2014 IEEE Annual Symposium on Foundations of Computer Science

Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds

Raef Bassily, Adam Smith
Computer Science and Engineering Department
The Pennsylvania State University
Email: {bassily, asmith}@psu.edu

Abhradeep Thakurta
Yahoo! Labs, Stanford University and Microsoft Research
Email: abhradeep@yahoo-inc.com

***Abstract*—Convex empirical risk minimization is a basic tool in machine learning and statistics. We provide new algorithms and matching lower bounds for differentially private convex empirical risk minimization assuming only that each data point's contribution to the loss function is Lipschitz and that the domain of optimization is bounded. We provide a separate set of algorithms and matching lower bounds for the setting in which the loss functions are known to also be strongly convex.**

Our algorithms run in polynomial time, and in some cases even match the optimal nonprivate running time (as measured by oracle complexity). We give separate algorithms (and lower bounds) for $(\epsilon, 0)$- and (ϵ, δ)-differential privacy; perhaps surprisingly, the techniques used for designing optimal algorithms in the two cases are completely different.

Our lower bounds apply even to very simple, smooth function families, such as linear and quadratic functions. This implies that algorithms from previous work can be used to obtain optimal error rates, under the additional assumption that the contributions of each data point to the loss function is *smooth*. We show that simple approaches to smoothing arbitrary loss functions (in order to apply previous techniques) do not yield optimal error rates. In particular, optimal algorithms were not previously known for problems such as training support vector machines and the high-dimensional median.

I. Introduction

Convex optimization is one of the most basic and powerful computational tools in statistics and machine learning. It is most commonly used for empirical risk minimization (ERM): the data set $\mathcal{D} = \{d_1, ..., d_n\}$ defines a convex loss function $\mathcal{L}(\cdot)$ which is minimized over a convex set $\mathcal{C}$. When run on sensitive data, however, the results of convex ERM can leak sensitive information. For example, medians and support vector machine parameters can, in many cases, leak entire records in the clear (see "Motivation", below).

In this paper, we provide new algorithms and matching lower bounds for *differentially private* convex ERM assuming only that each data point's contribution to the loss function is Lipschitz and that the domain of optimization is bounded. This builds on a line of work started by Chaudhuri et al. [11].

Problem formulation. Given a data set $\mathcal{D} = \{d_1, ..., d_n\}$ drawn from a universe $\mathcal{X}$, and a closed, convex set $\mathcal{C}$, our goal is to

$$\text{minimize } \mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i) \text{ over } \theta \in \mathcal{C}$$

The map ℓ defines, for each data point d, a loss function $\ell(\cdot; d)$ on $\mathcal{C}$. We will generally assume that $\ell(\cdot; d)$ is convex and L-Lipschitz for all $d \in \mathcal{X}$. One obtains variants on this basic problem by assuming additional restrictions, such as (i) that $\ell(\cdot; d)$ is Δ-strongly convex for all $d \in \mathcal{X}$, and/or (ii) that $\ell(\cdot; d)$ is β-smooth for all $d \in \mathcal{X}$. Definitions of Lipschitz, strong convexity and smoothness are provided at the end of the introduction.

For example, given a collection of data points in $\mathbb{R}^p$, the Euclidean 1-median is a point in $\mathbb{R}^p$ that minimizes the sum of the Euclidean distances to the data points. That is, $\ell(\theta; d_i) = \|\theta - d_i\|_2$, which is 1-Lipschitz in θ for any choice of d_i. Another common example is the support vector machine (SVM): given a data point $d_i = (x_i, y_i) \in \mathbb{R}^p \times \{-1, 1\}$, one defines a loss function $\ell(\theta; d_i) = hinge(y_i \cdot \langle \theta, x_i \rangle)$, where $hinge(z) = (1 - z)_+$ (here $(1 - z)_+$ equals $1 - z$ for $z \leq 1$ and 0, otherwise). The loss is L-Lipshitz in θ when $\|x_i\|_2 \leq L$.

Our formulation also captures *regularized* ERM, in which an additional (convex) function $r(\theta)$ is added to the loss function to penalize certain types of solutions; the loss function is then $r(\theta) + \sum_{i=1}^{n} \ell(\theta; d_i)$. One can fold the regularizer $r(\cdot)$ into the data-dependent functions by replacing $\ell(\theta; d_i)$ with $\tilde{\ell}(\theta; d_i) = \ell(\theta; d_i) + \frac{1}{n} r(\theta)$, so that $\mathcal{L}(\theta; \mathcal{D}) = \sum_i \tilde{\ell}(\theta; d_i)$. This folding comes at some loss of generality (since it may increase the Lipschitz constant), but it does not affect asymptotic results. Note that if r is Δn-strongly convex, then every $\tilde{\ell}$ is Δ-strongly convex.

We measure the success of our algorithms by the worst-case (over inputs) expected *excess empirical risk*, namely

$$\mathbb{E}(\mathcal{L}(\hat{\theta}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})), \tag{1}$$

where $\hat{\theta}$ is the output of the algorithm, $\theta^* = \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; \mathcal{D})$ is the true minimizer, and the expectation is only over the coins of the algorithm. Expected risk guarantees can be converted to high-probability guarantees using standard techniques (see full version [3]).

Another important measure of performance is an algorithm's (excess) generalization error, where loss is measured with respect to the average over an unknown distribution from which the data are assumed to be drawn i.i.d.. Our upper bounds on empirical risk imply upper bounds on generalization error (via uniform convergence and similar ideas); the resulting bounds are only known to be tight in certain ranges of

0272-5428/14 $31.00 © 2014 IEEE
DOI 10.1109/FOCS.2014.56

parameters, however. Detailed statements may be found in full version [3]. This proceedings version discusses only empirical error.

Motivation. Convex ERM is used for fitting models from simple least-squares regression to support vector machines, and their use may have significant implications to privacy. As a simple example, note that the Euclidean 1-median of a data set will typically be an actual data point, since the gradient of the loss function has discontinuities at each of the d_i. (Thinking about the one-dimensional median, where there is *always* a data point that minimizes the loss, is helpful.) Thus, releasing the median may well reveal one of the data points in the clear. A more subtle example is the support vector machine (SVM). The solution to an SVM program is often presented in its dual form, whose coefficients typically consist of a set of $p+1$ exact data points. [26] show how the results of many convex ERM problems can be combined to carry out reconstruction attacks in the spirit of [13].

Differential privacy is a rigorous notion of privacy that emerged from a line of work in theoretical computer science and cryptography [18, 7, 17]. We say two data sets $\mathcal{D}$ and $\mathcal{D}'$ of size n are neighbors if they differ in one entry (that is, $|\mathcal{D} \triangle \mathcal{D}'| = 2$). A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private ([17, 16]) if, for all neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$ and for all events S in the output space of $\mathcal{A}$, we have

$$\Pr(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\epsilon} \Pr(\mathcal{A}(\mathcal{D}') \in S) + \delta\,.$$

Algorithms that satisfy differential privacy for $\epsilon < 1$ and $\delta \ll 1/n$ provide meaningful privacy guarantees, even in the presence of side information. In particular, they avoid the problems mentioned in "Motivation" above. See [15, 27, 28] for discussion of the "semantics" of differential privacy.

Setting Parameters. We will aim to quantify the role of several basic parameters on the excess risk of differentially private algorithms: the size of the data set n, the dimension p of the parameter space $\mathcal{C}$, the Lipschitz constant L of the loss functions, the diameter $\|\mathcal{C}\|_2$ of the constraint set and, when applicable, the strong convexity Δ.

We may take L and $\|\mathcal{C}\|_2$ to be 1 without loss of generality: We can set $\|\mathcal{C}\|_2 = 1$ by rescaling θ (replacing by θ with $\theta \cdot \|\mathcal{C}\|_2$); we can then set $L = 1$ by rescaling the loss function $\mathcal{L}$ (replacing $\mathcal{L}$ by $\mathcal{L}/L$). These two transformations change the excess risk by $L\|C\|_2$. The parameter Δ cannot similarly be rescaled while keeping L and $\|\mathcal{C}\|_2$ the same. However, we always have $\Delta \leq 2L/\|\mathcal{C}\|_2$.

In the sequel, we thus focus on the setting where $L = \|\mathcal{C}\|_2 = 1$ and $\Delta \in [0, 2]$. To convert excess risk bounds for $L = \|\mathcal{C}\|_2 = 1$ to the general setting, one can multiply the risk bounds by $L\|\mathcal{C}\|_2$, and replace Δ by $\frac{\Delta}{L\|\mathcal{C}\|_2}$.

