Private Empirical Risk Minimization, Revisited

Raef Bassily* Adam Smith*† Abhradeep Thakurta‡

April 10, 2014

Abstract

In this paper, we initiate a systematic investigation of differentially private algorithms for convex empirical risk minimization. Various instantiations of this problem have been studied before. We provide new algorithms and matching lower bounds for private ERM assuming only that each data point's contribution to the loss function is Lipschitz bounded 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.

*Computer Science and Engineering Department, The Pennsylvania State University. {bassily,asmith}@psu.edu

†A.S. is on leave at, and partly supported by, Boston University's Hariri Center for Computational Science.

‡Stanford University and Microsoft Research. b-abhrag@microsoft.com

Contents

1 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 initiate a systematic investigation of *differentially private* algorithms for convex empirical risk minimization. Various instantiations of this problem have been studied before. We provide new algorithms and matching lower bounds for private ERM assuming only that each data point's contribution to the loss function is Lipschitz bounded 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.

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

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

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 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 over the coins of the algorithm. Expected risk guarantees can be converted to high-probability guarantees using standard techniques (Appendix C).

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.

Note that we can always set $L = \|\mathcal{C}\|_2 = 1$ by rescaling the set $\mathcal{C}$ and the loss functions; in that case, we always have $\Delta \leq 2$ (since strong convexity implies that the size of the gradient changes at a rate of Δ over the diameter of the set). To convert excess risk bounds for $L = \|\mathcal{C}\|_2 = 1$ to the general setting, multiply the risk bounds by $L\|\mathcal{C}\|_2$, and replace Δ by $\frac{\Delta}{L\|\mathcal{C}\|_2}$.

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. Kasiviswanathan et al. [28] show how the results of many convex ERM problems can be combined to carry out reconstruction attacks in the spirit of Dinur and Nissim [9].

Differential privacy is a rigorous notion of privacy that emerged from a line of work in theoretical computer science and cryptography [10, 13, 3, 15]. 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 (Dwork et al. [15, 14]) 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 Dwork [12], Kasiviswanathan and Smith [26], Kifer and Machanavajjhala [29] for discussion of the "semantics" of differential privacy.

1.1 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 1.

Gradient descent-based algorithms. For (ϵ, δ)-differential privacy, we give an algorithm based on stochastic gradient descent that achieves excess risk $\tilde{O}(\sqrt{p}/\epsilon)$. This matches our lower bound, $\Omega(\min(n, \sqrt{p}/\epsilon))$,

	$(\epsilon, 0)$-DP			(ϵ, δ)-DP		
	Previous [7]	This work		Previous [30]	This work	
Assumptions	Upper Bd	Upper Bd	Lower Bd	Upper Bd	Upper Bd	Lower Bd
1-Lipschitz and $\|C\|_2 = 1$	$\frac{p\sqrt{n}}{\epsilon}$	$\frac{p \log(n/r)}{\epsilon}$	$\frac{p}{\epsilon}$	$\frac{\sqrt{p \cdot n} \log(1/\delta)}{\epsilon}$	$\frac{\sqrt{p} \log^2(1/\delta)}{\epsilon}$	$\frac{\sqrt{p}}{\epsilon}$
... 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 $\Delta \leq 2$)	$\frac{p^2}{\sqrt{n}\Delta\epsilon^2}$	$\frac{\log^2(n/r)}{\Delta} \cdot \frac{p^2}{n\epsilon^2}$	$\frac{p^2}{n\epsilon^2}$	$\frac{p \log(1/\delta)}{\sqrt{n}\Delta\epsilon^2}$	$\frac{\log^3(1/\delta)}{\Delta} \cdot \frac{p}{n\epsilon^2}$	$\frac{p}{n\epsilon^2}$
... and $O(p)$-smooth	$\frac{p^2}{n\Delta\epsilon^2}$		$\frac{p^2}{n\Delta\epsilon^2}$	$\frac{p \log(1/\delta)}{n\Delta\epsilon^2}$		$\frac{p}{n\epsilon^2}$

Table 1: 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 $\mathcal{C}$ and $r > 0$ satisfies $r\mathbb{B} \subseteq \mathcal{C}$ where $\mathbb{B}$ is the unit ball. 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 also assume $\delta < 1/n$ to simplify the bounds.

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 (Chaudhuri et al. [7], Kifer et al. [30], Jain et al. [24], Duchi et al. [11])). Under the restriction that each data point's contribution to the loss function is sufficiently smooth, objective perturbation [7, 30] 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. Chaudhuri et al. [7] suggest applying their technique to support vector machines by smoothing ("huberizing") the loss function. As we show in Appendix E, though, this approach still yields expected excess risk $\Omega(\sqrt{pn}/\epsilon)$.

The algorithm's structure is very simple: 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 estimate to update. The algorithm is similar to algorithms that have appeared previously (Williams and McSherry [41] first investigated gradient descent with noisy updates; stochastic variants were studied by Jain et al. [24], Duchi et al. [11], Song et al. [39]). The novelty of our analysis lies in taking advantage of the randomness in the choice of d_i (following Kasiviswanathan et al. [27]) 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. We note that 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 [33, 1]). Thus, the running time (more precisely, query complexity) of our algorithm is also optimal.

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)-privacyDwork et al. [16] 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 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 has an additional factor of $\log(1/r)$ in the risk guarantee, where r is the radius of the largest ball contained in $\mathcal{C}$. The idea is to sample from the continuous distribution on all points in $\mathcal{C}$ with density $\mathcal{P}(\theta) \propto e^{-\epsilon\mathcal{L}(\theta)}$. (It is the utility analysis of this algorithm introduces a dependency on r.) 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 (Hardt and Talwar [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 Applegate and Kannan [2]. Along the way, we give several technical results of possibly independent interest. For example, we show that (roughly) every efficiently computable, convex Lipschitz function on a convex set $\mathcal{C}$ can be extended to a efficiently computable, convex Lipschitz function on all of $\mathbb{R}^p$.

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. Chaudhuri et al. [7] and [35] 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 $\rho = \frac{p}{\Delta\epsilon n}$, with Δ-strong convexity of each loss function). 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 lower bound the accuracy of 1-way marginals (due to Hardt and Talwar [20] for $(\epsilon, 0)-$ and Bun et al. [5] for (ϵ, δ)-privacy) to show that our algorithms have essentially optimal risk bounds. For each of the categories of functions we consider (with and without strong convexity), we construct very simple instances in the lower bound: the functions can be linear or quadratic (for the case of strong convexity), and optimization can be performed either over the unit ball or the hypercube. In particular, our lower bounds apply to the problem of optimizing smooth functions, demonstrating the optimality of objective perturbation [7, 30] in that setting. The reduction to lower-bounds for 1-way marginals is not black-box; our lower bounds start from the instances used by Hardt and Talwar

[20], Bun et al. [5], but require specific properties from their analysis.

1.2 Other Related Work

In addition to the previous work mentioned above, we mention several closely related works. Jain and Thakurta [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). Kifer et al. [30], Smith and Thakurta [38] considered parameter convergence for high-dimensional sparse regression (where $p \gg n$). The settings of all those papers are orthogonal to the one considered here, though it would be interesting to find a useful common generalization.

Efficient implementations of the exponential mechanism over infinite domains were discussed by Hardt and Talwar [20], Chaudhuri et al. [8] and Kapralov and Talwar [25]. 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 (other than ERM) has also been studied: for example, Jain et al. [24], Duchi et al. [11], Smith and Thakurta [37] study online optimization, Jain and Thakurta [22] study an interactive model tailored to high-dimensional kernel learning. Convex optimization techniques have also played an important role in the development of algorithms for "simultaneous query release" (e.g., the line of work emerging from Hardt and Rothblum [19]). We do not know of a direct connection between those works and our setting.

