilon_p$, $\Lambda$, $D$
  Outputs: Predictor $\mathbf{f}_{\mathrm{priv}}$ and pre-filter $\{\theta_j : j \in [D]\}$.
  Draw $\{\theta_j : j = 1, 2, \ldots, D\}$ iid according to $\bar{K}(\theta)$.
  Set $\mathbf{v}(i) = \sqrt{2/D}[\phi(\mathbf{x}(i); \theta_1) \cdots \phi(\mathbf{x}(i); \theta_D)]^T$ for each $i$.
  Run Algorithm 1 or Algorithm 2 with data $\{(\mathbf{v}(i), y(i))\}$ and parameters $\epsilon_p$, $\Lambda$.
```

As an example, consider the Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}'\right\|_2^2\right). \tag{98}$$

The Fourier transform of a Gaussian is a Gaussian, so we can sample $\theta_j = (\omega, \psi)$ according to the distribution Uniform$[-\pi, \pi] \times \mathcal{N}(0, 2\gamma I_d)$ and compute $v_j = \cos(\omega^T \mathbf{x} + \psi)$. The random phase is used to produce a real-valued mapping. The paper of Rahimi and Recht [2008a] has more examples of transforms.

5.3 Privacy guarantees

Because the workhorse of Algorithm 3 is Algorithm 1 (or Algorithm 2), and the points $\{\theta_j : j \in [D]\}$ are independent of the data, the privacy guarantees for Algorithm 3 follow trivially from Theorems 1 and 2.

Theorem 5. *Given data $\{(\mathbf{x}(i), y(i)) : i = 1, 2, \ldots, n\}$ with $(\mathbf{x}(i), y(i))$ and $\|\mathbf{x}(i)\| \le 1$, the outputs $(\mathbf{f}_{\mathrm{priv}}, \{\theta_j : j \in [D]\})$ of Algorithm 3 guarantee ϵ_p-differential privacy.*

The proof trivially follows from a combination of Theorem 1 or 2, and the fact that the θ_js are drawn independently of the input dataset.

5.4 Generalization performance

We now turn to generalization bounds for Algorithm 3. We will prove results using Algorithm 2 in Algorithm 3, but analogous results for Algorithm 1 are simple to prove. Our comparisons will be against arbitrary predictors $\mathbf{f}_0$ whose norm is bounded in some sense. That is, given an $\mathbf{f}_0$ with some properties, we will choose regularization parameter Λ, dimension D, and number of samples n so that the predictor $\mathbf{f}_{\mathrm{priv}}$ has expected loss close to that of $\mathbf{f}_0$.

In this section we will assume $N(\mathbf{f}) = \frac{1}{2}\left\|\mathbf{f}\right\|^2$ so that $N(\cdot)$ is 1-strongly convex, and that the loss function ℓ is convex, differentiable and $|\ell'(z)| \le 1$ for all z.

Our first generalization result is the simplest, since it assumes a strong condition that gives easy guarantees on the projections. We would like the predictor produced by Algorithm 3 to be competitive against an $\mathbf{f}_0$ such that

$$\mathbf{f}_0(\mathbf{x}) = \int_\Theta a_0(\theta)\phi(\mathbf{x}; \theta)\bar{K}(\theta) d\theta, \tag{99}$$

and $|a_0(\theta)| \le C$ (see Rahimi and Recht [2008b]). Our first result provides the technical building block for our other generalization results. The proof makes use of ideas from Rahimi and Recht [2008b] and techniques from Sridharan et al. [2008], Shalev-Shwartz and Srebro [2008].

Lemma 7. *Let $\mathbf{f}_0$ be a predictor such that $|a_0(\theta)| \leq C$, for all θ, where $a_0(\theta)$ is given by (99), and suppose $L(\mathbf{f}_0) = L^*$. Moreover, suppose that $\ell'(\cdot)$ is c-Lipschitz. If the data $\mathcal{D}$ is drawn i.i.d. according to P, then there exists a constant C_0 such that if*

$$n > C_0 \cdot \max\left(\frac{C^2\sqrt{\log(1/\delta)}}{\epsilon_p \epsilon_g^2} \cdot \log \frac{C \log(1/\delta)}{\epsilon_g \delta}, \frac{c\epsilon_g}{\epsilon_p \log(1/\delta)} \right), \tag{100}$$

then Λ and D can be chosen such that the output $\mathbf{f}_{\mathrm{priv}}$ of Algorithm 3 using Algorithm 2 satisfies

$$\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) - L^* \leq \epsilon_g\right) \geq 1 - 4\delta. \tag{101}$$

Proof. Since $|a_0(\theta)| \leq C$ and the $\bar{K}(\theta)$ is bounded, we have [Rahimi and Recht, 2008b, Theorem 1] that with probability $1 - 2\delta$ there exists an $\mathbf{f}_p \in \mathbb{R}^D$ such that

$$L(\mathbf{f}_p) \leq L(\mathbf{f}_0) + O\left(\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{D}} \right) C \sqrt{\log \frac{1}{\delta}} \right), \tag{102}$$

We will choose D to make this loss small. Furthermore, $\mathbf{f}_p$ is guaranteed to have $\|\mathbf{f}_p\|_\infty \leq C/D$, so

$$\|\mathbf{f}_p\|_2^2 \leq \frac{C^2}{D}. \tag{103}$$

Now given such an $\mathbf{f}_p$ we must show that $\mathbf{f}_{\mathrm{priv}}$ will have true risk close to that of $\mathbf{f}_p$ as long as there are enough data points. This can be shown using the techniques in Shalev-Shwartz and Srebro [2008]. Let

$$\bar{J}(\mathbf{f}) = L(\mathbf{f}) + \frac{\Lambda}{2} \|\mathbf{f}\|_2^2, \tag{104}$$

and let

$$\mathbf{f}_{\mathrm{rtr}} = \operatorname*{argmin}_{\mathbf{f} \in \mathbb{R}^D} \bar{J}(\mathbf{f}) \tag{105}$$

minimize the regularized true risk. Then

$$\bar{J}(\mathbf{f}_{\mathrm{priv}}) = \bar{J}(\mathbf{f}_p) + (\bar{J}(\mathbf{f}_{\mathrm{priv}}) - \bar{J}(\mathbf{f}_{\mathrm{rtr}})) + (\bar{J}(\mathbf{f}_{\mathrm{rtr}}) - \bar{J}(\mathbf{f}_p)). \tag{106}$$