A. Contributions

We give algorithms that significantly improve on the state of the art for optimizing non-smooth loss functions — for both the general case and strongly convex functions, we improve the excess risk bounds by a factor of $\sqrt{n}$, asymptotically. The algorithms we give for $(\epsilon, 0)$- and (ϵ, δ)-differential privacy work on very different principles. We group the algorithms below by technique: gradient descent, exponential sampling, and localization.

For the purposes of this section, $\tilde{O}(\cdot)$ notation hides factors polynomial in $\log n$ and $\log(1/\delta)$. Detailed bounds are stated in Table I.

Gradient descent-based algorithms. For (ϵ, δ)-differential privacy, we show that a noisy version of gradient descent achieves excess risk $\tilde{O}(\sqrt{p}/\epsilon)$. This matches our lower bound, $\Omega(\min(n, \sqrt{p}/\epsilon))$, up to logarithmic factors. (Note that every $\theta \in \mathcal{C}$ has excess risk at most n, so a lower bound of n can always be matched.) For Δ-strongly convex functions, a variant of our algorithm has risk $\tilde{O}(\frac{p}{\Delta n \epsilon^2})$, which matches the lower bound $\Omega(\frac{p}{n\epsilon^2})$ when Δ is bounded below by a constant (recall that $\Delta \leq 2$ since $L = \|\mathcal{C}\|_2 = 1$).

Previously, the best known risk bounds were $\Omega(\sqrt{pn}/\epsilon)$ for general convex functions and $\Omega(\frac{p}{\sqrt{n\Delta\epsilon^2}})$ for Δ-strongly convex functions (achievable via several different techniques ([11, 29, 21, 14])). Under the restriction that each data point's contribution to the loss function is sufficiently smooth, objective perturbation [11, 29] also has risk $\tilde{O}(\sqrt{p}/\epsilon)$ (which is tight, since the lower bounds apply to smooth functions). However, smooth functions do not include important special cases such as medians and support vector machines. [11] suggest applying their technique to support vector machines by smoothing ("huberizing") the loss function. We show in the full version [3]that this approach still yields expected excess risk $\Omega(\sqrt{pn}/\epsilon)$.

Although straightforward noisy gradient descent would work well in our setting, we present a faster variant based on *stochastic* gradient descent: At each step t, the algorithm samples a random point d_i from the data set, computes a noisy version of d_i's contribution to the gradient of $\mathcal{L}$ at the current estimate $\tilde{\theta}_t$, and then uses that noisy measurement to update the parameter estimate. The algorithm is similar to algorithms that have appeared previously ([41] first investigated gradient descent with noisy updates; stochastic variants were studied by [21, 14, 39]). The novelty of our analysis lies in taking advantage of the randomness in the choice of d_i (following [25]) to run the algorithm for many steps without a significant cost to privacy. Running the algorithm for $T = n^2$ steps, gives the desired expected excess risk bound. Even nonprivate first-order algorithms—i.e., those based on gradient measurements—must learn information about the gradient at $\Omega(n^2)$ points to get risk bounds that are independent of n (this follows from "oracle complexity" bounds showing that $1/\sqrt{T}$ convergence rate is optimal [32, 1]).

The gradient descent approach does not, to our knowledge, allow one to get optimal excess risk bounds for $(\epsilon, 0)$-differential privacy. The main obstacle is that "strong composition" of (ϵ, δ)-privacy [19] appears necessary to allow a first-order method to run for sufficiently many steps.

Exponential Sampling-based Algorithms. For $(\epsilon, 0)$-differential privacy, we observe that a straightforward

	$(\epsilon, 0)$-DP			(ϵ, δ)-DP		
	Previous [11]	This work		Previous [29]	This work	
Assumptions	Upper Bd	Upper Bd	Lower Bd	Upper Bd	Upper Bd	Lower Bd
1-Lipschitz and $\|C\|_2 = 1$	$\frac{p\sqrt{n}}{\epsilon}$	$\frac{p}{\epsilon}$	$\frac{p}{\epsilon}$	$\frac{\sqrt{p \cdot n}\log(1/\delta)}{\epsilon}$	$\frac{\sqrt{p}\log^2(n/\delta)}{\epsilon}$	$\frac{\sqrt{p}}{\epsilon}$
... and $O(p)$-smooth	$\frac{p}{\epsilon}$		$\frac{p}{\epsilon}$	$\frac{\sqrt{p}\log(1/\delta)}{\epsilon}$		$\frac{\sqrt{p}}{\epsilon}$
1-Lipschitz and Δ-strongly convex and $\|C\|_2 = 1$ (implies $\Delta \leq 2$)	$\frac{p^2}{\sqrt{n}\Delta\epsilon^2}$	$\frac{\log(n)}{\Delta} \cdot \frac{p^2}{n\epsilon^2}$	$\frac{p^2}{n\epsilon^2}$	$\frac{p\log(1/\delta)}{\sqrt{n}\Delta\epsilon^2}$	$\frac{\log^3(n/\delta)}{\Delta} \cdot \frac{p}{n\epsilon^2}$	$\frac{p}{n\epsilon^2}$
... and $O(p)$-smooth	$\frac{p^2}{n\Delta\epsilon^2}$		$\frac{p^2}{n\epsilon^2}$	$\frac{p\log(1/\delta)}{n\Delta\epsilon^2}$		$\frac{p}{n\epsilon^2}$

TABLE I

UPPER AND LOWER BOUNDS FOR EXCESS RISK OF DIFFERENTIALLY-PRIVATE CONVEX ERM. BOUNDS IGNORE LEADING MULTIPLICATIVE CONSTANTS, AND THE VALUES IN THE TABLE GIVE THE BOUND WHEN IT IS BELOW n. THAT IS, UPPER BOUNDS SHOULD BE READ AS $O(\min(n, ...))$ AND LOWER BOUNDS, AS $\Omega(\min(n, ...))$. HERE $\|\mathcal{C}\|_2$ IS THE DIAMETER OF $\mathcal{C}$. THE BOUNDS ARE STATED FOR THE SETTING WHERE $L = \|\mathcal{C}\|_2 = 1$, WHICH CAN BE ENFORCED BY RESCALING; TO GET GENERAL STATEMENTS, MULTIPLY THE RISK BOUNDS BY $L\|\mathcal{C}\|_2$, AND REPLACE Δ BY $\frac{\Delta\|\mathcal{C}\|_2}{L}$. WE ASSUME $\delta < 1/n$ TO SIMPLIFY THE BOUNDS.

use of the exponential mechanism — sampling from an appropriately-sized net of points in $\mathcal{C}$, where each point θ has probability proportional to $\exp(-\epsilon\mathcal{L}(\theta; \mathcal{D}))$ — has excess risk $\tilde{O}(p/\epsilon)$ on general Lipschitz functions, nearly matching the lower bound of $\Omega(p/\epsilon)$. (The bound would not be optimal for (ϵ, δ)-privacy because it scales as p, not $\sqrt{p}$). This mechanism is inefficient in general since it requires construction of a net and an appropriate sampling mechanism.

We give a polynomial time algorithm that achieves the optimal excess risk, namely $O(p/\epsilon)$. Note that the achieved excess risk does not have any logarithmic factors which is shown to be the case using a "peeling-"type argument that is specific to convex functions. The idea of our algorithm is to sample efficiently from the continuous distribution on all points in $\mathcal{C}$ with density $\mathcal{P}(\theta) \propto e^{-\epsilon\mathcal{L}(\theta)}$. Although the distribution we hope to sample from is log-concave, standard techniques do not work for our purposes: existing methods converge only in statistical difference, whereas we require a *multiplicative* convergence guarantee to provide $(\epsilon, 0)$-differential privacy. Previous solutions to this issue ([20]) worked for the uniform distribution, but not for log-concave distributions.

The problem comes from the combination of an arbitrary convex set and an arbitrary (Lipschitz) loss function defining $\mathcal{P}$. We circumvent this issue by giving an algorithm that samples from an appropriately defined distribution $\tilde{\mathcal{P}}$ on a cube containing $\mathcal{C}$, such that $\tilde{\mathcal{P}}$ (i) outputs a point in $\mathcal{C}$ with constant probability, and (ii) conditioned on sampling from $\mathcal{C}$, is within multiplicative distance $O(\epsilon)$ from the correct distribution. We use, as a subroutine, the random walk on grid points of the cube of [2].