1.3 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, in this case, 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 $\Pi_{\mathcal{C}}(\theta) = \arg\min_{x \in \mathcal{C}} \|\theta - x\|_2$.

2 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 [4]. Our algorithm (and the utility analysis) was inspired by the approach of Williams and McSherry [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 Section C we provide a generic tool to translate the expectation bounds into high probability bound albeit a loss of extra logarithmic factor in the inverse of the failure probability.

Note: 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.

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 $\sigma^2 \leftarrow O\left(\frac{L^2 n^2 \log(n/\delta) \log(1/\delta)}{\epsilon^2}\right)$.

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: $\widetilde{\theta}_{t+1} = \Pi_{\mathcal{C}}\left(\widetilde{\theta}_t - \eta(t)\left[n \bigtriangledown \ell(\widetilde{\theta}_t; d) + b_t\right]\right), b_t \sim \mathcal{N}\left(0, \mathbb{I}_p \sigma^2\right)$.

6: Output $\theta^{priv} = \widetilde{\theta}_{n^2}$.

Theorem 2.1 (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 $\mu^t_{\mathcal{D}}(y)$ to be the measure of the random variable $X_t(\mathcal{D})$ for given $y \in \mathbb{R}$. For any two neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$ define the *privacy loss* random variable [16] to be $W_t = \left|\log \frac{\mu_{\mathcal{D}}(X_t(\mathcal{D}))}{\mu_{\mathcal{D}'}(X_t(\mathcal{D}))}\right|$. Standard differential privacy arguments with Gaussian noise (see [30, 34]) will ensure that with probability $1 - \frac{\delta}{2}$ (over the randomness of the random variables b_t's), $W_t \leq \frac{\epsilon}{2\sqrt{\log(1/\delta)}}$ for all $t \in [n^2]$. Now using the following lemma (Lemma 2.2) 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 $t \in [n^2]$. (Notice the randomness of the random variable $X_t(\mathcal{D})$ is now both over b_t and the sampling.) While using Lemma 2.2, we set $\gamma = 1/n$ in the lemma and ensure that the condition $\frac{\epsilon}{2\sqrt{\log(1/\delta)}} \leq 1$ is true.

Lemma 2.2 (Privacy amplification via sampling [27])**.** *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 2.3) from [16]. 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 2.3 (Strong composition [16])**.** *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 the following 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. In Section 5 we argue that these excess risk bounds are essentially tight.

Theorem 2.4 (Utility guarantee). *Let $\sigma^2 = O\left(\frac{L^2 n^2 \log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$. 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.*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] = O\left(\frac{L\|\mathcal{C}\|_2 \log^{3/2}(n/\delta)\sqrt{p\log(1/\delta)}}{\epsilon}\right).$$

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.*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] = O\left(\frac{L^2\log^2(n/\delta)p\log(1/\delta)}{n\Delta\epsilon^2}\right).$$

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 $\mathbb{E}[\|G_t\|_2^2]$.

$$\begin{aligned}\mathbb{E}[\|G_t\|_2^2] &= n^2\mathbb{E}[\|\bigtriangledown \ell(\widetilde{\theta}_t; d)\|_2^2] + 2n\mathbb{E}[\langle\bigtriangledown\ell(\widetilde{\theta}_t; d), b_t\rangle] + \mathbb{E}[\|b_t\|_2^2] \\ &\leq n^2L^2 + p\sigma^2 \qquad \text{[Here } \sigma^2 \text{ is the variance of } b_t \text{ in Line 5]} \end{aligned} \tag{2}$$

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 Theorem 2.5 to obtain the required error guarantee for Lipschitz convex functions, and Theorem 2.6 for Lipschitz and strongly convex functions.

Lemma 2.5 (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 2.5 (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 2.5, gives us the required excess risk bound for Lipschitz convex functions. For Lipschitz and strongly convex functions we use the following theorem by [36].

Lemma 2.6 (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 Theorem 2.6 (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 Theorem 2.6, 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 $O(p)$. Variants of Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$ have appeared in [41, 24, 11, 40]. The most relevant work in our current context is that of [40]. The main idea in [40] is to run stochastic gradient descent with gradients computed over small batches of *disjoint* samples from the data set (as opposed to one single sample used in Algorithm $\mathcal{A}_{\mathsf{Noise-GD}}$). The issue with the algorithm is that it cannot provide excess risk guarantee which is $o(\sqrt{n})$, where n is the number of data samples. One observation that we make is that if one removes the constraint of disjointness and use the amplification lemma (Lemma 2.2), then one can ensure a much tighter privacy guarantees for the same setting of parameters used in the paper.

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

In the previous section we provided gradient descent based algorithms for both Lipschitz and, Lipschitz and strongly convex functions which are optimal for (ϵ, δ)-differential privacy. In this section we concentrate on the pure ϵ-differential privacy case, and provide optimal algorithms for the settings of the loss functions mentioned above. The key building block for our algorithms in this section is the well-known exponential mechanism [31].

For the Lipschitz case (Section 3.1), we show that a variant of the exponential mechanism is optimal. A major technical contribution of this section is to make the exponential mechanism computationally efficient. We borrow tools from rapidly mixing random walk theory to obtain a computationally efficient variant of the exponential mechanism. Our analysis is based on the grid random walk of [2], and a discussion with the second author of this paper.

3.1 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 convex sets which are in *isotropic positions*. For non-isotropic convex sets $\mathcal{C}$, via fairly generic transformations (see [18] for a reference) one can place $\mathcal{C}$ in a isotropic position with an increase in the Lipschitz constant of the loss function ℓ by a factor of $L\|\mathcal{C}\|_2$ and increase in the diameter of the set $\mathcal{C}$ by a factor of p. Hence, the excess risk bound in Algorithm 2 is off by a factor of p for arbitrary convex sets. However, in Appendix A (Algorithm 6) we provide a different exponential mechanism which achieves the optimal excess risk bound for arbitrary convex sets, however, algorithm is computationally inefficient.

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 $\exp\left(-\frac{\epsilon}{L\|\mathcal{C}\|_2}\mathcal{L}(\theta; \mathcal{D})\right)$ and output.

Theorem 3.1 (Privacy guarantee)**.** *Algorithm 2 is* 2ϵ*-differentially private.*

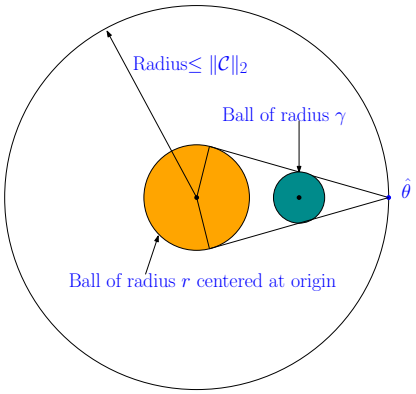

Figure 1: Geometry of convex set $\mathcal{C}$

Proof. First, notice that the distribution induced by the exponential weight function in step 2 of Algorithm 2 is 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 $\theta_0 \in \mathcal{C}$. Since ℓ is L-Lipschitz, the sensitivity of $\mathcal{L}(\theta;\mathcal{D})-\mathcal{L}(\theta_0;\mathcal{D})$ is at most $L\|C\|_2$. The proof then follows directly from the analysis of *exponential mechanism* by [31]. □

In the following we first prove the utility guarantee for a special class of convex sets $\mathcal{C}$ which contain ball of radius $r < \|\mathcal{C}\|_2$. Later we extend this guarantee to arbitrary convex sets via isotropic transformation.