Now, since $\bar{J}(\cdot)$ is minimized by $\mathbf{f}_{\mathrm{rtr}}$, the last term is negative and we can disregard it. Then we have

$$L(\mathbf{f}_{\mathrm{priv}}) - L(\mathbf{f}_p) \leq (\bar{J}(\mathbf{f}_{\mathrm{priv}}) - \bar{J}(\mathbf{f}_{\mathrm{rtr}})) + \frac{\Lambda}{2} \|\mathbf{f}_p\|_2^2 - \frac{\Lambda}{2} \|\mathbf{f}_{\mathrm{priv}}\|_2^2. \tag{107}$$

From Lemma 6, with probability at least $1 - \delta$ over the noise $\mathbf{b}$,

$$J(\mathbf{f}_{\mathrm{priv}}) - J\left(\operatorname*{argmin}_{\mathbf{f}} J(\mathbf{f}) \right) \leq \frac{4D^2 \log^2(D/\delta)}{\Lambda n^2 \epsilon_p^2}. \tag{108}$$

Now using [Sridharan et al., 2008, Corollary 2], we can bound the term $(\bar{J}(\mathbf{f}_{\text{priv}}) - \bar{J}(\mathbf{f}_{\text{rtr}}))$ by twice the gap in the regularized empirical risk difference (108) plus an additional term. That is, with probability $1 - \delta$:

$$\bar{J}(\mathbf{f}_{\text{priv}}) - \bar{J}(\mathbf{f}_{\text{rtr}}) \leq 2(J(\mathbf{f}_{\text{priv}}) - J(\mathbf{f}_{\text{rtr}})) + O\left(\frac{\log(1/\delta)}{\Lambda n}\right) \tag{109}$$

If we set $n > \frac{c}{4\epsilon_p \Lambda}$, then, $\epsilon'_p > 0$, and we can plug in Lemma 6 to Equation 109 to write:

$$\bar{J}(\mathbf{f}_{\text{priv}}) - \bar{J}(\mathbf{f}_{\text{rtr}}) \leq \frac{8D^2 \log^2(D/\delta)}{\Lambda n^2 \epsilon_p^2} + O\left(\frac{\log(1/\delta)}{\Lambda n}\right). \tag{110}$$

Plugging (110) into (107), discarding the negative term involving $\|\mathbf{f}_{\text{priv}}\|_2^2$ and setting $\Lambda = \epsilon_g / \|\mathbf{f}_0\|^2$ gives

$$L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_p) \leq \frac{8\,\|\mathbf{f}_p\|_2^2\, D^2 \log^2(D/\delta)}{n^2 \epsilon_p^2 \epsilon_g} + O\left(\frac{\|\mathbf{f}_p\|_2^2 \log\frac{1}{\delta}}{n\epsilon_g}\right) + \frac{\epsilon_g}{2}. \tag{111}$$

Now we have, using (102) and (111), that with probability $1 - 4\delta$:

$$\begin{aligned} L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_0) &\leq (L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_p)) + (L(\mathbf{f}_p) - L(\mathbf{f}_0)) && (112) \\ &\leq \frac{8\,\|\mathbf{f}_p\|_2^2\, D^2 \log^2(D/\delta)}{n^2 \epsilon_p^2 \epsilon_g} + O\left(\frac{\|\mathbf{f}_p\|_2^2 \log(1/\delta)}{n\epsilon_g}\right) + \frac{\epsilon_g}{2} \\ &\qquad + O\left(\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{D}}\right) C \sqrt{\log\frac{1}{\delta}}\right), && (113) \end{aligned}$$

Substituting (103), we have

$$\begin{aligned} L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_0) \leq &\frac{8C^2 D \log^2(D/\delta)}{n^2 \epsilon_p^2 \epsilon_g} + O\left(\frac{C^2 \log(1/\delta)}{Dn\epsilon_g}\right) + \frac{\epsilon_g}{2} \\ &+ O\left(\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{D}}\right) C \sqrt{\log\frac{1}{\delta}}\right). \end{aligned}$$

To set the parameters, we will choose $D < n$ so that

$$L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_0) \leq \frac{8C^2 D \log^2(D/\delta)}{n^2 \epsilon_p^2 \epsilon_g} + O\left(\frac{C^2 \log(1/\delta)}{Dn\epsilon_g}\right) + \frac{\epsilon_g}{2} + O\left(\frac{C\sqrt{\log(1/\delta)}}{\sqrt{D}}\right).$$

Focusing on the last term, we set $D = O(C^2 \log(1/\delta)/\epsilon_g^2)$ to make the last term $\epsilon_g/6$, and:

$$L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_0) \leq O\left(\frac{C^4 \log\frac{1}{\delta} \log^2 \frac{C^2 \log(1/\delta)}{\epsilon_g^2 \delta}}{n^2 \epsilon_p^2 \epsilon_g^3}\right) + O\left(\frac{\epsilon_g}{n}\right) + \frac{2\epsilon_g}{3}. \tag{114}$$

Setting n as in (100) proves the result. Moreover, setting $n > \frac{c\|\mathbf{f}_p\|^2}{4\epsilon_p \epsilon_g} = C_0 \cdot \frac{c\epsilon_g}{\epsilon_p \log(1/\delta)}$ ensures that $n > \frac{c}{4\Lambda\epsilon_p}$. □

We can adapt the proof procedure to show that we can be competitive against any classifier $\mathbf{f}_0$ with a given bound on $\|\mathbf{f}_0\|_\infty$. It can be shown that for some constant ζ that $|a_0(\theta)| \leq \mathrm{Vol}(\mathcal{X})\zeta \|\mathbf{f}_0\|_\infty$. Then we can set this as C in (100) to obtain our next result.

Theorem 6. *Let $\mathbf{f}_0$ be a classifier with norm $\|\mathbf{f}_0\|_\infty$, and let $\ell'(\cdot)$ be c-Lipschitz. Then for any distribution P, there exists a constant C_0 such that if*

$$n > C_0 \cdot \max \left(\frac{\|\mathbf{f}_0\|_\infty^2 \zeta^2 (\mathrm{Vol}(\mathcal{X}))^2 \sqrt{\log(1/\delta)}}{\epsilon_p \epsilon_g^2} \cdot \log \frac{\|\mathbf{f}_0\|_\infty \mathrm{Vol}(\mathcal{X})\zeta \log(1/\delta)}{\epsilon_g \delta \Gamma(\frac{d}{2}+1)}, \frac{c\epsilon_g}{\epsilon_p \log(1/\delta)} \right), \quad (115)$$

then Λ and D can be chosen such that the output $\mathbf{f}_{\mathrm{priv}}$ of Algorithm 3 with Algorithm 2 satisfies $\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) - L(\mathbf{f}_0) \leq \epsilon_g\right) \geq 1 - 4\delta$.

Proof. Substituting $C = \mathrm{Vol}(\mathcal{X})\zeta \|\mathbf{f}_0\|_\infty$ in Lemma 7 we get the result. □

We can also derive a generalization result with respect to classifiers with bounded $\|\mathbf{f}_0\|_{\mathcal{H}}$.