Localization: Optimal Algorithms for Strongly Convex Functions. The exponential-sampling-based technique discussed above does not take advantage of strong convexity of the loss function. We show, however, that a novel combination of two standard techniques—the exponential mechanism and Laplace-noise-based output perturbation—does yield an optimal algorithm. [11] and [34] showed that strongly convex functions have low-sensitivity minimizers, and hence that one can release the minimum of a strongly convex function with Laplace noise (with total Euclidean length about $\rho = \frac{p}{\Delta\epsilon n}$ if each loss function is Δ-strongly convex). Simply using this first estimate as a candidate output does not yield optimal utility in general; instead it gives a risk bound of roughly $\frac{p}{\Delta\epsilon}$.

The main insight is that this first estimate defines us a small neighborhood $\mathcal{C}_0 \subseteq \mathcal{C}$, of radius about ρ, that contains the true minimizer. Running the exponential mechanism in this small set improves the excess risk bound by a factor of about ρ over running the same mechanism on all of $\mathcal{C}$. The final risk bound is then $\tilde{O}(\rho\frac{p}{\epsilon n}) = \tilde{O}(\frac{p^2}{\Delta\epsilon^2 n})$, which matches the lower bound of $\Omega(\frac{p^2}{\epsilon^2 n})$ when $\Delta = \Omega(1)$. This simple "localization" idea is not needed for (ϵ, δ)-privacy, since the gradient descent method can already take advantage of strong convexity to converge more quickly.

Lower bounds. We use techniques developed to bound the accuracy of releasing 1-way marginals (due to [20] for $(\epsilon, 0)-$ and [9] for (ϵ, δ)-privacy) to show that our algorithms have essentially optimal risk bounds. The instances that arise in our lower bounds are simple: the functions can be linear (or quadratic, for the case of strong convexity) and the constraint set $\mathcal{C}$ can be either the unit ball or the hypercube. In particular, our lower bounds apply to special case of smooth functions, demonstrating the optimality of objective perturbation [11, 29] in that setting. The reduction to lower-bounds for 1-way marginals is not quite black-box; we exploit specific properties of the instances used by [20, 9].

Finally, we provide a much stronger lower bound on the utility of a specific algorithm, the Huberization-based algorithm proposed by [11] for support vector machines. In order to apply their algorithm to nonsmooth loss functions, they proposed smoothing the loss function by Huberization, and then running their algorithm (which requires smoothness for the privacy analysis) on the resulting, modified loss functions.

We show that for any setting of the Huerization parameters, there are simple, one-dimensional nonsmooth loss functions for which the algorithm has error $\Omega(n)$. This bound justifies the effort we put into designing new algorithms for nonsmooth loss functions.

B. Other Related Work

In addition to the previous work mentioned above, we mention several closely related works. A rich line of work seeks to characterize the optimal error of differentially private algorithms for learning and optimization [25, 4, 10, 5, 6]. In particular, our results on $(\epsilon, 0)$-differential privacy imply nearly-tight bounds on the "representation dimension" [6] of convex Lipschitz functions.

[23] gave dimension-independent expected excess risk bounds for the special case of "generalized linear models" with a strongly convex regularizer, assuming that $\mathcal{C} = \mathbb{R}^p$ (that is, unconstrained optimization). [29, 37] considered parameter convergence for high-dimensional sparse regression (where $p \gg n$). Efficient implementations of the exponential mechanism over infinite domains were discussed by [20], [12] and [24]. The latter two works were specific to sampling (approximately) singular vectors of a matrix, and their techniques do not obviously apply here.

Differentially private convex learning in different models has also been studied: for example, [21, 14, 38] study online optimization, [22] study an interactive model tailored to high-dimensional kernel learning.

C. Additional Definitions

For completeness, we state a few additional definitions related to convex sets and functions.

- $\ell : \mathcal{C} \to \mathbb{R}$ is L-Lipschitz (in the Euclidean norm) if, for all pairs $x, y \in \mathcal{C}$, we have $|\ell(x) - \ell(y)| \leq L\|x - y\|_2$. A subgradient of a convex ℓ function at x, denoted $\partial\ell(x)$, is the set of vectors z such that for all $y \in \mathcal{C}$, $\ell(y) \geq \ell(x) + \langle z, y - x\rangle$.
- ℓ is Δ-*strongly convex* on $\mathcal{C}$ if, for all $x \in \mathcal{C}$, for all subgradients z at x, and for all $y \in \mathcal{C}$, we have $\ell(y) \geq \ell(x) + \langle z, y - x\rangle + \frac{\Delta}{2}\|y - x\|_2^2$ (i.e., ℓ is bounded *below* by a quadratic function tangent at x).
- ℓ is β-*smooth* on $\mathcal{C}$ if, for all $x \in \mathcal{C}$, for all subgradients z at x and for all $y \in \mathcal{C}$, we have $\ell(y) \leq \ell(x) + \langle z, y - x\rangle + \frac{\beta}{2}\|y - x\|_2^2$ (i.e., ℓ is bounded *above* by a quadratic function tangent at x). Smoothness implies differentiability, so the subgradient at x is unique.
- Given a convex set $\mathcal{C}$, we denote its diameter by $\|\mathcal{C}\|_2$. We denote the projection of any vector $\theta \in \mathbb{R}^p$ to the convex set $\mathcal{C}$ by $\Pi_{\mathcal{C}}(\theta) = \arg\min_{x \in \mathcal{C}} \|\theta - x\|_2$.

II. Gradient Descent and Optimal (ϵ, δ)-differentially private Optimization

In this section we provide an algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ (Algorithm 1) for computing θ^{priv} using a *noisy stochastic variant* of the classic gradient descent algorithm from the optimization literature [8]. Our algorithm (and the utility analysis) was inspired by the approach of [41] for logistic regression.

All the excess risk bounds (1) in this section and the rest of this paper, are presented in expectation over the randomness of the algorithm. In the full version [3]we provide a generic tool to translate the expectation bounds into high probability bound albeit at a loss of extra logarithmic factor in the inverse of the failure probability.

Note(1): The results in this section do *not* require the loss function ℓ to be differentiable. Although we present Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ (and its analysis) using the gradient of the loss function $\ell(\theta; d)$ at θ, the same guarantees hold if instead of the gradient, the algorithm is run with any sub-gradient of ℓ at θ.

Note(2): Instead of using the stochastic variant in Algorithm 1, one can use the complete gradient (i.e., $\bigtriangledown\mathcal{L}(\theta; \mathcal{D})$) in Step 5 and still have the same utility guarantee as Theorem II.4. However, the running time goes up by a factor of n.

Algorithm 1 $\mathcal{A}_{\mathsf{Noise-GD}}$: Differentially Private Gradient Descent

Input: Data set: $\mathcal{D} = \{d_1, \cdots, d_n\}$, loss function ℓ (with Lipschitz constant L), privacy parameters (ϵ, δ), convex set $\mathcal{C}$, and the learning rate function $\eta : [n^2] \to \mathbb{R}$.

1: Set noise variance $\sigma^2 \leftarrow \frac{32L^2n^2\log(n/\delta)\log(1/\delta)}{\epsilon^2}$.
2: $\widetilde{\theta}_1$: Choose any point from $\mathcal{C}$.
3: **for** $t = 1$ to $n^2 - 1$ **do**
4: Pick $d \sim_u \mathcal{D}$ with replacement.
5: $\widetilde{\theta}_{t+1} = \Pi_{\mathcal{C}}\left(\widetilde{\theta}_t - \eta(t)\left[n \bigtriangledown \ell(\widetilde{\theta}_t; d) + b_t\right]\right)$, where $b_t \sim \mathcal{N}\left(0, \mathbb{I}_p\sigma^2\right)$.
6: Output $\theta^{priv} = \widetilde{\theta}_{n^2}$.