Theorem 3.2 (Utility guarantee)**.** *Let $r\mathbb{B} \subseteq \mathcal{C} \subset \mathbb{R}^p$, where $\mathbb{B}$ is the p-dimensional unit ball. For θ^{priv} output by $\mathcal{A}_{\mathsf{exp-samp}}$ (Algorithm 2) we have the following. (The expectation is over the randomness of the algorithm.)*

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

Proof. First, we provide the following lemma which will be useful for the rest of the analysis.

Lemma 3.3. *Let $\mathcal{C} \subset \mathbb{R}^p$ be a convex set. Suppose there exists $r > 0$ such that $r\mathbb{B} \subseteq \mathcal{C}$ where $\mathbb{B}$ is a p-dimensional ball of unit radius. Let $\hat{\theta}$ be any point in $\mathcal{C}$. Then, for every γ such that $0 < \gamma \leq r$, there exists a ball $\mathcal{G} \subseteq \mathcal{C}$ of radius γ such that for all $\theta \in \mathcal{G}$, $\|\theta-\hat{\theta}\|_2 \leq \frac{\gamma}{r}\|\mathcal{C}\|_2$.*

The proof of the lemma follows immediately using similarity of triangles property. (See Figure 1.)

By Lemma 3.3, there is a ball of radius $\mathcal{G}$ of radius $\gamma = \frac{r}{\epsilon n}$ contained in $\mathcal{C}$ such that for all $\theta \in \mathcal{G}$, $\|\theta-\theta^*\|_2 \leq \frac{\|\mathcal{C}\|_2}{\epsilon n}$. Hence,for all $\theta \in \mathcal{G}$, $\mathcal{L}(\theta;\mathcal{D})-\mathcal{L}(\theta^*;\mathcal{D}) \leq \frac{L\|\mathcal{C}\|_2}{\epsilon}$.

Now, for any $t > 0$, we have

$$\Pr\left[\mathcal{L}(\theta^{priv};\mathcal{D})-\mathcal{L}(\theta^*;\mathcal{D}) \geq (1+t)\frac{L\|\mathcal{C}\|_2}{\epsilon}\right] \leq \frac{\mathsf{Vol}(\mathcal{C})}{\mathsf{Vol}(\mathcal{G})}e^{-t} = \left(\frac{\|\mathcal{C}\|_2/2}{\gamma}\right)^p e^{-t} \quad = e^{p\log\left(\frac{\epsilon n\|\mathcal{C}\|_2}{2r}\right)-t}$$

where the second equality above follows from the standard expression of the volume of p-dimensional L_2 balls. Setting $t = 2p\log\left(\frac{\epsilon n\|\mathcal{C}\|_2}{2r}\right)$ completes the proof.

□

3.2 Efficient Implementation of Algorithm $\mathcal{A}_{\mathsf{exp-samp}}$ (Algorithm 2)

In this section, we give a high-level description of a computationally efficient version of Algorithm 2. Our efficient algorithm, denoted as $\mathcal{A}_{\mathsf{eff-exp-samp}}$, outputs a sample $\theta \in \mathcal{C}$ from a distribution that is arbitrarily close (with respect to L_∞ distance) to a distribution that has the same form of the distribution in Algorithm $\mathcal{A}_{\mathsf{exp-samp}}$ (i.e., a distribution proportional to $e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\mathcal{L}(\theta;\mathcal{D})}$), where $\tilde{\epsilon}$ is within some constant factor of ϵ). (See definition Dist_∞ below.) The running time of $\mathcal{A}_{\mathsf{eff-exp-samp}}$ is polynomial in n, p.

In [20], a computationally efficient implementation of the exponential mechanism based on efficient sampling was given for a specific problem setting. In particular, in [20], it sufficed for their result to implement an efficient *uniform* sampling from a bounded convex set. To do this, Hardt and Talwar [20] used the grid-walk algorithm of [17] as the main block of their algorithm. However, in this section, what we want to achieve is more general than this. Namely, our goal is to sample efficiently from a logconcave distribution (i.e., the distribution proportional to $e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\mathcal{L}(\theta;\mathcal{D})}$) defined over an arbitrary bounded convex set $\mathcal{C}$. Hence, the same approach that is based on the algorithm of [17] is not applicable in our setting.

Our construction relies on a set of tools provided in [2]. Since our construction requires extending some ideas from previous work on efficient sampling from log-concave distribution over convex sets, in this section, we give some preliminary discussion of our tools and describe our approach. We defer the details of our construction and the proof of our main result in this section (Theorem thm:eff-samp-guarantees below) to Section 6. We show that our efficient algorithm yields the same privacy and utility guarantees of Theorems 3.1 and 3.2. This is formally stated in the following theorem.

Theorem 3.4. *There is an efficient version of Algorithm 2 (Algorithm 5 in Section 6) that has the following guarantees.*

1. ***Privacy:*** $\mathcal{A}_{\mathsf{eff-exp-samp}}$ *is* ϵ *-differentially private.*
2. ***Utility:*** *If* $r\mathbb{B} \subseteq \mathcal{C} \subset \mathbb{R}^p$*, where* $\mathbb{B}$ *is the* p*-dimensional unit ball, then the output* $\theta^{priv} \in \mathcal{C}$ *of* $\mathcal{A}_{\mathsf{eff-exp-samp}}$ *satisfies*

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

3. ***Running time:*** $\mathcal{A}_{\mathsf{eff-exp-samp}}$ *runs in time*[1]

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

Before describing our approach, we first introduce some useful notation. For any two probability measures μ, ν defined with respect to the same sample space $\mathcal{Q} \subseteq \mathbb{R}^p$, the relative L_∞ distance between μ and ν, denoted as $\mathsf{Dist}_\infty(\mu,\nu)$ is defined as[2]

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

[1]The expression of the running time assumes $\mathcal{C}$ to be in isotropic position. Otherwise, we replace $\|\mathcal{C}\|_2$ by $p\|\mathcal{C}\|_2^2$ in the running time where we pay an extra factor of $p\|\mathcal{C}\|_2$ since the diameter of $\mathcal{C}$ is inflated by at most a factor of p and the Lipschitz constant is amplified by a factor of $\|\mathcal{C}\|_2$ when $\mathcal{C}$ is placed in isotropic position.

[2]Note that this definition is slightly weaker than the standard definition, but it suffices for our analysis in this section.

where $\frac{d\mu(q)}{d\nu(q)}$ (resp., $\frac{d\nu(q)}{d\mu(q)}$) denotes the ratio of densities (Radon-Nikodym derivative) when both μ and ν are absolutely continuous, and denotes the ratio of the probability mass functions if both μ and ν are discrete probability measures.

In the following (as well as in the detailed analysis in Section 6), we will assume that the convex set $\mathcal{C}$ is in isotropic position. This assumption requires no loss of generality since one can always carry out an efficient transformation of $\mathcal{C}$ into a convex set $\mathcal{C}'$ such that $\mathcal{C}'$ is in isotropic position (see [18] for a reference), apply the efficient sampling algorithm on $\mathcal{C}'$, then apply the inverse transformation to the output to obtain a sample from the desired distribution over the original set $\mathcal{C}$. This would only cost an extra polynomial factor in the running time required to perform these transformations.

Let τ denote the L_∞ diameter of $\mathcal{C}$. That is, $\tau = \|\mathcal{C}\|_\infty \leq \|\mathcal{C}\|_2$. In fact, the running time of our algorithm depends on τ rather than $\|\mathcal{C}\|_2$. Namely, all the $\|\mathcal{C}\|_2$ terms in the running time in Theorem 3.4 can be replaced with τ, however, we chose to write it in this less conservative way since all the bounds in this paper are expressed in terms of $\|\mathcal{C}\|_2$.