Theorem 7. *Let $\mathbf{f}_0$ be a classifier with norm $\|\mathbf{f}_0\|_{\mathcal{H}}$, and let ℓ' be c-Lipschitz. Then for any distribution P, there exists a constant C_0 such that if,*

$$n = C_0 \cdot \max \left(\frac{\|\mathbf{f}_0\|_{\mathcal{H}}^4 \zeta^2 (\mathrm{Vol}(\mathcal{X}))^2 \sqrt{\log(1/\delta)}}{\epsilon_p \epsilon_g^4} \cdot \log \frac{\|\mathbf{f}_0\|_{\mathcal{H}} \mathrm{Vol}(\mathcal{X})\zeta \log(1/\delta)}{\epsilon_g \delta \Gamma(\frac{d}{2}+1)}, \frac{c\epsilon_g}{\epsilon_p \log(1/\delta)} \right), \quad (116)$$

then Λ and D can be chosen such that the output of Algorithm 3 run with Algorithm 2 satisfies $\mathbb{P}\left(L(\mathbf{f}_{\mathrm{priv}}) - L(\mathbf{f}_0) \leq \epsilon_g\right) \geq 1 - 4\delta$.

Proof. Let $\mathbf{f}_0$ be a classifier with norm $\|\mathbf{f}_0\|_{\mathcal{H}}^2$ and expected loss $L(\mathbf{f}_0)$. Now consider

$$\mathbf{f}_{\mathrm{rtr}} = \operatorname*{argmin}_{\mathbf{f}} L(\mathbf{f}) + \frac{\Lambda_{\mathrm{rtr}}}{2} \|\mathbf{f}\|_{\mathcal{H}}^2, \quad (117)$$

for some Λ_{rtr} to be specified later. We will first need a bound on $\|\mathbf{f}_{\mathrm{rtr}}\|_\infty$ in order to use our previous sample complexity results. Since $\mathbf{f}_{\mathrm{rtr}}$ is a minimizer, we can take the derivative of the regularized expected loss and set it to 0 to get:

$$\mathbf{f}_{\mathrm{rtr}}(\mathbf{x}') = \frac{-1}{\Lambda_{\mathrm{rtr}}} \left(\frac{\partial}{\partial \mathbf{f}} \int_{\mathcal{X}} \ell(\mathbf{f}(\mathbf{x}'), y) dP(\mathbf{x}, y) \right) \quad (118)$$

$$= \frac{-1}{\Lambda_{\mathrm{rtr}}} \left(\int_{\mathcal{X}} \left(\frac{\partial}{\partial \mathbf{f}(\mathbf{x}')} \ell(\mathbf{f}(\mathbf{x}), y) \right) \cdot \left(\frac{\partial}{\partial \mathbf{f}(\mathbf{x}')} \mathbf{f}(\mathbf{x}) \right) dP(\mathbf{x}, y) \right), \quad (119)$$

where $P(\mathbf{x}, y)$ is a distribution on pairs $(\mathbf{x}, y)$. Now, using the representer theorem, $\frac{\partial}{\partial \mathbf{f}(\mathbf{x}')}\mathbf{f}(\mathbf{x}) = k(\mathbf{x}', \mathbf{x})$. Since the kernel function is bounded and the derivative of the loss is always upper bounded by 1, so the integrand can be upper bounded by a constant. Since $P(\mathbf{x}, y)$ is a probability distribution, we have for all $\mathbf{x}'$ that $|\mathbf{f}_{\mathrm{rtr}}(\mathbf{x}')| = O(1/\Lambda_{\mathrm{rtr}})$. Now we set $\Lambda_{\mathrm{rtr}} = \epsilon_g / \|\mathbf{f}_0\|_{\mathcal{H}}^2$ to get

$$\|\mathbf{f}_{\mathrm{rtr}}\|_\infty = O\left(\frac{\|\mathbf{f}_0\|_{\mathcal{H}}^2}{\epsilon_g} \right). \quad (120)$$

We now have two cases to consider, depending on whether $L(\mathbf{f}_0) < L(\mathbf{f}_{\mathrm{rtr}})$ or $L(\mathbf{f}_0) > L(\mathbf{f}_{\mathrm{rtr}})$.

Case 1: Suppose that $L(\mathbf{f}_0) < L(\mathbf{f}_{\text{rtr}})$. Then by the definition of $\mathbf{f}_{\text{rtr}}$,

$$L(\mathbf{f}_{\text{rtr}}) + \frac{\epsilon_g}{2} \cdot \frac{\|\mathbf{f}_{\text{rtr}}\|_{\mathcal{H}}^2}{\|\mathbf{f}_0\|_{\mathcal{H}}^2} \leq L(\mathbf{f}_0) + \frac{\epsilon_g}{2}. \tag{121}$$

Since $\frac{\epsilon_g}{2} \cdot \frac{\|\mathbf{f}_{\text{rtr}}\|_{\mathcal{H}}^2}{\|\mathbf{f}_0\|_{\mathcal{H}}^2} \geq 0$, we have $L(\mathbf{f}_{\text{rtr}}) - L(\mathbf{f}_0) \leq \frac{\epsilon_g}{2}$.

Case 2: Suppose that $L(\mathbf{f}_0) > L(\mathbf{f}_{\text{rtr}})$. Then the regularized classifier has better generalization performance than the original, so we have trivially that $L(\mathbf{f}_{\text{rtr}}) - L(\mathbf{f}_0) \leq \frac{\epsilon_g}{2}$.

Therefore in both cases we have a bound on $\|\mathbf{f}_{\text{rtr}}\|_\infty$ and a generalization gap of $\epsilon_g/2$. We can now apply Theorem 6 to show that for

$$n = C_0 \cdot \max\left(\frac{\|\mathbf{f}_0\|_{\mathcal{H}}^4 \, \zeta^2 (\mathrm{Vol}(\mathcal{X}))^2 \sqrt{\log(1/\delta)}}{\epsilon_p \epsilon_g^4} \cdot \log \frac{\|\mathbf{f}_0\|_{\mathcal{H}}^2 \, \zeta \, \mathrm{Vol}(\mathcal{X}) \log(1/\delta)}{\epsilon_g^2 \delta \Gamma(\frac{d}{2}+1)}, \frac{c\epsilon_g}{\epsilon_p \log(1/\delta)} \right), \tag{122}$$

we have $\mathbb{P}\left(L(\mathbf{f}_{\text{priv}}) - L(\mathbf{f}_0) \leq \epsilon_g\right) \geq 1 - 4\delta$. □

6 Parameter tuning

The privacy-preserving learning algorithms presented so far in this paper assume that the regularization constant Λ is provided as an input, and is independent of the data. In actual applications of ERM, Λ is selected based on the data itself. In this section, we address this issue: how to design an ERM algorithm with end-to-end privacy, which selects Λ based on the data itself.