Theorem II.1 (Privacy guarantee)**.** *Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ (Algorithm 1) is (ϵ, δ)-differentially private.*

Proof: At any time step $t \in [n^2]$ in Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$, fix the randomness due to sampling in Line 4. Let $X_t(\mathcal{D}) = n \bigtriangledown \ell(\widetilde{\theta}_t; d) + b_t$ be a random variable defined over the randomness of b_t and conditioned on $\widetilde{\theta}_t$ (see Line 5 for a definition), where $d \in \mathcal{D}$ is the data point picked in Line 4. Denote $\mu_{X_t(\mathcal{D})}(y)$ to be the measure of the random variable $X_t(\mathcal{D})$ induced on $y \in \mathbb{R}$. For any two neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$, define the *privacy loss* random variable [19] to be $W_t = \left|\log \frac{\mu_{X_t(\mathcal{D})}(X_t(\mathcal{D}))}{\mu_{X_t(\mathcal{D}')}(X_t(\mathcal{D}))}\right|$. Standard differential privacy arguments for Gaussian noise addition (see [29, 33]) will ensure that with probability $1 - \frac{\delta}{2}$ (over the randomness of the random variables b_t's and conditioned on the randomness due to sampling), $W_t \leq \frac{\epsilon}{2\sqrt{\log(1/\delta)}}$ for all $t \in [n^2]$. Now using the following lemma (Lemma II.2 with $\epsilon' = \frac{\epsilon}{2\sqrt{\log(1/\delta)}}$ and $\gamma = 1/n$) we ensure that over the randomness of b_t's and the randomness due to sampling in Line 4 , w.p. at least $1 - \frac{\delta}{2}$, $W_t \leq \frac{\epsilon}{n\sqrt{\log(1/\delta)}}$ for all $t \in [n^2]$. While using Lemma II.2, we ensure that the condition $\frac{\epsilon}{2\sqrt{\log(1/\delta)}} \leq 1$ is satisfied.

Lemma II.2 (Privacy amplification via sampling. Lemma 4 in [4])**.** *Over a domain of data sets $\mathcal{T}^n$, if an algorithm $\mathcal{A}$ is $\epsilon' \leq 1$ differentially private, then for any data set $\mathcal{D} \in \mathcal{T}^n$, executing $\mathcal{A}$ on uniformly random γn entries of $\mathcal{D}$ ensures $2\gamma\epsilon'$-differential privacy.*

To conclude the proof, we apply "strong composition" (Lemma II.3) from [19]. With probability at least $1-\delta$, the privacy loss $W = \sum_{t=1}^{n^2} W_t$ is at most ϵ. This concludes the proof.

Lemma II.3 (Strong composition [19])**.** *Let $\epsilon, \delta' \geq 0$. The class of ϵ-differentially private algorithms satisfies (ϵ', δ')-differential privacy under T-fold adaptive composition for $\epsilon' = \sqrt{2T\ln(1/\delta')}\epsilon + T\epsilon(e^\epsilon - 1)$.*

■

In Theorem II.4 we provide the utility guarantees for Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ under two different settings, namely, when the function ℓ is Lipschitz, and when the function ℓ is Lipschitz and strongly convex. (For a proof, seefull version [3].) In Section V we argue that these excess risk bounds are essentially tight.

Note: In the full version [3], we show that one can plug in the empirical risk bounds into standard results from learning theory [35], to obtain excess generalization error (excess risk) bounds. The main crux of our results is that we obtain the same dependence on the number of samples (n), when compared to the non-private bounds. However, the private bounds have an explicit dependence on the dimensionality (p).

Theorem II.4 (Utility guarantee)**.** *Let $\sigma^2 = O\left(\frac{L^2 n^2 \log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$ and let $\mathsf{EmpRisk}(\theta) = \mathbb{E}\left[\mathcal{L}(\theta;\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right]$. For θ^{priv} output by Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ we have the following. (The expectation is over the randomness of the algorithm.)*

1) ***Lipschitz functions:*** *If we set the learning rate function $\eta_t(t) = \frac{\|\mathcal{C}\|_2}{\sqrt{t(n^2L^2+p\sigma^2)}}$, then we have the following excess risk bound. Here L is the Lipscthiz constant of the loss function ℓ.*

$$\mathsf{EmpRisk}(\theta^{priv}) = O\left(\frac{L\|\mathcal{C}\|_2 \log^{3/2}(n/\delta)\sqrt{p\log(1/\delta)}}{\epsilon}\right).$$

2) ***Lipschitz and strongly convex functions:*** *If we set the learning rate function $\eta_t(t) = \frac{1}{\Delta n t}$, then we have the following excess risk bound. Here L is the Lipscthiz constant of the loss function ℓ and Δ is the strong convexity parameter.*

$$\mathsf{EmpRisk}(\theta^{priv}) = O\left(\frac{L^2\log^2(n/\delta)p\log(1/\delta)}{n\Delta\epsilon^2}\right).$$

Proof: Let $G_t = n \bigtriangledown \ell(\widetilde{\theta}_t; d) + b_t$ in Line 5 of Algorithm 1. First notice that over the randomness of the sampling of the data entry d from $\mathcal{D}$, and the randomness of b_t, $\mathbb{E}[G_t] = \bigtriangledown\mathcal{L}(\widetilde{\theta}_t;\mathcal{D})$. Additionally, we have the following bound on $\mathbb{E}[\|G_t\|_2^2]$.

$$\begin{aligned}\mathbb{E}[\|G_t\|_2^2] &= n^2\mathbb{E}[\| \bigtriangledown \ell(\widetilde{\theta}_t; d)\|_2^2] + \\ &\quad 2n\mathbb{E}[\langle \bigtriangledown\ell(\widetilde{\theta}_t; d), b_t\rangle] + \mathbb{E}[\|b_t\|_2^2] \\ &\leq n^2L^2 + p\sigma^2 \qquad (2)\end{aligned}$$

In the above expression we have used the fact that since $\widetilde{\theta}_t$ is independent of b_t, so $\mathbb{E}[\langle\bigtriangledown\ell(\widetilde{\theta}_t; d), b_t\rangle] = 0$. Also, we have $\mathbb{E}[\|b_t\|_2^2] = p\sigma^2$. We can now directly use Lemma II.5 to obtain the required error guarantee for Lipschitz convex functions, and Lemma II.6 for Lipschitz and strongly convex functions.

Lemma II.5 (Theorem 2 from [36])**.** *Let $F(\theta)$ (for $\theta \in \mathcal{C}$) be a convex function and let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F(\theta)$. Let θ_1 be any arbitrary point from $\mathcal{C}$. Consider the stochastic gradient descent algorithm $\theta_{t+1} = \Pi_{\mathcal{C}}\left[\theta_t - \eta(t)G_t(\theta_t)\right]$, where $E[G_t(\theta_t)] = \bigtriangledown F(\theta_t)$, $E[\|G_t\|_2^2] \leq G^2$ and the learning rate function $\eta(t) = \frac{\|\mathcal{C}\|_2}{G\sqrt{t}}$. Then for any $T > 1$, the following is true.*

$$\mathbb{E}\left[F(\theta_T) - F(\theta^*)\right] = O\left(\frac{\|\mathcal{C}\|_2 G\log T}{\sqrt{T}}\right).$$

Using the bound from (2) in Lemma II.5 (i.e., set $G = \sqrt{n^2L^2 + p\sigma^2}$), and setting $T = n^2$ and the learning rate function $\eta_t(t)$ as in Lemma II.5, gives us the required excess risk bound for Lipschitz convex functions. For Lipschitz and strongly convex functions we use the following result by [36].

Lemma II.6 (Theorem 1 from [36])**.** *Let $F(\theta)$ (for $\theta \in \mathcal{C}$) be a λ-strongly convex function and let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F(\theta)$. Let θ_1 be any arbitrary point from $\mathcal{C}$. Consider the stochastic gradient descent algorithm $\theta_{t+1} = \Pi_{\mathcal{C}}\left[\theta_t - \eta(t)G_t(\theta_t)\right]$, where $E[G_t(\theta_t)] = \bigtriangledown F(\theta_t)$, $E[\|G_t\|_2^2] \leq G^2$ and the learning rate function $\eta(t) = \frac{1}{\lambda t}$. Then for any $T > 1$, the following is true.*

$$\mathbb{E}\left[F(\theta_T) - F(\theta^*)\right] = O\left(\frac{G^2\log T}{\lambda T}\right).$$

Using the bound from (2) in Lemma II.6 (i.e., set $G = \sqrt{n^2L^2 + p\sigma^2}$), $\lambda = n\Delta$, and setting $T = n^2$ and the learning rate function $\eta_t(t)$ as in Lemma II.6, gives us the required excess risk bound for Lipschitz and strongly convex convex functions. ■

Note: Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ has a running time of $O(pn^2)$, assuming that the gradient computation for ℓ takes time $O(p)$.