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 some $\alpha > 0$ (to be specified later). Note that $\bar{\psi}_\alpha(\theta) > 0$ if and only if $\theta \notin \mathcal{C}$ since $\mathbf{0}^p \in \mathcal{C}$ (as $\mathcal{C}$ is assumed to be in isotropic position) and $\mathcal{C}$ is convex. Moreover, it is not hard to verify that $\bar{\psi}_\alpha$ is α-Lipschitz when $\mathcal{C}$ is in isotropic position.

Our approach: We use the grid-walk algorithm of [2] for sampling from a logconcave distribution defined over a *cube* as a building block. To extend this algorithm to the case of an arbitrary convex set (in isotropic position and with finite L_∞ diameter), we do the following. First, we enclose the set $\mathcal{C}$ with a cube A whose edges are of length τ (the L_∞ diameter of $\mathcal{C}$). We find a convex Lipschitz extension $\bar{\mathcal{L}}(.;\mathcal{D})$ of our loss function $\mathcal{L}(.;\mathcal{D})$ over A. Since we want the weight attributed to the region $A \setminus \mathcal{C}$ to be as small as possible, we modulate our logconcave distribution by a *gauge function* which is a standard trick in literature. Namely, we set our weight function[3] over A to be $F(\theta) \triangleq e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\theta;\mathcal{D}) - \bar{\psi}_\alpha(\theta)}$ for some choice of $\tilde{\epsilon}$ (See Section 6). Note that we use $\bar{\psi}_\alpha(\theta)$ as a gauge function to put far less weight on the points outside $\mathcal{C}$. Moreover, $F(\theta)$ is logconcave and its exponent (i.e., $\log(F(\theta))$) is $(\frac{n\tilde{\epsilon}}{\|\mathcal{C}\|_2} + \alpha)$-Lipschitz. Next, we use the grid-walk of [2] to generate a sample $\hat{u}$ (one of the grid points) whose distribution is close, with respect to Dist_∞, to the discrete distribution $\left(\frac{F(\hat{u})}{\sum_{\hat{v}\in\mathcal{S}} F(\hat{v})} : \hat{u} \in \mathcal{S}\right)$ where $\mathcal{S}$ is the set of grid points inside A. Then, we transform the resulting discrete distribution into a continuous distribution by sampling a point uniformly at random from the grid cell whose center is $\hat{u}$. The induced distribution (over A) of the output of this procedure is close with respect to Dist_∞ to the continuous distribution whose density is $\frac{F(u)}{\int\limits_{v\in A} F(v)dv},\ u \in A$. This is guaranteed by a proper choice of the cell size of the grid and by the fact that $\log F$ is $(\frac{n\tilde{\epsilon}}{\|\mathcal{C}\|_2} + \alpha)$- Lipschitz over A. Now, note that what we have so far is an efficient algorithm (let's denote it by $\mathcal{A}_{\mathsf{samp-A}}$) that outputs a sample from a distribution over A which close, with respect to Dist_∞, to the continuous distribution $\frac{F(u)}{\int\limits_{v\in A} F(v)dv},\ u \in A$. In fact, if our set $\mathcal{C}$ was a cube to begin with, then we would be already done. However, in general this is not the case and we need to do more. Namely, we construct an algorithm (which we denote by $\mathcal{A}_{\mathsf{eff-exp-samp}}$) that outputs a sample from a distribution that is close to the desired distribution over $\mathcal{C}$ by making black-box calls to $\mathcal{A}_{\mathsf{samp-A}}$ multiple times (say, at most k times where $k = poly(p, n)$) where fresh random coins are used by $\mathcal{A}_{\mathsf{samp-A}}$ at each time it is called. If, in any of such calls, $\mathcal{A}_{\mathsf{samp-A}}$ returns a sample $\theta \in \mathcal{C}$, then $\mathcal{A}_{\mathsf{eff-exp-samp}}$ terminates outputting $\theta^{priv} = \theta$. Otherwise, $\mathcal{A}_{\mathsf{eff-exp-samp}}$ outputs some random point $\theta^{priv} \in \mathcal{C}$. The use of the gauge function $\bar{\psi}_\alpha$ is, in fact, what makes this algorithm works for arbitrary

[3]The weight function is the function that induces the desired distribution over a given set.

bounded convex sets. By appropriately choosing the parameter α, we can ensure that, with high probability, the output θ^{priv} of $\mathcal{A}_{\mathsf{eff-exp-samp}}$ is drawn from a distribution (over $\mathcal{C}$) that is close, with respect to Dist_∞, to the desired continuous distribution $\frac{F(\theta^{priv})}{\int\limits_{\theta\in\mathcal{C}} F(\theta)d\theta}$.

Remark: In our efficient sampling algorithm, 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 running time required to perform such operations since this factor is highly dependent on the specific structure of the set $\mathcal{C}$.

We discuss the details of the implementation of our algorithm $\mathcal{A}_{\mathsf{eff-exp-samp}}$ and the proof of Theorem 3.4 in Section 6.

4 Localization and Optimal Private Algorithms for Strongly Convex Loss

It is unclear how to get a direct variant of Algorithm 2 in Section 3 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 mechanism of Section3.1 (Algorithm 2) to obtain an algorithm with optimal excess risk bound (Theorem 4.3).

Note: The localization technique is not specific to pure ϵ-differential privacy, and extends naturally to (ϵ, δ) case. Although it is not relevant in our current context, since we already have gradient descent based algorithm which achieves optimal excess risk bound. We defer the details for the (ϵ, δ) case to Appendix B.2.

Details of the generic algorithm: We first give a simple algorithm that carries out the desired localization step. The crux of the algorithm is the same as to that of the output perturbation algorithm of [6, 7]. The high-level idea is to first compute $\theta^* = \arg\min_{\theta\in\mathcal{C}} \mathcal{L}(\theta; \mathcal{D})$ and add noise according to the sensitivity of θ^*. The details of the algorithm are given in Algorithm 3.

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 and convex set $\mathcal{C}$.

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: $\theta_0 = \Pi_{\mathcal{C}}\left(\theta^* + b\right)$, 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 $\mathcal{C}_0 = \{\theta \in \mathcal{C} : \|\theta - \theta_0\|_2 \leq \zeta \frac{2Lp}{\Delta\epsilon n}\}$ for some $\zeta > 1$ (possibly a fixed function of p and 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 5 of Section 3.2 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 $\epsilon/2$ 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}.