Our solution is to present a privacy-preserving parameter tuning technique that is applicable in general machine learning algorithms, beyond ERM. In practice, one typically tunes parameters (such as the regularization parameter Λ) as follows: using data held out for validation, train predictors $\mathbf{f}(\cdot; \Lambda)$ for multiple values of Λ, and select the one which provides the best empirical performance. However, even though the output of an algorithm preserves ϵ_p-differential privacy for a fixed Λ (as is the case with Algorithms 1 and 2), using a Λ which itself is a function of the private data may violate ϵ_p-differential privacy guarantees. That is, if the procedure that provides Λ is not private, then an adversary may use the released classifier to infer the value of Λ and therefore something about the values in the database.

We suggest two ways of resolving this issue. First, if we have access to some publicly available data from the same distribution, then we can use this as a holdout set to tune Λ. This Λ can be subsequently used to train a classifier on the private data. Since the value of Λ does not depend on the values in the private data set, this procedure will still preserve the privacy of individuals in the private data.

If no such public data is available, then we need a differentially private tuning procedure. We provide such a procedure below. The main idea is to train for different values of Λ on separate subsets of the training dataset, so that the total training procedure still maintains ϵ_p-differential privacy. We score each of these predictors on a validation set, and choose a Λ (and hence $\mathbf{f}(\cdot; \Lambda)$) using a randomized privacy-preserving comparison procedure [McSherry and Talwar, 2007]. The last step is needed to guarantee ϵ_p-differential privacy for individuals in the validation set. This final algorithm provides an end-to-end guarantee of differential privacy, and renders our privacy-preserving ERM procedure complete. We observe that both these procedures can be used for tuning multiple parameters as well.

6.1 Tuning algorithm

Algorithm 4 Privacy-preserving parameter tuning

```
**Inputs:** Database $\mathcal{D}$, parameters $\{\Lambda_1, \ldots, \Lambda_m\}$, $\epsilon_p$.
**Outputs:** Parameter $\mathbf{f}_{\text{priv}}$.
Divide $\mathcal{D}$ into $m+1$ equal portions $\mathcal{D}_1, \ldots, \mathcal{D}_{m+1}$, each of size $\frac{|\mathcal{D}|}{m+1}$.
For each $i = 1, 2, \ldots, m$, apply a privacy-preserving learning algorithm (e.g. Algorithm 1 or 2) on $\mathcal{D}_i$ with parameter $\Lambda_i$ and $\epsilon_p$ to get output $\mathbf{f}_i$.
Evaluate $z_i$, the number of mistakes made by $\mathbf{f}_i$ on $\mathcal{D}_{m+1}$. Set $\mathbf{f}_{\text{priv}} = \mathbf{f}_i$ with probability
    $q_i = \frac{e^{-\epsilon_p z_i/2}}{\sum_{i=1}^{m} e^{-\epsilon_p z_i/2}}.  \qquad (123)$
```

We note that the list of potential Λ values input to this procedure should not be a function of the private dataset. It can also be shown that the empirical error on $\mathcal{D}_{m+1}$ of the classifier output by this procedure is close to the empirical error of the best classifier in the set $\{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ on $\mathcal{D}_{m+1}$, provided $|\mathcal{D}|$ is high enough.

6.2 Privacy and utility

Theorem 8. *The output of the tuning procedure of Algorithm 4 is ϵ_p-differentially private.*

Proof. To show that Algorithm 4 preserves ϵ_p-differential privacy, we first consider an alternative procedure $\mathcal{M}$. Let $\mathcal{M}$ be the procedure that releases the values $(\mathbf{f}_1, \ldots, \mathbf{f}_m, i)$ where, $\mathbf{f}_1, \ldots, \mathbf{f}_m$ are the intermediate values computed in the second step of Algorithm 4, and i is the index selected by the exponential mechanism step. We first show that $\mathcal{M}$ preserves ϵ_p-differential privacy.

Let $\mathcal{D}$ and $\mathcal{D}'$ be two datasets that differ in the value of one individual such that $\mathcal{D} = \bar{\mathcal{D}} \cup \{(\mathbf{x}, y)\}$, and $\mathcal{D}' = \bar{\mathcal{D}} \cup \{(\mathbf{x}', y')\}$.

Recall that the datasets $\mathcal{D}_1, \ldots, \mathcal{D}_{m+1}$ are disjoint; moreover, the randomness in the privacy mechanisms are independent. Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{f}_1 \in \mathcal{S}_1, \ldots, \mathbf{f}_m \in \mathcal{S}_m, i = i^* | \mathcal{D}\right) \\
&\qquad = \int_{\mathcal{S}_1 \times \ldots \mathcal{S}_m} \mathbb{P}\left(i = i * | \mathbf{f}_1, \ldots, \mathbf{f}_m, \mathcal{D}_{m+1}\right) \mu(\mathbf{f}_1, \ldots, \mathbf{f}_m | \mathcal{D}) d\mathbf{f}_1 \ldots d\mathbf{f}_m \\
&\qquad = \int_{\mathcal{S}_1 \times \ldots \mathcal{S}_m} \mathbb{P}\left(i = i * | \mathbf{f}_1, \ldots, \mathbf{f}_m, \mathcal{D}_{m+1}\right) \prod_{j=1}^{m} \mu_j(\mathbf{f}_j | \mathcal{D}_j) d\mathbf{f}_1 \ldots d\mathbf{f}_m, \qquad (124)
\end{aligned}$$

where $\mu_j(\mathbf{f})$ is the density at $\mathbf{f}$ induced by the classifier run with parameter Λ_j, and $\mu(\mathbf{f}_1, \ldots, \mathbf{f}_m)$ is the joint density at $\mathbf{f}_1, \ldots, \mathbf{f}_m$, induced by $\mathcal{M}$. Now suppose that $(\mathbf{x}, y) \in \mathcal{D}_j$, for $j = m+1$. Then, $\mathcal{D}_k = \mathcal{D}'_k$, and $\mu_j(\mathbf{f}_j | \mathcal{D}_j) = \mu_j(\mathbf{f}_j | \mathcal{D}'_j)$, for $k \in [m]$. Moreover, given any fixed set $\mathbf{f}_1, \ldots, \mathbf{f}_m$,

$$\mathbb{P}\left(i = i^* | \mathcal{D}'_{m+1}, \mathbf{f}_1, \ldots, \mathbf{f}_m\right) \leq e^{\epsilon_p} \mathbb{P}\left(i = i^* | \mathcal{D}_{m+1}, \mathbf{f}_1, \ldots, \mathbf{f}_m\right). \qquad (125)$$

Instead, if $(\mathbf{x}, y) \in \mathcal{D}_j$, for $j \in [m]$, then, $\mathcal{D}_k = \mathcal{D}'_k$, for $k \in [m+1], k \neq j$. Thus, for a fixed $\mathbf{f}_1, \ldots, \mathbf{f}_m$,

$$\mathbb{P}\left(i = i^* | \mathcal{D}'_{m+1}, \mathbf{f}_1, \ldots, \mathbf{f}_m\right) = \mathbb{P}\left(i = i^* | \mathcal{D}_{m+1}, \mathbf{f}_1, \ldots, \mathbf{f}_m\right) \tag{126}$$

$$\mu_k(\mathbf{f}_k | \mathcal{D}_k) \leq e^{\epsilon_p} \mu_k(\mathbf{f}_k | \mathcal{D}'_k). \tag{127}$$

The lemma follows by combining (124)-(127).