III. Exponential Sampling and Optimal $(\epsilon, 0)$-private Optimization

In this section, we focus on the case of pure ϵ-differential privacy and provide an optimal efficient algorithm for empirical risk minimization for the general class of convex and Lipschitz loss functions. The main building block of this section is the well-known exponential mechanism [31].

First, we show that a variant of the exponential mechanism is optimal. A major technical contribution of this section is

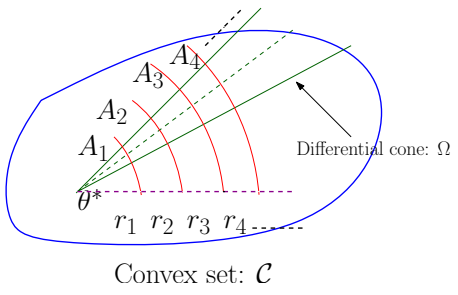

Fig. 1. Differential cone Ω inside the convex set $\mathcal{C}$

to make the exponential mechanism computationally efficient which is discussed in Section III-B.

A. Exponential Mechanism for Lipschitz Convex Loss

In this section, we only deal with loss functions which are Lipschitz. We provide an ϵ-differentially private algorithm (Algorithm 2) which achieves the optimal excess risk for arbitrary convex bounded sets.

Algorithm 2 $\mathcal{A}_{\mathsf{exp-samp}}$: Exponential sampling based convex optimization

Input: Data set of size n: $\mathcal{D}$, loss function ℓ, privacy parameter ϵ and convex set $\mathcal{C}$.

1: $\mathcal{L}(\theta;\mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i)$.

2: Sample a point θ^{priv} from the convex set $\mathcal{C}$ w.p. proportional to $\exp\left(-\frac{\epsilon}{2L\|\mathcal{C}\|_2}\mathcal{L}(\theta;\mathcal{D})\right)$ and output.

Theorem III.1 (Privacy guarantee)**.** *Algorithm 2 is ϵ-differentially private.*

Proof: Note that the distribution in step 2 will remain the same if we used $\exp\left(-\frac{\epsilon}{L\|\mathcal{C}\|_2}\left(\mathcal{L}(\theta;\mathcal{D}) - \mathcal{L}(\theta_0;\mathcal{D})\right)\right)$ for some arbitrary point $\theta_0 \in \mathcal{C}$. The proof then follows from the fact that the sensitivity of $\mathcal{L}(\theta;\mathcal{D}) - \mathcal{L}(\theta_0;\mathcal{D})$ is at most $L\|\mathcal{C}\|_2$ and the analysis of the *exponential mechanism* by [31]. ∎

Theorem III.2 (Utility guarantee)**.** *Let θ^{priv} be the output of $\mathcal{A}_{\mathsf{exp-samp}}$ (Algorithm 2 above). Then, we have the following guarantee on the expected excess risk. (The expectation is over the randomness of the algorithm.)*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] = O\left(\frac{pL\|\mathcal{C}\|_2}{\epsilon}\right).$$

Proof: Consider a differential cone Ω centered at θ^* (see Figure 1). We will bound the expected excess risk of θ^{priv} by $O\left(\frac{pL\|\mathcal{C}\|_2}{\epsilon}\right)$ conditioned on $\theta^{priv} \in \Omega \cap \mathcal{C}$ for every differential cone. This immediately implies the above theorem by the properties of conditional expectation.

Let Γ be a fixed threshold (to be set later) and let $\mathsf{R}(\theta) = \mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})$ for the purposes of brevity. Let the marked sets A_i's in Figure 1 be defined as

$$A_i = \{\theta \in \Omega \cap \mathcal{C} : (i-1)\Gamma \le \mathsf{R}(\theta) \le i \cdot \Gamma\}.$$

Instead of directly computing the probability of θ^{priv} being outside A_1, we will analyze the probabilities for being in each of the A_i's individually. This form of "peeling" arguments have been used for risk analysis of convex loss in the machine learning literature (e.g., see [40]) and will allow us to get rid of the extra logarithmic factor that would have otherwise shown up in the excess risk if we use the standard analysis of the exponential mechanism in [31].

Since the Ω is a differential cone and since $\mathsf{R}(\theta)$ is continuous on $\mathcal{C}$, it follows that within $\Omega \cap \mathcal{C}$, $\mathsf{R}(\theta)$ only depends on $\|\theta - \theta^*\|_2$. Therefore, let $r_1, r_2, \cdots$ be the distance of the set boundaries of $A_1, A_2, \cdots$ from θ^*. (See Figure 1.) One can equivalently write each A_i as follows:

$$A_i = \{\theta \in \Omega \cap \mathcal{C} : r_{i-1} < \|\theta - \theta^*\|_2 \le r_i\}.$$

The following claim (which is proved in the full version [3]) is the key part of the proof.

Claim III.3. *Convexity of $\mathsf{R}(\theta)$ for all $\theta \in \mathcal{C}$ implies that $r_i - r_{i-1} \le r_{i-1} - r_{i-2}$ for all $i \ge 3$.*

Now, the volume of the set A_i is given by $\mathsf{Vol}(A_i) = \kappa \int_{r_{i-1}}^{r_i} r^{p-1} dr$ for some fixed constant κ. Hence,

$$\frac{\mathsf{Vol}(A_i)}{\mathsf{Vol}(A_2)} = \frac{r_{i-1}^p}{r_1^p} \cdot \frac{(r_i/r_{i-1})^p - 1}{(r_2/r_1)^p - 1} \le \frac{r_{i-1}^p}{r_1^p} \le (i-1)^p.$$

where the last two inequalities follows from Claim III.3. Let $\gamma = \frac{\Pr[\theta^{priv} \in \bigcup_{i=4}^{\infty} A_i]}{\Pr[\theta^{priv} \in A_2]}$. Hence, γ can be bounded as

$$\gamma \le \sum_{i=4}^{\infty} \frac{\mathsf{Vol}(A_i)}{\mathsf{Vol}(A_2)} \cdot e^{-\epsilon(i-3)\frac{\Gamma}{2L\|\mathcal{C}\|_2}}$$
$$\le \sum_{i=4}^{\infty} (i-1)^p \cdot e^{-\epsilon(i-3)\frac{\Gamma}{2L\|\mathcal{C}\|_2}} \le \frac{3^p e^{-\epsilon\frac{\Gamma}{2L\|\mathcal{C}\|_2}}}{1 - 2^p e^{-\epsilon\frac{\Gamma}{2L\|\mathcal{C}\|_2}}}$$

where we use the fact that $(i-1)^p \le 3^p \cdot \left(2^{i-4}\right)^p$ for $i \ge 4$ in the last inequality which holds when Γ is sufficiently large. Hence, for every $t > 0$, if we choose $\Gamma = \frac{2L\|\mathcal{C}\|_2}{\epsilon}\left((p+1)\ln 3 + t\right)$, we get $\gamma \le e^{-t}$. Thus, conditioned on $\theta^{priv} \in \mathcal{C} \cap \Omega$, we have $\Pr[\mathsf{R}(\theta^{priv}) \ge \frac{8L\|\mathcal{C}\|_2}{\epsilon}\left((p+1)\ln 3 + t\right)] \le e^{-t}$. Since this is true for every $t > 0$, we have our required bound as a corollary. ∎

B. Efficient Implementation of Algorithm 2

In this section, we give a high-level description of a computationally efficient construction of Algorithm 2. Our algorithm runs in polynomial time in n, p and outputs a sample $\theta \in \mathcal{C}$ from a distribution that is arbitrarily close (in the multiplicative sense) to the distribution of the output of Algorithm 2.

Since we are interested in an efficient pure ϵ-differentially private algorithm, we need an efficient sampler with a multiplicative distance guarantee. In fact, if we were interested

in (ϵ, δ) algorithms, efficient sampling with a total variation guarantee would have sufficed which would have made our task a lot easier as we could have used one of the exisiting algorithms, e.g., [30]. In [20], it was shown how to sample efficiently with a multiplicative guarantee from the *unifrom* distribution over a convex bounded set. However, what we want to achieve here is more general, that is, to sample efficiently from any given logconcave distribution defined over a convex bounded set. To the best of our knowledge, this task has not been explicitly worked out before, nevertheless, all the ingredients needed to accomplish it are present in the literature, mainly [2].

We highlight here the main ideas of our constrution, however, due to space constraints and since such construction is not specific to our privacy problem, we provide the details of such construction and the proof of our main result in this section (Theorem III.4 below) in the full version [3].