Lemma 4.1 (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 [7]) and that $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ is $\frac{\epsilon}{2}$-differentially private by assumption. □

In the following theorem, 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 4.2 (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

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

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

Proof. The proof follows from the fact that, in Algorithm $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$, the norm of the noise vector $\|b\|_2$ is distributed according to Gamma distribution $\Gamma(p, \frac{4L}{\Delta\epsilon n})$ and hence satisfies

$$\Pr\left(\|b\|_2 \leq \zeta\frac{4Lp}{\Delta\epsilon n}\right) \geq 1 - e^{-\zeta}$$

(see, for example, [7]). Now, set $\zeta = 3\log(n)$. Hence, with probability $1 - \frac{1}{n^3}$, $\mathcal{C}_0$ (the output of $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$) contains θ^*. Hence, by setting $\tilde{\mathcal{C}}$ in the statement of the lemma to $\mathcal{C}_0$ (and noting that $\|\mathcal{C}_0\|_2 = O\left(\frac{Lp\log(n)}{\Delta\epsilon n}\right)$), then *conditioned on* the event that $\mathcal{C}_0$ contains θ^*, we have $\hat{\theta} = \theta^*$ and hence

$$E\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D}) \,|\theta^* \in \mathcal{C}_0\right] \leq F\left(p, n, \epsilon, L, O\left(\frac{Lp\log(n)}{\Delta\epsilon n}\right)\right)$$

Thus,

$$E\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] \leq F\left(p, n, \epsilon, L, O\left(\frac{Lp\log(n)}{\Delta\epsilon n}\right)\right)(1-\frac{1}{n^3}) + nL\|\mathcal{C}\|_2\frac{1}{n^3}$$

Note that the second term on the right-hand side above becomes $O(\frac{1}{n^2})$. From our lower bound (Section 5.2 below), $F(.,n,.,.,.)$ must be at least $\Omega(\frac{1}{n})$. Hence, we have

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

which completes the proof of the theorem. □

Instantiation of Algorithm $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ with the exponential sampling Algorithm 5 (See Section 3.2): 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 mechanism from Section 3.1 (Algorithm 2), or its efficient version Algorithm 5 (See Theorem 3.4 in Section 3.2) to obtain the optimal excess risk bound. We formally state the bound in Theorem 4.3) below. The proof of Theorem 4.3 follows from Theorem 3.4, Lemma 3.3, and Lemma 4.2 above. (See Appendix B.1 for details.)

Theorem 4.3 (Utility guarantee with Algorithm 5 as an instantiation of $\mathcal{A}_{\mathsf{gen-Lip}}$)**.** *Let $r\mathbb{B} \subseteq \mathcal{C} \subset \mathbb{R}^p$, where $\mathbb{B}$ is a p-dimensional ball of unit radius. Suppose we replace $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$ in Algorithm 4 with Algorithm 5 (See Theorem 3.4 and Section 6 below for details). Then, the output θ^{priv} satisfies*

$$E\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] = O\left(\frac{p^2L^2}{n\Delta\epsilon^2}\log(n)\log\left(\frac{\epsilon n\|\mathcal{C}\|_2}{r}\right)\right)$$

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

5 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. As before, for a dataset $\mathcal{D} = \{d_1, \ldots, d_n\}$, our decomposable loss function is expressed as $\mathcal{L}(\theta;\mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i),\ \theta \in \mathcal{C}$ for some convex set $\mathcal{C} \subset \mathbb{R}^p$. In Section 5.1, we consider the case of convex Lipschitz loss functions, whereas in Section 5.2, we consider the case of strongly convex and Lipschitz loss functions.

In our lower bounds, we consider the convex set $\mathcal{C}$ to be the p-dimensional unit ball $\mathbb{B}$. In our lower bounds for Lipschitz convex functions (Section 5.1), we consider a loss function ℓ that is 1-Lipschitz. On the other hand, in our lower bounds for Lipschitz and strongly convex functions (Section 5.2), we consider a loss function ℓ that is $\frac{\|C\|_2}{2}$-Lipschitz (where $\|C\|_2$=2 since $\mathcal{C} = \mathbb{B}$). Hence, without loss of generality, $L\|C\|_2 = 2$ in all our bounds. That is, for a general setting of $\|C\|_2$ and L, one can think of our lower bounds as being normalized (i.e., divided by) $L\|C\|_2$. Hence, one can see, given our results in Sections 2 and 3.1, that our lower bounds in Section 5.1 are tight (up to logarithmic factors).

In our lower bounds in Section 5.2, the loss function ℓ that we consider is 1-strongly convex (i.e., $\Delta = 1$). Although we can always set Δ to any arbitrary value by rescaling the loss function, such rescaling will also affect its Lipschitz constant L. This is due to the fact that our choice of the loss function $\ell(\theta; d)$ is the

squared L_2 distance between θ and d. Hence, one can easily show that for any arbitrary vlaues of $\|C\|_2, L$, and Δ, our lower bounds in Section 5.2 are a factor of $\frac{L}{\Delta\|C\|_2}$ smaller than the corresponding upper bounds in Sections 2 and 4 (ignoring the logarithmic factors in the upper bounds). Thus, if $\frac{L}{\Delta\|C\|_2} = O(1)$, then our lower bounds in Section 5.2 are tight (up to logarithmic factors).

5.1 Lower bounds for Lipschitz Convex Functions

In this section, we give lower bounds for both ϵ and (ϵ, δ) differentially private algorithms for minimizing any convex Lipschitz loss function $\mathcal{L}(\theta; \mathcal{D})$. We consider the following loss function. Define $\ell(\theta; d) = -\langle\theta, d\rangle,\ \theta \in \mathbb{B},\ d \in \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$. For any dataset $\mathcal{D} = \{d_1, \ldots, d_n\}$ with data points drawn from $\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, and any $\theta \in \mathbb{B}$, define $\mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^n \ell(\theta; d_i)$. Clearly, $\mathcal{L}$ is linear and, hence, Lipschitz and convex. Note that $\theta^* = \frac{\sum_{i=1}^n d_i}{\|\sum_{i=1}^n d_i\|_2}$ is the minimizer of $\mathcal{L}(.; \mathcal{D})$ over $\mathbb{B}$. Next, we show lower bounds on the excess risk incurred by any ϵ and (ϵ, δ) differentially private algorithm with output $\theta^{priv} \in \mathbb{B}$.

Before we state and prove our lower bounds, we first give the following useful lemma.

Lemma 5.1 (Lower bounds for 1-way marginals)**.** *The statements below follow from the results of [20] and [5], respectively.*

1. ***ϵ-differential private algorithms:*** *Let $\epsilon = O(1)$. There is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^n d_i\|_2 = \Omega\left(\min\left(n, p/\epsilon\right)\right)$ such that for any ϵ-differentially private algorithm (whose output is denoted by θ^{priv}) for answering 1-way marginals $q^{\mathcal{D}} \triangleq \frac{1}{n}\sum_{i=1}^n d_i$, there exists a subset $\mathcal{S} \subseteq [p]$ with $|\mathcal{S}| = \Omega(p)$ such that, with a positive constant probability (taken over the algorithm random coins), we have*

$$|\theta_j^{priv} - q_j^{\mathcal{D}}| \geq \Omega\left(\min\left(\frac{1}{\sqrt{p}}, \frac{\sqrt{p}}{\epsilon n}\right)\right)\ \forall j \in \mathcal{S}$$

where θ_j^{priv} and $q_j^{\mathcal{D}}$ denote the jth coordinate of θ^{priv} and $q^{\mathcal{D}}$, respectively.

2. ***(ϵ, δ)-differential private algorithms:*** *Let $\epsilon = O(1)$ and $\delta = o(\frac{1}{n})$. There is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ with $\|\sum_{i=1}^n d_i\|_2 = \Omega\left(\min\left(n, \sqrt{p}/\epsilon\right)\right)$ such that for any (ϵ, δ)-differentially private algorithm (whose output is denoted by θ^{priv}) for answering 1-way marginals $q^{\mathcal{D}} \triangleq \frac{1}{n}\sum_{i=1}^n d_i$, there exists a subset $\mathcal{S} \subseteq [p]$ with $|\mathcal{S}| = \Omega(p)$ such that, with a positive constant probability (taken over the algorithm random coins), we have*

$$|\theta_j^{priv} - q_j^{\mathcal{D}}| \geq \Omega\left(\min\left(\frac{1}{\sqrt{p}}, \frac{1}{\epsilon n}\right)\right)\ \forall j \in \mathcal{S}$$

where θ_j^{priv} and $q_j^{\mathcal{D}}$ denote the jth coordinate of θ^{priv} and $q^{\mathcal{D}}$, respectively.