Now, an adversary who has access to the output of $\mathcal{M}$ can compute the output of Algorithm 4 itself, without any further access to the dataset. Therefore, by a simulatibility argument, as in Dwork et al. [2006b], Algorithm 4 also preserves ϵ_p-differential privacy. □

In the theorem above, we assume that the individual algorithms for privacy-preserving classification satisfy Definition 2; a similar theorem can also be shown when they satisfy a guarantee as in Corollary 4.

The following theorem shows that the empirical error on $\mathcal{D}_{K+1}$ of the classifier output by the tuning procedure is close to the empirical error of the best classifier in the set $\{\mathbf{f}_1, \ldots, \mathbf{f}_K\}$. The proof of this Theorem follows from Lemma 7 of McSherry and Talwar [2007].

Theorem 9. *Let $z_{\min} = \min_i z_i$, and let z be the number of mistakes made on D_{m+1} by the classifier output by our tuning procedure. Then, with probability $1 - \delta$,*

$$z \leq z_{\min} + \frac{2\log(m/\delta)}{\epsilon_p}. \tag{128}$$

Proof. In the notation of McSherry and Talwar [2007], the $z_{\min} = OPT$, the base measure μ is uniform on $[m]$, and $S_t = \{i : z_i < z_{\min} + t\}$. Their Lemma 7 shows that

$$\mathbb{P}\left(\bar{S}_{2t}\right) \leq \frac{\exp(-\epsilon_p t)}{\mu(S_t)}, \tag{129}$$

where μ is the uniform measure on $[m]$. Using $\min \mu(S_t) = \frac{1}{m}$ to upper bound the right side and setting it equal to δ we obtain

$$t = \frac{1}{\epsilon_p} \log \frac{m}{\delta}. \tag{130}$$

From this we have

$$\mathbb{P}\left(z \geq z_{min} + \frac{2}{\epsilon_p} \log \frac{m}{\delta}\right) \leq \delta, \tag{131}$$

and the result follows. □

7 Experiments

In this section we give experimental results for training linear classifiers with Algorithms 1 and 2 on two real datasets. Imposing privacy requirements necessarily degrades classifier performance. Our experiments show that provided there is sufficient data, objective perturbation (Algorithm 2) typically outperforms the sensitivity method (1) significantly, and achieves error rate close to that of the analogous non-private ERM algorithm. We first demonstrate how the accuracy of the classification algorithms vary with ϵ_p, the privacy requirement. We then show how the performance of privacy-preserving classification varies with increasing training data size.

The first dataset we consider is the `Adult` dataset from the UCI Machine Learning Repository (Asuncion and Newman [2007]). This moderately-sized dataset contains demographic information about approximately $47,000$ individuals, and the classification task is to predict whether the annual income of an individual is below or above $50,000, based on variables such as age, sex, occupation, and education. For our experiments, the average fraction of positive labels is about 0.25; therefore, a trivial classifier that always predicts -1 will achieve this error-rate, and only error-rates below 0.25 are interesting.

The second dataset we consider is the `KDDCup99` dataset (Hettich and Bay [1999]); the task here is to predict whether a network connection is a denial-of-service attack or not, based on several attributes. The dataset includes about 5,000,000 instances. For this data the average fraction of positive labels is 0.20.

In order to implement the convex minimization procedure, we use the convex optimization library provided by Okazaki [2009].

7.1 Preprocessing

In order to process the `Adult` dataset into a form amenable for classification, we removed all entries with missing values, and converted each categorial attribute to a binary vector. For example, an attribute such as `(Male,Female)` was converted into 2 binary features. Each column was normalized to ensure that the maximum value is 1, and then each row is normalized to ensure that the norm of any example is at most 1. After preprocessing, each example was represented by a 105-dimensional vector, of norm at most 1.

For the `KDDCup99` dataset, the instances were preprocessed by converting each categorial attribute to a binary vector. Each column was normalized to ensure that the maximum value is 1, and finally, each row was normalized, to ensure that the norm of any example is at most 1. After preprocessing, each example was represented by a 119-dimensional vector, of norm at most 1.

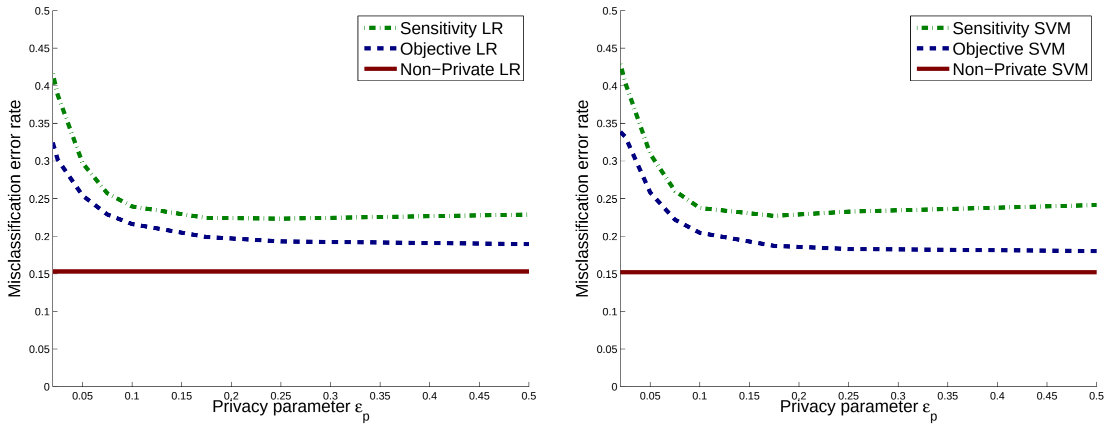
7.2 Privacy-Accuracy Tradeoff

For our first set of experiments, we study the tradeoff between the privacy requirement on the classifier, and its classification accuracy, when the classifier is trained on data of a fixed size. The privacy requirement is quantified by the value of ϵ_p; increasing ϵ_p implies a higher change in the belief of the adversary when one entry in $\mathcal{D}$ changes, and thus lower privacy. To measure accuracy, we use classification (test) error; namely, the fraction of times the classifier predicts a label with the wrong sign.