Theorem III.4. *There is an efficient version of Algorithm 2 that has the following guarantees.*

1) ***Privacy:*** *The algorithm is ϵ -differentially private.*
2) ***Utility:*** *The output $\theta^{priv} \in \mathcal{C}$ of the algorithm satisfies*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{pL\|\mathcal{C}\|_2}{\epsilon}\right).$$

3) ***Running time:*** *Assuming $\mathcal{C}$ is in isotropic position, the algorithm runs in time*[1]

$$O\left(\|\mathcal{C}\|_2^2 p^3 n^3 \max\left\{p \log(\|\mathcal{C}\|_2 pn), \epsilon \|\mathcal{C}\|_2 n\right\}\right).$$

In fact, the running time of our algorithm depends on $\|\mathcal{C}\|_\infty$ rather than $\|\mathcal{C}\|_2$. Namely, all the $\|\mathcal{C}\|_2$ terms in the running time can be replaced with $\|\mathcal{C}\|_\infty$, however, we chose to write it in this less conservative way since all the bounds in this paper are expressed in terms of $\|\mathcal{C}\|_2$.

Before describing our construction, we first introduce some useful notation and discuss some preliminaries.

For any two probability measures μ, ν defined with respect to the same sample space $\mathcal{Q} \subseteq \mathbb{R}^p$, the relative (multiplicative) distance between μ and ν, denoted as $\mathsf{Dist}_\infty(\mu, \nu)$ is defined as

$$\mathsf{Dist}_\infty(\mu, \nu) = \sup_{q \in \mathcal{Q}} \left| \log \frac{d\mu(q)}{d\nu(q)} \right|.$$

where $\frac{d\mu(q)}{d\nu(q)}$ (resp., $\frac{d\nu(q)}{d\mu(q)}$) denotes the ratio of the two measures (more precisely, the Radon-Nikodym derivative).

Assumptions: We assume that we can efficiently test whether a given point $\theta \in \mathbb{R}^p$ lies in $\mathcal{C}$ using a membership oracle. We also assume that we can efficienly optimize an efficiently computable convex function over a convex set. To do this, it suffices to have a projection oracle. We do not take into account the extra polynomial factor in the running time which is required to perform such operations since this factor is highly dependent on the specific structure of the set $\mathcal{C}$.

[1]If $\mathcal{C}$ is not in isotropic position, the running time will pick up an extra factor of $O(\max\left(p^2, \text{polylog}\left(\frac{1}{r}\right)\right)$ where r is the diameter of the largest ball we can fit inside $\mathcal{C}$. See the full version [3]for details.

C. Our construction

Let τ denote the L_∞ diameter of $\mathcal{C}$. The Minkowski's norm of $\theta \in \mathbb{R}^p$ with respect to $\mathcal{C}$, denoted as $\psi(\theta)$, is defined as $\psi(\theta) = \inf\{r > 0 : \theta \in r\mathcal{C}\}$. We define $\bar{\psi}_\alpha(\theta) \triangleq \alpha \cdot \max\{0, \psi(\theta) - 1\}$ for $\alpha > 0$. Note that $\bar{\psi}_\alpha(\theta) > 0$ if and only if $\theta \notin \mathcal{C}$. Moreover, it is not hard to verify that $\bar{\psi}_\alpha$ is α-Lipschitz.

We use the grid-walk algorithm of [2] for sampling from a logconcave distribution defined over a *cube* as a building block. Our construction is described as follows:

1) Enclose the set $\mathcal{C}$ with a cube A with edges of length τ.
2) Obtain a convex Lipschitz extension $\bar{\mathcal{L}}(.; \mathcal{D})$ of the loss function $\mathcal{L}(.; \mathcal{D})$ over A. This can be done efficiently using a projection oracle.
3) Define $F(\theta) \triangleq e^{-\frac{\epsilon}{6L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\theta;\mathcal{D}) - \bar{\psi}_\alpha(\theta)}$, $\theta \in A$, for a specific choice of $\alpha = O(\frac{\epsilon n}{\|C\|_2})$ (See full version [3] for details).
4) Run the grid-walk algorithm of [2] with F as the input weight function and A as the input cube, and output a sample θ whose distribution is close, with respect to Dist_∞, to the distribution induced by F on A which is given by $\frac{F(\theta)}{\int_{v \in A} F(v)dv}$, $\theta \in A$.

Let's denote the aove efficient procedure by $\mathcal{A}_{\mathsf{cube-samp}}$. We then argue that due to the choices made for the values of the parameters above, $\mathcal{A}_{\mathsf{cube-samp}}$ outputs a sample in $\mathcal{C}$ with probability at least $\frac{1}{2}$. That is, the algorithm succeeds to output a sample from a distribution close to the right distribution on $\mathcal{C}$ with probability at least $1/2$. Hence, we can amplify the probability of success by repeating $\mathcal{A}_{\mathsf{cube-samp}}$ sufficiently many times where fresh random coins are used by $\mathcal{A}_{\mathsf{cube-samp}}$ in every time (specifically, $O(n)$ iterations would suffice). If $\mathcal{A}_{\mathsf{cube-samp}}$ returns a sample $\theta \in \mathcal{C}$ in one of those iterations, then our algorithm terminates outputting θ. Otherwise, it outputs a uniformly random sample $\theta^\perp$ from the unit ball $\mathbb{B}$ (Note that $\mathbb{B} \subseteq \mathcal{C}$ since $\mathcal{C}$ is assumed to be in isotropic position). We finally show that this termination condition can only change the distribution of the output sample by a constant factor sufficiently close to 1. Hence, we obtain our efficient algorithm referred to in Theorem III.4.

IV. Localization and Optimal Private Algorithms for Strongly Convex Loss

It is unclear how to get a direct variant of Algorithm 2 in Section III for Lipschitz and strongly convex losses that can achieve optimal excess risk guarantees. The issue in extending Algorithm 2 directly is that the convex set $\mathcal{C}$ over which the exponential mechanism is defined is "too large" to provide tight guarantees.

We show a generic ϵ-differentially private algorithm for minimizing Lipschitz strongly convex loss functions based on a combination of a simple pre-processing step (called the *localization step*) and any generic ϵ-differentially private algorithm for Lipschitz convex loss functions. We carry out the localization step using a simple output perturbation algorithm which ensures that the convex set over which the ϵ-

differentially private algorithm (in the second step) is run has diameter $\tilde{O}(p/n)$.

Next, we instantiate the generic ϵ-differentially private algorithm in the second step with our efficient exponential sampling (Algorithm 2) to obtain an algorithm with optimal excess risk bound (Theorem IV.3).

Details of the generic algorithm: We first give a simple algorithm (Algorithm 3 below) that carries out the desired localization step. The crux of the algorithm is the same as to that of the output perturbation algorithm of [11].

Algorithm 3 $\mathcal{A}^{\epsilon}_{\mathsf{out-pert}}$: Output Perturbation for Strongly Convex Loss

Input: data set of size n: $\mathcal{D}$, loss function ℓ, strong convexity parameter Δ, privacy parameter ϵ, convex set $\mathcal{C}$, and radius parameter $\zeta < 1$.

1: $\mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i)$.

2: Find $\theta^* = \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; \mathcal{D})$.

3: $\theta_0 = \Pi_{\mathcal{C}}(\theta^* + b)$, where b is random noise vector with density $\frac{1}{\alpha} e^{-\frac{n\Delta\epsilon}{2L}\|b\|_2}$ (where α is a normalizing constant) and $\Pi_{\mathcal{C}}$ is the projection on to the convex set $\mathcal{C}$.

4: Output $\mathcal{C}_0 = \{\theta \in \mathcal{C} : \|\theta - \theta_0\|_2 \leq \zeta \frac{2Lp}{\Delta\epsilon n}\}$.

Having Algorithm 3 in hand, we now give a generic ϵ-differentially private algorithm for minimizing $\mathcal{L}$ over $\mathcal{C}$. Let $\mathcal{A}^{\epsilon}_{\mathsf{gen-Lip}}$ denote any generic ϵ-differentially private algorithm for optimizing $\mathcal{L}$ over some arbitrary convex set $\tilde{\mathcal{C}} \subseteq \mathcal{C}$. Algorithm 2 from Section III-A (or its efficient version from Section III-B) is an example of $\mathcal{A}^{\epsilon}_{\mathsf{gen-Lip}}$. The algorithm we present here (Algorithm 4 below) makes a black-box call in its first step to $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$ (Algorithm 3 shown above), then, in the second step, it feeds the output of $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$ into $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ and ouptut.