Theorem 5.2 (Lower bound for ϵ-differentially private algorithms)**.** *Let $\epsilon = O(1)$. There is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ such that for any ϵ-differentially private algorithm (whose output is denoted by θ^{priv}), with positive constant ptobability, we must have*

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

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. First, observe that for any $\theta \in \mathbb{B}$ and any dataset $\mathcal{D}$, $\mathcal{L}(\theta;\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D}) = \|\sum_{i=1}^n d_i\|_2 \left(1 - \langle \theta, \theta^* \rangle\right)$. Define $\mathcal{E} = \frac{1}{\|\sum_{i=1}^n d_i\|_2}\left(\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right)$. It is easy to see that $\mathcal{E} \geq \frac{1}{2}\|\theta^{priv} - \theta^*\|_2^2$ since $\|\theta^{priv} - \theta^*\|_2^2 = \|\theta^*\|_2^2 + \|\theta^{priv}\|_2^2 - 2\langle \theta^{priv}, \theta^* \rangle$ and $\theta^*, \theta^{priv} \in \mathbb{B}$. Part 1 of Lemma 5.1 implies that there is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ and a set $\mathcal{S}$ with $|\mathcal{S}| = \Omega(p)$ such that, with positive constant probability, $|\theta_j^{priv} - \frac{\|\sum_{i=1}^n d_i\|_2}{n}\theta_j^*| \geq \Omega\left(\min\left(\frac{1}{\sqrt{p}}, \frac{\sqrt{p}}{\epsilon n}\right)\right)\ \forall j \in \mathcal{S}$ where $\|\sum_{i=1}^n d_i\|_2 = \Omega\left(\min(n, p/\epsilon)\right)$. Hence, we have (with positive constant probability) $|\theta_j^{priv} - \theta_j^*| \geq \Omega\left(\frac{1}{\sqrt{p}}\right)\ \forall j \in \mathcal{S}$. This implies that, with constant positive probability, $\|\theta^{priv} - \theta^*\|_2^2 \geq \Omega(1)$. Hence, from the observation we made above, we get

$$\left(\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right) = \Omega\left(\min\left(n, p/\epsilon\right)\right) \cdot \mathcal{E} \geq \Omega\left(\min\left(n, p/\epsilon\right)\right) \|\theta^{priv} - \theta^*\|_2^2 \geq \Omega\left(\min\left(n, p/\epsilon\right)\right).$$

□

Theorem 5.3 (Lower bound for (ϵ, δ)-differentially private algorithms)**.** *Let $\epsilon = O(1)$ and $\delta = o(\frac{1}{n})$. There is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ such that for any ϵ-differentially private algorithm (whose output is denoted by θ^{priv}), with positive constant ptobability, we must have*

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

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 5.1 and follow the same lines of the proof of Theorem 5.2. □

5.2 Lower bounds for Strongly Convex Functions

We give here lower bounds on the excess error of ϵ and (ϵ, δ) differentially private optimization algorithms for the class of strongly convex decomposable loss function $\mathcal{L}(\theta;\mathcal{D})$. Let $\ell(\theta; d)$ be half the squared L_2-distance between $\theta \in \mathbb{B}$ and $d \in \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, that is, $\ell(\theta; d) = \frac{1}{2}\|\theta - d\|_2^2$. Note that ℓ, as defined, is 1-Lipschitz and 1-strongly convex. For a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$, the decomposable loss function $\mathcal{L}(\theta;\mathcal{D}) = \sum_{i=1}^n \ell(\theta; d_i)$ is thus n-Lipschitz and n-strongly convex. Notice that the minimizer of $\mathcal{L}(.;\mathcal{D})$ over $\mathbb{B}$ is $\theta^* = \frac{1}{n}\sum d_i$ and that the excess error $\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*)$ can be written as $\frac{n}{2}\|\theta^{priv} - \frac{1}{n}\sum_{i=1}^n d_i\|_2^2$.

Theorem 5.4 (Lower bound for ϵ-differentially private algorithms)**.** *Let $\epsilon = O(1)$. Let $\theta^{priv} \in \mathbb{B}$ be the output of any ϵ-differentially private algorithm for minimizing $\mathcal{L}$ (as a function of θ) over $\mathbb{B}$. Then there exists a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ such that with constant positive probability over the algorithm random coins, we must have*

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

Proof. From Part 1 of Lemma 5.1, we know that there is a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ and a set $\mathcal{S} \subseteq [p]$ whose size $|\mathcal{S}| = \Omega(p)$ such that with a positive constant probability $|\theta_j^{priv} - \theta_j^*| \geq \Omega\left(\min\left(\frac{1}{\sqrt{p}}, \frac{\sqrt{p}}{\epsilon n}\right)\right)\ \forall j \in \mathcal{S}$. Hence, with a positive constant probability, we have

$$\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*) \geq \frac{n}{2}\sum_{j \in \mathcal{S}} |\theta_j^{priv} - \theta_j^*|^2 \geq \Omega\left(\min\left(n, \frac{p^2}{\epsilon^2 n}\right)\right).$$

□

Theorem 5.5 (Lower bound for (ϵ, δ)-differentially private algorithms)**.** *Let $\epsilon = O(1)$ and $\delta = o(\frac{1}{n})$. Let $\theta^{priv} \in \mathbb{B}$ be the output of any (ϵ, δ)-differentially private algorithm for minimizing $\mathcal{L}$ (as a function of θ) over $\mathbb{B}$. Then there exists a dataset $\mathcal{D} = \{d_1, \ldots, d_n\} \subseteq \{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\}^p$ such that with constant positive probability over the algorithm random coins, we must have*

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

Proof. We use Part 2 of Lemma 5.1 and follow the same lines of the proof of Theorem 5.4. □

6 Efficient Sampling from Logconcave Distributions over Convex Sets and The Proof of Theorem 3.4

In this section, we discuss the details of our efficient construction described in Section 3.2 and prove Theorem thm:eff-samp-guarantees which will involve the ability to carry out efficient sampling from logconcave distributions over arbitrary convex bounded sets. Towards the goal of proving Theorem 3.4, we start by giving the following lemma which describes Algorithm $\mathcal{A}_{\mathsf{samp}-\mathsf{A}}$ for sampling from a logconcave weight function F defined over a hypercube A.

Lemma 6.1. *Let $A \subset \mathbb{R}^p$ be a p-dimensional hypercube with edge length τ. Let F be a logconcave function that is strictly positive over A where $\log F$ is η-Lipschitz. Let μ_{A} be the probability measure induced by the density $\frac{F}{\int\limits_{u \in A} F(u)du}$. Let $\tilde{\epsilon} > 0$. There is an algorithm $\mathcal{A}_{\mathsf{samp}-\mathsf{A}}$ that takes A, F, and $\tilde{\epsilon}$ as inputs, and outputs a sample $\widetilde{\theta} \in A$ that is drawn from a continuous distribution $\hat{\mu}_{\mathsf{A}}$ over A with the property that $\mathsf{Dist}_\infty(\hat{\mu}_{\mathsf{A}}, \mu_{\mathsf{A}}) \leq \tilde{\epsilon}$. Moreover, the running time of $\mathcal{A}_{\mathsf{samp}-\mathsf{A}}$ is*

$$O\left(\frac{\eta^2\tau^2}{\tilde{\epsilon}^2} p^3 \max\left(p \log\left(\frac{\eta\tau p}{\tilde{\epsilon}}\right), \eta\tau\right)\right).$$