To study the privacy-accuracy tradeoff, we compare objective perturbation with the sensitivity method for logistic regression and Huber SVM. For Huber SVM, we picked the Huber constant $h = 0.5$, a typical value in Chapelle [2007][2] For each data set we trained classifiers for a few fixed values of Λ and tested the error of these classifiers. For each algorithm we chose the value of Λ that minimizes the error-rate for $\epsilon_p = 0.1$.[3] We then plotted the error-rate against ϵ_p for

[2] Chapelle [2007] recommends using h between 0.01 and 0.5; we use $h = 0.5$ as we found that a higher value typically leads to more numerical stability, as well as better performance for both privacy-preserving methods.

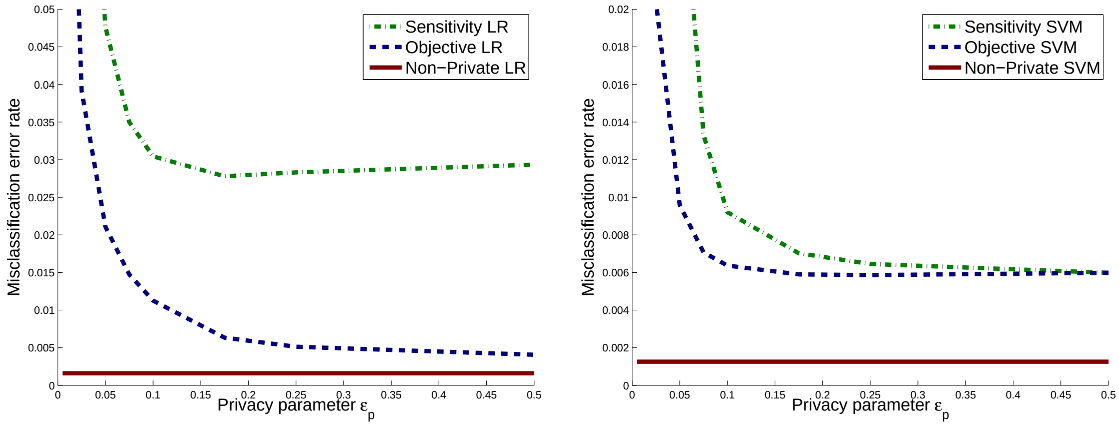
[3] For `KDDCup99` the error of the non-private algorithms did not increase with decreasing Λ.



(a) Regularized logistic regression, `Adult`

(b) Regularized SVM, `Adult`

Figure 2: Privacy-Accuracy trade-off for the `Adult` dataset



(a) Regularized logistic regression, `KDDCup99`

(b) Regularized SVM, `KDDCup99`

Figure 3: Privacy-Accuracy trade-off for the `KDDCup99` dataset

the chosen value of Λ. The results are shown in Figures 2 and 3 for both logistic regression and support vector machines[4]. The optimal values of Λ are shown in Tables 1 and 2. For non-private logistic regression and SVM, each presented error-rate is an average over 10-fold cross-validation; for the sensitivity method as well as objective perturbation, the presented error-rate is an average over 10-fold cross-validation and 50 runs of the randomized training procedure. For `Adult`, the privacy-accuracy tradeoff is computed over the entire dataset, which consists of $45,220$ examples; for `KDDCup99` we use a randomly chosen subset of $70,000$ examples.

For the `Adult` dataset, the constant classifier that classifies all examples to be negative acheives a classification error of about 0.25. The sensitivity method thus does slightly better than this constant classifier for most values of ϵ_p for both logistic regression and support vector machines. Objective perturbation outperforms sensitivity, and objective perturbation for support vector machines achieves lower classification error than objective perturbation for logistic regression. Non-private

[4] The slight kink in the SVM curve on `Adult` is due to a switch to the second phase of the algorithm.

Λ	$10^{-10.0}$	$10^{-7.0}$	$10^{-4.0}$	$10^{-3.5}$	$10^{-3.0}$	$10^{-2.5}$	$10^{-2.0}$	$10^{-1.5}$
Logistic								
Non-Private	0.1540	**0.1533**	0.1654	0.1694	0.1758	0.1895	0.2322	0.2478
Output	0.5318	0.5318	0.5175	0.4928	0.4310	0.3163	**0.2395**	0.2456
Objective	0.8619	0.8619	0.8619	0.2694	0.2369	**0.2161**	0.2305	0.2475
Huber								
Non-Private	0.1527	**0.1521**	0.1632	0.1669	0.1719	0.1793	0.2454	0.2478
Output	0.5318	0.5318	0.5211	0.5011	0.4464	0.3352	**0.2376**	0.2476
Objective	0.2582	0.2582	0.2582	0.2582	0.2559	**0.2046**	0.2319	0.2478

Table 1: Error for different regularization parameters on `Adult` for $\epsilon_p = 0.1$. The best error per algorithm is in bold.

Λ	$10^{-9.0}$	$10^{-7.0}$	$10^{-5.0}$	$10^{-3.5}$	$10^{-3.0}$	$10^{-2.5}$	$10^{-2.0}$	$10^{-1.5}$
Logistic								
Non-Private	**0.0016**	0.0016	0.0021	0.0038	0.0037	0.0037	0.0325	0.0594
Output	0.5245	0.5245	0.5093	0.3518	0.1114	0.0359	**0.0304**	0.0678
Objective	0.3170	0.3170	0.3170	0.0196	0.0118	**0.0113**	0.0285	0.0909
Huber								
Non-Private	**0.0013**	0.0013	0.0013	0.0029	0.0051	0.0056	0.0061	0.0163
Output	0.5245	0.5245	0.5229	0.4611	0.3353	0.0590	**0.0092**	0.0179
Objective	0.1207	0.1207	0.1207	0.2622	0.0478	0.0086	**0.0064**	0.0157

Table 2: Error for different regularization parameters on `KDDCup99` for $\epsilon_p = 0.1$. The best error per algorithm is in bold.

logistic regression and support vector machines both have classification error about 0.15.

For the KDDCup99 dataset, the constant classifier that classifies all examples as negative, has error 0.19. Again, objective perturbation outperforms sensitivity for both logistic regression and support vector machines; however, for SVM and high values of ϵ_p (low privacy), the sensitivity method performs almost as well as objective perturbation. In the low privacy regime, logistic regression under objective perturbation is better than support vector machines. In contrast, in the high privacy regime (low ϵ_p), support vector machines with objective perturbation outperform logistic regression. For this dataset, non-private logistic regression and support vector machines both have a classification error of about 0.001.