Algorithm 4 Output-perturbation-based Generic Algorithm

Input: data set of size n: $\mathcal{D}$, loss function ℓ, strong convexity parameter Δ, privacy parameter ϵ, and convex set $\mathcal{C}$.

1: Run $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$ (Algorithm 3) with input privacy parameter $\epsilon/2$, radius parameter $\zeta = 3\log(n)$, and output $\mathcal{C}_0$.

2: Run $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ on inputs $n, \mathcal{D}, \ell$, privacy parameter $\epsilon/2$, and convex set $\mathcal{C}_0$, and output θ^{priv}.

Theorem IV.1 (Privacy guarantee)**.** *Algorithm 4 is ϵ-differentially private.*

Proof: The privacy guarantee follows directly from the composition theorem together with the fact that $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$ is $\frac{\epsilon}{2}$-differentially private (see [11]) and that $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ is $\frac{\epsilon}{2}$-differentially private by assumption. ∎

In the following lemma (see the full version [3] for a proof), we provide a generic expression for the excess risk of Algorithm 4 in terms of the expected excess risk of any given algorithm $\mathcal{A}_{\mathsf{gen-Lip}}$.

Lemma IV.2 (Generic utility guarantee)**.** *Let $\widetilde{\theta}$ denote the output of Algorithm $\mathcal{A}^{\epsilon}_{\mathsf{gen-Lip}}$ on inputs $n, \mathcal{D}, \ell, \epsilon, \tilde{\mathcal{C}}$ (for an arbitrary convex set $\tilde{\mathcal{C}} \subseteq \mathcal{C}$). Let $\hat{\theta}$ denote the minimizer of $\mathcal{L}(.; \mathcal{D})$ over $\tilde{\mathcal{C}}$. If*

$$\mathbb{E}\left[\mathcal{L}(\widetilde{\theta}; \mathcal{D}) - \mathcal{L}(\hat{\theta}; \mathcal{D})\right] \leq F\left(p, n, \epsilon, L, \|\tilde{\mathcal{C}}\|_2\right)$$

for some function F, then the output θ^{priv} of Algorithm 4 satisfies

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(F\left(p, n, \epsilon, L, \frac{Lp\log(n)}{\Delta\epsilon n}\right)\right),$$

where $\theta^* = \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; \mathcal{D})$.

Instantiation of $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ with Algorithm 2: Next, we give our optimal ϵ-differentially private algorithm for Lipschitz strongly convex loss functions. To do this, we instantiate the generic Algorithm $\mathcal{A}_{\mathsf{gen-Lip}}$ in Algorithm 4 with our exponential sampling algorithm from Section III-A (Algorithm 2), or its efficient version (Section III-B) to obtain the optimal excess risk bound. We formally state the bound in Theorem IV.3) below whose proof follows from Theorem III.2 and Lemma IV.2 above.

Theorem IV.3 (Utility guarantee)**.** *Suppose we replace $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ in Algorithm 4 with Algorithm 2 (Section III-A). Then, the output θ^{priv} satisfies*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{p^2 L^2}{n\Delta\epsilon^2}\log(n)\right).$$

where $\theta^* = \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; \mathcal{D})$.

V. Lower Bounds on Excess Risk

In this section, we complete the picture by deriving lower bounds on the excess risk caused by differentially private algorithm for risk minimization. In Section V-A, we consider the case of convex Lipschitz loss functions, whereas in Section V-B, we consider the case of strongly convex and Lipschitz loss functions.

Before we state and prove our lower bounds, we first give the following useful lemma which gives lower bounds on the L_2-error incurred by ϵ and (ϵ, δ)-differentially private algorithms for estimating the 1-way marginals of datasets over $\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$. This lemma is based on the results of [20] and [9]. We give a detailed proof of this lemma in the full version of our paper [3].

Lemma V.1 (Lower bounds for 1-way marginals)**.**

1) ***ϵ-differential private algorithms:*** *Let $n, p \in \mathbb{N}$ and $\epsilon > 0$. There is a number $M = \Omega(\min(n, p/\epsilon))$ such that for every ϵ-differentially private algorithm $\mathcal{A}$, there is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^{n} d_i\|_2 \in [M-1, M+1]$ such that, with probability at least $1/2$ (taken over the algorithm random coins), we have*

$$\|\mathcal{A}(\mathcal{D}) - q(\mathcal{D})\|_2 = \Omega\left(\min\left(1, \frac{p}{\epsilon n}\right)\right)$$

where $q(\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} d_i$.

2) **(ϵ, δ)-differential private algorithms:** *Let* $n, p \in \mathbb{N}$, $\epsilon > 0$, *and* $\delta = o(\frac{1}{n})$. *There is a number* $M = \Omega\left(\min\left(n, \sqrt{p}/\epsilon\right)\right)$ *such that for every* (ϵ, δ)*-differentially private algorithm* $\mathcal{A}$, *there is a dataset* $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ *with* $\|\sum_{i=1}^{n} d_i\|_2 \in [M-1, M+1]$ *such that, with probability at least* $1/3$ *(taken over the algorithm random coins), we have*

$$\|\mathcal{A}(\mathcal{D}) - q(\mathcal{D})\|_2 = \Omega\left(\min\left(1, \frac{\sqrt{p}}{\epsilon n}\right)\right)$$

where $q(\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} d_i$.

A. Lower bounds for Lipschitz Convex Functions

We consider the case where the data points are drawn from $\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, the parameter set is the p-dimensional unit ball $\mathbb{B}$, and the loss function ℓ is given by

$$\ell(\theta; d) = -\langle \theta, d\rangle, \quad \theta \in \mathbb{B}, \quad d \in \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p.$$

Clearly, ℓ is linear, hence convex, and 1-Lipschitz. Hence, for any dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, and any $\theta \in \mathbb{B}$, we have $\mathcal{L}(\theta; \mathcal{D}) = -\langle \theta, \sum_{i=1}^{n} d_i\rangle$.

Note that, whenever $\|\sum_{i=1}^{n} d_i\|_2 > 0$, $\theta^* = \frac{\sum_{i=1}^{n} d_i}{\|\sum_{i=1}^{n} d_i\|_2}$ is the minimizer of $\mathcal{L}(.; \mathcal{D})$ over $\mathbb{B}$. Our lower bounds are formally stated below.

Theorem V.2 (Lower bound for ϵ-differentially private algorithms)**.** *Let* $n, p \in \mathbb{N}$ *and* $\epsilon > 0$. *For every* ϵ*-differentially private algorithm (whose output is denoted by* θ^{priv}*), there is a dataset* $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ *such that, with probability at least* $1/2$ *(over the algorithm random coins), we must have*

$$\mathcal{L}(\theta; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \Omega\left(\min\left(n, p/\epsilon\right)\right)$$

where $\theta^* = \frac{\sum_{i=1}^{n} d_i}{\|\sum_{i=1}^{n} d_i\|_2}$ *is the minimizer of* $\mathcal{L}(.; \mathcal{D})$ *over* $\mathbb{B}$.

Proof: Let $\mathcal{A}$ be an ϵ-differentially private algorithm for minimizing $\mathcal{L}$ and let θ^{priv} denote its output. First, observe that for any $\theta \in \mathbb{B}$ and dataset $\mathcal{D}$, $\mathcal{L}(\theta; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \|\sum_{i=1}^{n} d_i\|_2 \left(1 - \langle \theta, \theta^*\rangle\right)$. Hence, we have $\mathcal{L}(\theta; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) \geq \frac{1}{2}\|\sum_{i=1}^{n} d_i\|_2 \|\theta - \theta^*\|_2^2$. This is due to the fact that $\|\theta - \theta^*\|_2^2 = \|\theta^*\|_2^2 + \|\theta\|_2^2 - 2\langle \theta, \theta^*\rangle$ and the fact that $\theta^*, \theta \in \mathbb{B}$.