Proof. Let $\gamma = \frac{\tilde{\epsilon}}{2\eta\sqrt{p}}$. We construct a grid $\mathcal{G}_\gamma \triangleq \{u \in \mathbb{R}^p : \ u_j + \frac{\gamma}{2}$ is integer multiple of $\gamma, 1 \leq j \leq p\}$. Next, we run the grid-walk algorithm of [2] with the logconcave weight function F on $A \cap \mathcal{G}_\gamma$. It follows from the results of [2] that (i) the grid-walk is a lazy, time-reversible Markov chain, (ii) the stationary distribution of such grid-walk is $\pi = \frac{F}{\sum_{u \in A \cap \mathcal{G}_\gamma} F(u)}$, and (iii) the grid-walk has conductance $\phi \geq \frac{\tilde{\epsilon}}{8\eta\tau p^{\frac{3}{2}} e^{\frac{\tilde{\epsilon}}{2}}}$. We run the grid-walk for t_∞ steps (namely, the L_∞ mixing time of the walk) and output a sample $\hat{u} \in A \cap \mathcal{G}_\gamma$. Then, we uniformly sample a point θ from the grid cell whose center is $\hat{u}$. Let $\hat{\pi}$ denote the distribution of the output $\hat{u}$ of the grid-walk after t_∞ steps. Let $\hat{\mu}_{\mathsf{A}}$, as in the statement of the lemma, denote the distribution of θ that is uniformly sampled from the grid cell whose center is $\hat{u}$. Now, suppose that after t_∞ steps it is guaranteed to have $\mathsf{Dist}_\infty(\hat{\pi}, \pi) \leq \frac{\tilde{\epsilon}}{2}$. Then, since $\log F$ is η-Lipschitz and $\gamma = \frac{\tilde{\epsilon}}{2\eta\sqrt{p}}$ (where, as defined above, γ is the edge length of every cell of $\mathcal{G}_\gamma$), it is easy to show that $\mathsf{Dist}_\infty(\hat{\mu}_{\mathsf{A}}, \mu_{\mathsf{A}}) \leq \tilde{\epsilon}$. Hence, it remains to show a bound on t_∞, the L_∞ mixing time of the Markov chain given by the grid-walk. Specifically, t_∞ is the number of the steps on the grid-walk required to have $\mathsf{Dist}_\infty(\hat{\pi}, \pi) \leq \frac{\tilde{\epsilon}}{2}$. Towards this end, we use the result of [32] on the rapid mixing of lazy Markov chains with countable state space. We formally restate this result in the following lemma.

Lemma 6.2 (Theorem 1 in [32]). *Let P be a lazy, time reversible Markov chain over a countable state space Γ. Then, the time t_∞ reuired for relative L_∞ convergence of ϵ' is at most $1 + \int_{4\pi^*}^{4/\epsilon'} \frac{4dx}{x\Phi^2(x)}$. Here, $\Phi(x) = \inf\{\phi_S : \pi(S) \leq x\}$ where ϕ_S denotes the conductance of the set $S \subseteq \Gamma$ and π^* is the minimum probability assigned by the stationary distribution.*

Now, setting $\epsilon' = \frac{\tilde{\epsilon}}{2}$ in the above lemma and using the fact that $\Phi(x) \geq \phi \geq \frac{\tilde{\epsilon}}{8\eta\tau p^{\frac{3}{2}} e^{\frac{\tilde{\epsilon}}{2}}}$ for all x, we get

$$t_\infty = O\left(\frac{\eta^2\tau^2 p^3}{\tilde{\epsilon}^2} \log\left(\frac{1}{\tilde{\epsilon}\pi^*}\right)\right).$$

Observe that

$$\pi(u) = \frac{F(u)}{\sum_{v \in A \cap \mathcal{G}_\gamma} F(v)} \geq \frac{e^{-\eta\tau}}{\left(\frac{\tau}{\gamma}\right)^p}$$

where the last inequality follows from the fact that $\log F$ is η-Lipschitz. Plugging the value we set for γ, we get $t_\infty = O\left(\frac{\eta^2\tau^2}{\tilde{\epsilon}^2} p^3 \max\left(p \log\left(\frac{\eta\tau p}{\tilde{\epsilon}}\right), \eta\tau\right)\right)$. This completes the proof. □

Having Lemma 6.1 in hand, we now discuss our construction of Algorithm $\mathcal{A}_{\mathsf{eff-exp-samp}}$ (Algorithm 5 below) that we informally described in Section 3.2. Our algorithm starts by finding a cube A that contains $\mathcal{C}$ and then, with an overwhelming fixed probability (in n), it runs $\mathcal{A}_{\mathsf{samp-A}}$ multiple times with such A as an input as will be described shortly. With the remainder probability (which is negligible in n), $\mathcal{A}_{\mathsf{eff-exp-samp}}$ just outputs a random point in $\mathcal{C}$ and aborts.

Remark: Although we give our construction for the specific case of F above, *one can easily generalize this construction to any logconcave weight function F using the same gauge-function trick.*

Fix a dataset $\mathcal{D}$ of size n. In the remainder of this section, we set the logconcave weight function F in Lemma 6.1 to be $e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\theta;\mathcal{D}) - \bar{\psi}_\alpha(\theta)}$ where $\tilde{\epsilon} = \frac{\epsilon}{20}$, $\bar{\mathcal{L}}(.;\mathcal{D})$ is a convex Lipschitz extension of $\mathcal{L}(.;\mathcal{D})$ to the cube A, and $\bar{\psi}_\alpha$ is the gauge function described earlier. Note that $\mathcal{L}(.;\mathcal{D})$ may not be defined outside $\mathcal{C}$ and thus we define an extension for it over the set $A \setminus \mathcal{C}$ such that the new function remains convex and nL-Lipschitz over A. McShane-Whitney extension theorem [21] gives an explicit construction of an extension for any Lipschitz function defined over an arbitrary set to a Lipschitz function (with the same Lipschitz constant) defined over $\mathbb{R}^p$. The following lemma asserts that if the original Lipschitz function is also convex and defined over a convex set $\mathcal{C}$, then the same construction of McShane-Whitney (i) is efficiently computable and (ii) yields a function that is Lipschitz and also convex.

Lemma 6.3 (Convex Lipschitz extension). *Let f be an efficiently computable, η-Lipschitz, convex function defined on a convex bounded set $\mathcal{C} \subset \mathbb{R}^p$. Then there exists an efficiently computable, η-Lipschitz convex function F defined over $\mathbb{R}^p$ such that F, restricted to $\mathcal{C}$, is equal to f. The efficient computation of F is based on the assumption of the existence of a projection oracle.*

Proof. For the sake of simplicity, let's assume that $\mathcal{C}$ is closed. Actually, this is no loss of generality since we can always redefine f such that it is defined on the closure of $\mathcal{C}$ which is possible because f is continuous on $\mathcal{C}$. We use the same extension in the proof McShane-Whitney theorem [21]. Namely, define

$$g_y(x) \triangleq f(y) + \eta\|x - y\|_2,\ y \in \mathcal{C}, x \in \mathbb{R}^p$$

$$F(x) = \min_{y \in \mathcal{C}} g_y(x), x \in \mathbb{R}^p.$$

By McShane-Whitney theorem, we know that the function F on $\mathbb{R}^p$ is η-Lipschitz extension of f. Moreover, since f is convex and $\mathcal{C}$ is a convex set, then for every $x \in \mathbb{R}^p$, the computation of $F(x)$ is a convex program which can beimplemented efficiently using a linear optimization oracle. In particular, a projection oracle would suffice and hence F is efficiently computable. It remains to show that F is convex. Let $x_1, x_2 \in \mathbb{R}^p$. Let y_1 and y_2 denote the minimizers of $g_y(x_1)$ and $g_y(x_2)$ over $y \in \mathcal{C}$, respectively. Let $0 \leq \lambda \leq 1$. Define $x_\lambda = \lambda x_1 + (1-\lambda)x_2$ and let y_λ denote the minimzer of $g_y(x_\lambda)$ over $y \in \mathcal{C}$. Now, observe that

$$\begin{aligned} F(x_\lambda) &= g_{y_\lambda}(x_\lambda) \leq g_{\lambda y_1 + (1-\lambda)y_2}(x_\lambda) \\ &= f(\lambda y_1 + (1-\lambda)y_2) + \eta \|\lambda(y_1 - x_1) + (1-\lambda).(y_2 - x_2)\|_2 \\ &\leq \lambda \left(f(y_1) + \eta\|y_1 - x_1\|_2\right) + (1-\lambda)\left(f(y_2) + \eta\|y_2 - x_2\|_2\right) \\ &= \lambda F(x_1) + (1-\lambda)F(x_2) \end{aligned}$$

where the inequality in the first line follows from the fact that y_λ is the minimzer (w.r.t. y) of $g_y(x_\lambda)$ and the inequality in the third line follows from the convexity of f and the L_2-norm. This completes the proof of the lemma. □