For SVMs on both Adult and KDDCup99, for large ϵ_p (0.25 onwards), the error of either of the private methods can increase slightly with increasing ϵ_p. This seems counterintuitive, but appears to be due the imbalance in fraction of the two labels. As the labels are imbalanced, the optimal classifier is trained to perform better on the negative labels than the positives. As ϵ_p increases, for a fixed training data size, so does the perturbation from the optimal classifier, induced by either of the private methods. Thus, as the perturbation increases, the number of false positives increases, whereas the number of false negatives decreases (as we verified by measuring the average false positive and false negative rates of the private classifiers). Therefore, the total error may increase slightly with decreasing privacy.

7.3 Accuracy vs. Training Data Size Tradeoffs

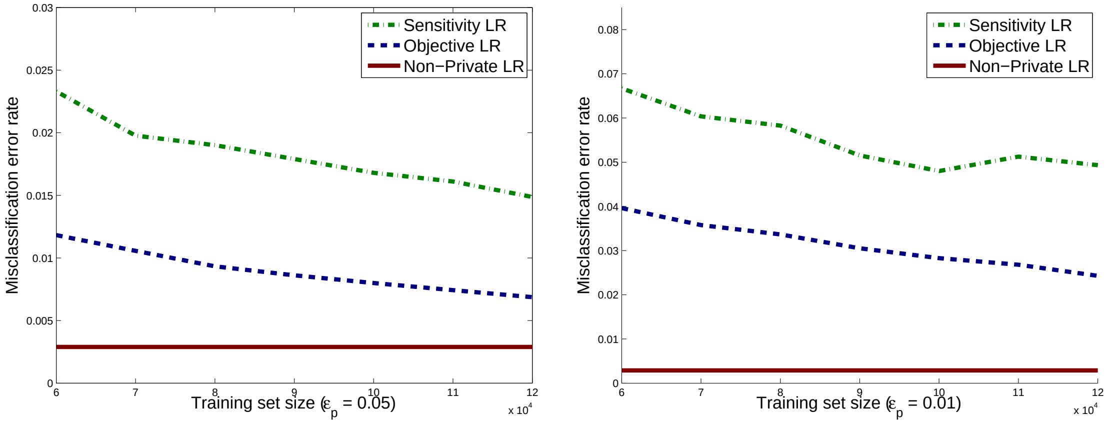
Next we examine how classification accuracy varies as we increase the size of the training set. We measure classification accuracy as the accuracy of the classifier produced by the tuning procedure in Section 6. As the Adult dataset is not sufficiently large to allow us to do privacy-preserving tuning, for these experiments, we restrict our attention to the KDDCup99 dataset.

Figures 4 and 5 present the learning curves for objective perturbation, non-private ERM and the sensitivity method for logistic loss and Huber loss, respectively. Experiments are shown for $\epsilon_p = 0.01$ and $\epsilon_p = 0.05$ for both loss functions. The training sets (for each of 5 values of Λ) are chosen to be of size $n = 60,000$ to $n = 120,000$, and the validation and test sets each are of size $25,000$. Each presented value is an average over 5 random permutations of the data, and 50 runs of the randomized classification procedure. For objective perturbation we performed experiment in the regime when $\epsilon'_p > 0$, so $\Delta = 0$ in Algorithm 2.[5]

For non-private ERM, we present result for training sets from $n = 300,000$ to $n = 600,000$. The non-private algorithms are tuned by comparing 5 values of Λ on the same training set, and the test set is of size $25,000$. Each reported value is an average over 5 random permutations of the data.

We see from the figures that for non-private logistic regression and support vector machines, the error remains constant with increasing data size. For the private methods, the error usually decreases as the data size increases. In all cases, objective perturbation outperforms the sensitivity method, and support vector machines generally outperform logistic regression.

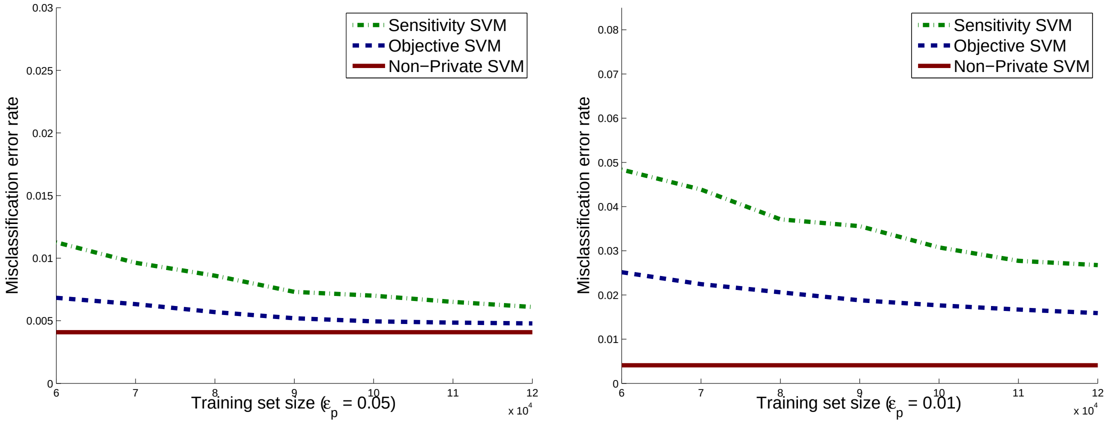
[5] This was chosen for a fair comparison with non-private as well as the output perturbation method, both of which had access to only 5 values of Λ.



(a) $\epsilon_p = 0.05$

(b) $\epsilon_p = 0.01$

Figure 4: Learning curves for logistic regression on the `KDDCup99` dataset



(a) $\epsilon_p = 0.05$

(b) $\epsilon_p = 0.01$

Figure 5: Learning curves for SVM on the `KDDCup99` dataset

8 Conclusions and Open Problems

In this paper we study the problem of learning classifiers with regularized empirical risk minimization in a privacy-preserving manner. We consider privacy in the ϵ_p-differential privacy model of Dwork et al. [2006b] and provide two algorithms for privacy-preserving ERM. The first one is based on the sensitivity method due to Dwork et al. [2006b], in which the output of the non-private algorithm is perturbed by adding noise. We introduce a second algorithm based on the paradigm of objective perturbation. We provide bounds on the sample requirement of these algorithms for achieving generalization error ϵ_g. We show how to apply these algorithms with kernels, and finally, we provide experiments with both algorithms on two real datasets. Our work is, to our knowledge, the first to propose computationally efficient classification algorithms satisfying differential privacy, together with validation on standard data sets.

In general, for classification, the error rate increases as the privacy requirements are made more stringent. Our generalization guarantees formalize this “price of privacy.” Our experiments, as well as theoretical results, indicate that objective perturbation usually outperforms the sensitivity methods at managing the tradeoff between privacy and learning performance. Both algorithms perform better with more training data, and when abundant training data is available, the performance of both algorithms can be close to non-private classification. One area for improvement upon our work is to reduce the amount of data needed for near-optimal performance.