Let $M = \Omega\left(\min\left(n, p/\epsilon\right)\right)$ be as in Part 1 of Lemma V.1. Suppose, for the sake of a contradiction, that for every dataset $\mathcal{D} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^{n} d_i\|_2 \in [M-1, M+1]$, with probability more than $1/2$, we have $\|\theta^{priv} - \theta^*\|_2 \neq \Omega(1)$. Let $\tilde{\mathcal{A}}$ be an ϵ-differentially private algorithm that first runs $\mathcal{A}$ on the data and then outputs $\frac{M}{n}\theta^{priv}$. Note that this implies that for every dataset $\mathcal{D} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^{n} d_i\|_2 \in [M-1, M+1]$, with probability more than $1/2$, $\|\tilde{\mathcal{A}}(\mathcal{D}) - q(\mathcal{D})\|_2 \neq \Omega\left(\min\left(1, \frac{p}{\epsilon n}\right)\right)$ which contradicts Part 1 of Lemma V.1. Thus, there must exist a dataset $\mathcal{D} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^{n} d_i\|_2 = \Omega\left(\min\left(n, p/\epsilon\right)\right)$ such that with probability at least $1/2$, we have $\|\theta^{priv} - \theta^*\|_2 = \Omega(1)$. Therefore, from the observation we made in the previous paragraph, we have, with probability at least $1/2$, $\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \Omega\left(\min\left(n, p/\epsilon\right)\right)$. ∎

Theorem V.3 (Lower bound for (ϵ, δ)-differentially private algorithms)**.** *Let* $n, p \in \mathbb{N}$, $\epsilon > 0$, *and* $\delta = o(\frac{1}{n})$. *For every* (ϵ, δ)*-differentially private algorithm (whose output is denoted by* θ^{priv}*), there is a dataset* $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ *such that, with probability at least* $1/3$ *(over the algorithm random coins), we must have*

$$\mathcal{L}(\theta; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \Omega\left(\min\left(n, \sqrt{p}/\epsilon\right)\right)$$

where $\theta^* = \frac{\sum_{i=1}^{n} d_i}{\|\sum_{i=1}^{n} d_i\|_2}$ *is the minimizer of* $\mathcal{L}(.; \mathcal{D})$ *over* $\mathbb{B}$.

Proof: We use Part 2 of Lemma V.1 and follow the same lines of the proof of Theorem V.2. ∎

B. Lower bounds for Strongly Convex Functions

We consider the same data universe and parameter set above. We choose our loss function here $\ell(\theta; d)$ to be

$$\ell(\theta; d) = \frac{1}{2}\|\theta - d\|_2^2, \quad \theta \in \mathbb{B}, \quad d \in \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p.$$

Note that ℓ is 1-Lipschitz and 1-strongly convex. Hence, for a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, we have $\mathcal{L}(\theta; \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n} \|\theta - d_i\|_2^2$.

Notice that the minimizer of $\mathcal{L}(.; \mathcal{D})$ over $\mathbb{B}$ is $\theta^* = \frac{1}{n}\sum d_i$ which is equal to $q(\mathcal{D})$ in the terminology of Lemma V.1. Note also that we can write the excess risk as

$$\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*) = \frac{n}{2}\|\theta^{priv} - q(\mathcal{D})\|_2^2. \tag{3}$$

Theorem V.4 (Lower bound for ϵ-differentially private algorithms)**.** *Let* $n \in \mathbb{N}$ *and* $\epsilon > 0$. *For every* ϵ*-differentially private algorithm (whose output is denoted by* θ^{priv}*), there is a dataset* $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ *such that, with probability at least* $1/2$ *(over the algorithm random coins), we must have*

$$\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \Omega\left(\min\left(n, \frac{p^2}{\epsilon^2 n}\right)\right)$$

where $\theta^* = \frac{1}{n}\sum d_i$ *is the minimizer of* $\mathcal{L}(.; \mathcal{D})$ *over* $\mathbb{B}$.

Proof: The proof follows directly from (3) and Part 1 of Lemma V.1. ∎

Theorem V.5 (Lower bound for (ϵ, δ)-differentially private algorithms)**.** *Let* $n \in \mathbb{N}$, $\epsilon > 0$, *and* $\delta = o(\frac{1}{n})$. *For every* (ϵ, δ)*-differentially private algorithm (whose output is denoted by* θ^{priv}*), there is a dataset* $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ *such that, with probability at least* $1/3$ *(over the algorithm random coins), we must have*

$$\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) = \Omega\left(\min\left(n, \frac{p}{\epsilon^2 n}\right)\right)$$

where $\theta^* = \frac{1}{n}\sum d_i$ *is the minimizer of* $\mathcal{L}(.; \mathcal{D})$ *over* $\mathbb{B}$.

Proof: The proof follows directly from (3) and Part 2 of Lemma V.1. ∎

Note: In the full version [3], we provide a simple reduction to transform our lower bounds above to the case of arbitrary $L, \|\mathcal{C}\|_2$, and Δ.

ACKNOWLEDGMENTS

We are grateful to Santosh Vempala and Ravi Kannan for discussions about efficient sampling algorithms for log-concave distributions over convex bodies. In particular, Ravi suggested the idea of using a penalty term to reduce from sampling over $\mathcal{C}$ to sampling over the cube.

R.B. and A.S. were supported in part by NSF awards #0747294 and #0941553. A.S. was also partly supported by Boston University's Hariri Institute for Computing and Center for RISCS, as well as by the Harvard Center for Research on Computation and Society, through a Simons Investigator grant to Salil Vadhan. A.T. was supported in part by an award from the Sloan Foundation.

REFERENCES

[1] A. Agarwal, P. L. Bartlett, P. D. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

[2] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *STOC*, 1991.

[3] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. *CoRR*, arXiv:1405.7085 [cs.LG], 2014.

[4] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3), 2014.

[5] A. Beimel, K. Nissim, and U. Stemmer. Characterizing the sample complexity of private learners. *CoRR*, abs/1402.2224, 2014.

[6] A. Beimel, K. Nissim, and U. Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *CoRR*, abs/1407.2674, 2014.

[7] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *PODS*, pages 128–138. ACM, 2005.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[9] M. Bun, J. Ullman, and S. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, 2014.

[10] K. Chaudhuri and D. Hsu. Sample complexity bounds for differentially private learning. In S. M. Kakade and U. von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

[11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *JMLR*, 12:1069–1109, 2011.

[12] K. Chaudhuri, A. D. Sarwate, and K. Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14(1):2905–2943, 2013.

[13] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210. ACM, 2003.

[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2013.

[15] C. Dwork. Differential privacy. In *ICALP*, LNCS, pages 1–12, 2006.

[16] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

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

[18] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO*, LNCS, pages 528–544. Springer, 2004.

[19] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, 2010.

[20] M. Hardt and K. Talwar. On the geometry of differential privacy. In *STOC*, 2010.

[21] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24.1–24.34, 2012.

[22] P. Jain and A. Thakurta. Differentially private learning with kernels. In *ICML (3)*, volume 28 of *JMLR Proceedings*, pages 118–126. JMLR.org, 2013.

[23] P. Jain and A. Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning (ICML)*, 2014.

[24] M. Kapralov and K. Talwar. On differentially private low rank approximation. In S. Khanna, editor, *SODA*, pages 1395–1414. SIAM, 2013.

[25] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *FOCS*, 2008.

[26] S. P. Kasiviswanathan, M. Rudelson, and A. Smith. The power of linear reconstruction attacks. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.

[27] S. P. Kasiviswanathan and A. Smith. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*, arXiv:0803.39461 [cs.CR], 2008.

[28] D. Kifer and A. Machanavajjhala. A rigorous and customizable framework for privacy. In *PODS*, pages 77–88, 2012.

[29] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25.1–25.40, 2012.

[30] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

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

[32] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, 1983.

[33] A. Nikolov, K. Talwar, and L. Zhang. The geometry of differential privacy: The sparse and approximate cases. In *STOC*, 2013.

[34] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *CoRR*, abs/0911.5708, 2009.

[35] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic Convex Optimization. In *COLT*, 2009.

[36] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, pages 71–79, 2013.

[37] A. Smith and A. Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory (COLT)*, 2013.

[38] A. Smith and A. Thakurta. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Neural Information Processing Systems (NIPS)*, 2013.

[39] S. Song, K. Chaudhuri, and A. Sarwate. Stochastic gradient descent with differentially private updates. In *Proc. of the Global Conference on Signal and Information Processing*, pages 245–248, December 2013.

[40] K. Sridharan, S. Shalev-shwartz, and N. Srebro. Fast rates for regularized objectives. In *NIPS*, 2008.

[41] O. Williams and F. McSherry. Probabilistic inference and differential privacy. In *NIPS*, 2010.