An open question in this work is to extend objective perturbation methods to more general convex optimization problems. Currently, the objective perturbation method applies to strongly convex regularization functions and differentiable losses. Convex optimization problems appear in many contexts within and without machine learning: density estimation, resource allocation for communication systems and networking, social welfare optimization in economics, and others. In some cases these algorithms will also operate on sensitive or private data. Extending the ideas and analysis here to those settings would provide a rigorous foundation for privacy analysis.

A second open question is to find a better solution for privacy-preserving classification with kernels. Our current method is based on a reduction to the linear case, using the algorithm of Rahimi and Recht [2008b]; however, this method can be statistically inefficient, and require a lot of training data, particularly when coupled with our privacy mechanism. The reason is that the algorithm of Rahimi and Recht [2008b] requires the dimension D of the projected space to be very high for good performance. However, most differentially-private algorithms perform worse as the dimensionality of the data grows. Is there a better linearization method, which is possibly data-dependent, that will provide a more statistically efficient solution to privacy-preserving learning with kernels?

A final question is to provide better upper and lower bounds on the sample requirement of privacy-preserving linear classification. Some progress in this direction has been made on this by Chaudhuri and Hsu [2010], who provide some lower bounds on the sample complexity of differentially private learning, as well as a computationally inefficient algorithm, which has a better sample requirement than the results in this paper. However, the main open question is to provide a computationally efficient algorithm for linear classification which has better statistical efficiency.

Privacy-preserving machine learning is the endeavor of designing private analogues of widely used machine learning algorithms. We believe the present study is a starting point for further study of the differential privacy model in this relatively new subfield of machine learning. The work of Dwork et al. [2006b] set up a framework for assessing the privacy risks associated with publishing the results of data analyses. Demanding high privacy requires sacrificing utility, which in the context of classification and prediction is excess loss or regret. In this paper we demonstrate the privacy-utility tradeoff for ERM, which is but one corner of the machine learning world. Applying these privacy concepts to other machine learning problems will lead to new and interesting tradeoffs and towards a set of tools for practical privacy-preserving learning and inference. We hope that our work provides a benchmark of the current price of privacy, and inspires improvements in future work.

Acknowledgements

The authors would like to thank Sanjoy Dasgupta and Daniel Hsu for several pointers, and to acknowledge Adam Smith, who helped point out some important technical aspects in the previous version of the paper. The work of K. Chaudhuri and A.D. Sarwate was supported in part by the

California Institute for Telecommunications and Information Technologies (CALIT2) at UC San Diego. K. Chaudhuri was also supported by National Science Foundation IIS-0713540. Part of this work was done while C. Monteleoni was at UC San Diego, with support from National Science Foundation IIS-0713540. The experimental results were made possibly by support from the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622.

References

R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, 2000.

A. Asuncion and D.J. Newman. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *Proc. 16th Intl. World Wide Web Conference*, 2007.

K. Ball. An elementary introduction to modern convex geometry. In S. Levy, editor, *Flavors of Geometry*, volume 31 of *Mathematical Sciences Research Institute Publications*, pages 1–58. Cambridge University Press, 1997.

B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *PODS*, pages 273–282, 2007.

Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In *TCC*, pages 437–454, 2010.

P Billingsley. *Probability and measure.* A Wiley-Interscience publication. Wiley, New York [u.a.], 3. ed edition, 1995. ISBN 0471007102.

A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In R. E. Ladner and C. Dwork, editors, *STOC*, pages 609–618. ACM, 2008. ISBN 978-1-60558-047-0.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

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

K. Chaudhuri and D. Hsu. Sample complexity bounds for differentially private classification. Manuscript, 2010.

K. Chaudhuri and N. Mishra. When random sampling preserves privacy. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 198–213. Springer, 2006. ISBN 3-540-37432-9.

K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *NIPS*, 2008.

C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. ISBN 3-540-35907-9.

C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC*, 2009.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006a.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *3rd IACR Theory of Cryptography Conference,*, pages 265–284, 2006b.

A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003.

S.R. Ganta, S.P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, pages 265–273, 2008.

A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private approximation algorithms. In *SODA*, 2010.

S. Hettich and S.D. Bay. The UCI KDD Archive. University of California, Irvine, Department of Information and Computer Science, 1999. URL `http://kdd.ics.uci.edu`.

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genet*, 4(8):e1000167, 08 2008.

Ilse C. F. Ipsen and Rizwana Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776, 2008.

Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. ”i know what you did last summer”: query logs and user privacy. In *CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-803-9.

S.P. Kasivishwanathan, M. Rudelson, A. Smith, and J. Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *STOC*, 2010.

S. A. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proc. of FOCS*, 2008.

G.S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.

S. Laur, H. Lipmaa, and T. Mielikäinen. Cryptographically private support vector machines. In *KDD*, pages 618–624, 2006.

A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proc. of ICDE*, 2006.

A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, pages 277–286, 2008.

O. L. Mangasarian, E. W. Wild, and G. Fung. Privacy-preserving classification of vertically partitioned data via random kernels. *TKDD*, 2(3), 2008.

F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets (how to break anonymity of the netflix prize dataset). In *Proc. of 29th IEEE Symposium on Security and Privacy*, pages 111–125, 2008.

K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 75–84. ACM, 2007. ISBN 978-1-59593-631-8.

N. Okazaki. liblbfgs: a library of limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). http://www.chokkan.org/software/liblbfgs/index.html, 2009.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *Proceedings of the 46th Allerton Conference on Communication, Control, and Computing*, 2008a.

A. Rahimi and B. Recht. Weighted sums of random kitchen sinks : Replacing minimization with randomization in learning. In *NIPS*, 2008b.

R.T. Rockafellar and R J-B. Wets. *Variational Analysis*. Springer, Berlin, 1998.

B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. In *http://arxiv.org/abs/0911.5708*, 2009.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, July 2007.

S. Shalev-Shwartz and N. Srebro. SVM optimization : Inverse dependence on training set size. In *ICML*, 2008.

K. Sridharan, N. Srebro, and S. Shalev-Shwartz. Fast rates for regularized objectives. In *NIPS*, 2008.

L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. on Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

L. Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine and Ethics*, 25:98–110, 1997.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiao yong Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In *ACM Conference on Computer and Communications Security*, pages 534–544, 2009.

L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164, 1982.

J. Z. Zhan and S. Matwin. Privacy-preserving support vector machine classification. *IJIIDS*, 1 (3/4):356–385, 2007.

S. Zhou, K. Ligett, and L. Wasserman. Differential privacy with compression. In *Proceedings of the 2009 International Symposium on Information Theory*, Seoul, South Korea, 2009a.

S. Zhou, K. Ligett, and L. Wasserman. Differential privacy with compression. In *IEEE International Symposium of Information Theory*, 2009b.