Now, we construct the extension of our loss $\mathcal{L}(.;\mathcal{D})$ over the cube A in the same fashion described in Lemma 6.3. Namely, we define our Lipscitz extension $\bar{\mathcal{L}}(.;\mathcal{D})$ as

$$\bar{\mathcal{L}}(\theta;\mathcal{D}) = \min_{u \in \mathcal{C}} \left(\mathcal{L}(u;\mathcal{D}) + nL\|\theta - u\|_2\right),\ \theta \in A.$$

By Lemma 6.3, for every $\theta \in A$, $\bar{\mathcal{L}}(\theta;\mathcal{D})$ is efficiently computable, convex, and nL-Lipschitz.

Algorithm 5 $\mathcal{A}_{\mathsf{eff-exp-samp}}$: Efficient Log-Concave Sampling over Convex Set $\mathcal{C}$

Input: data set $\mathcal{D}$ of size n, convex set $\mathcal{C}$, loss function ℓ, Lipschitz constant L of ℓ, privacy parameter ϵ.

1: Find a cube $A \supseteq \mathcal{C}$ with edge length $\tau = \|\mathcal{C}\|_\infty$.
2: $\mathcal{L}(\theta;\mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i)$.
3: Find a convex Lipschitz extension $\bar{\mathcal{L}}(\theta;\mathcal{D}) = \min_{u \in \mathcal{C}} \left(\mathcal{L}(u;\mathcal{D}) + nL\|\theta - u\|_2\right)$.
4: $\bar{\psi}_\alpha(\theta) = \alpha \cdot \max\{0, \psi(\theta) - 1\}$ with $\alpha = \frac{3}{20} e^{\frac{2\epsilon}{5}} (\epsilon n + p)$, where $\psi(\theta)$ is the Minkowski's norm of θ w.r.t. $\mathcal{C}$ as defined above.
5: $F(\theta) = e^{-\frac{\epsilon}{20L\|\mathcal{C}\|_2} \bar{\mathcal{L}}(\theta;\mathcal{D}) - \bar{\psi}_\alpha(\theta)}$.
6: With probability $\frac{1}{\epsilon 2^n}$: Output $\theta^{priv} = \theta_0 \in \mathcal{C}$ and **abort** (where θ_0 is an arbitrary fixed[4] point in $\mathcal{C}$); **otherwise**, continue.
7: **for** $1 \leq i \leq n$ **do**
8: $\quad \theta_i = \mathcal{A}_{\mathsf{samp-A}}(A, F, \frac{\epsilon}{5})$.
9: $\quad$ **if** $\theta_i \in \mathcal{C}$ **then**
10: $\quad\quad$ Output $\theta^{priv} = \theta_i$ and **abort**.
11: Output $\theta^{priv} = \theta_0$.

Now, we analyze our algorithm and prove Theorem 3.4. Let Bad denote the event that $\mathcal{A}_{\mathsf{eff-exp-samp}}$ does not abort during the **for** loop, i.e., Bad is the event that $\mathcal{A}_{\mathsf{eff-exp-samp}}$ outputs an arbitrary fixed data-independent point $\theta_0 \in \mathcal{C}$ which happens either in step 6 or in step 11. Let Good denote the complement of Bad, that is, the event that one of the n calls to $\mathcal{A}_{\mathsf{samp-A}}$ returns a sample θ_i inside $\mathcal{C}$. The next lemma bounds the probability of Bad and characterizes the distribution of θ^{priv} (the output of $\mathcal{A}_{\mathsf{eff-exp-samp}}$) *conditioned on* the event Good.

Lemma 6.4. *The probability of the event* Bad *defined above satisfies* $\Pr[\mathsf{Bad}] \in [\frac{1}{\epsilon 2^n}, \frac{1}{\epsilon 2^n} + \frac{1}{2^n})$. *Let* $\hat{\mu}_{\mathsf{Good}}$ *denote the conditional distribution of* θ^{priv} *(the output of* $\mathcal{A}_{\mathsf{eff-exp-samp}}$*) conditioned on the event* Good. *Then, we have* $\mathsf{Dist}_\infty\left(\hat{\mu}_{\mathsf{Good}}, \frac{F}{\int_{u\in\mathcal{C}} F(u)du}\right) \leq \frac{2\epsilon}{5}$.

Proof. For simplicity of notation, since the dataset $\mathcal{D}$ is fixed, we will drop $\mathcal{D}$ from $\bar{\mathcal{L}}(\theta;\mathcal{D})$ and denote it by just $\bar{\mathcal{L}}(\theta)$. We start by proving that $\Pr[\mathsf{Bad}] \in [\frac{1}{\epsilon 2^n}, \frac{1}{\epsilon 2^n} + \frac{1}{2^n})$. To do this, it suffices to prove the following claim.

Claim 6.5. *For the given function* F *and the given setting of* α, $\mathcal{A}_{\mathsf{samp-A}}$ *outputs* $\theta \notin \mathcal{C}$ *with probability at most* $\frac{1}{2}$.

Proof. By Lemma 6.1, we know that θ (the output of $\mathcal{A}_{\mathsf{samp-A}}$) has a distribution $\hat{\mu}_{\mathsf{A}}$ with the property that $\mathsf{Dist}_\infty(\hat{\mu}_{\mathsf{A}}, \mu_{\mathsf{A}}) \leq \frac{\epsilon}{5}$ where $\mu_{\mathsf{A}}(u) = \frac{F(u)}{\int_{v\in A} F(v)dv}$, $u \in A$. We will show that $\int_{\theta\in A\setminus\mathcal{C}} \hat{\mu}_{\mathsf{A}}(\theta)d\theta \leq \int_{\theta\in\mathcal{C}} \hat{\mu}_{\mathsf{A}}(\theta)d\theta$. In particular, it suffices to show that $\int_{\theta\in A\setminus\mathcal{C}} F(\theta)d\theta \leq e^{-\frac{2\epsilon}{5}} \int_{\theta\in\mathcal{C}} F(\theta)d\theta$. Towards this end, consider a differential (p-dimensional) cone with a differential angle $d\omega$ at its vertex which is located at the origin (i.e., inside $\mathcal{C}$ since $\mathcal{C}$ is in isotropic position). Let $\widetilde{\theta}$ be the point where the axis of the cone intersects with the boundary of $\mathcal{C}$. The set $\mathcal{C}$ divides the cone into two regions; one inside $\mathcal{C}$ and the other is outside $\mathcal{C}$. We now show that, for any such cone, the integral of F over its region outside $\mathcal{C}$, denoted by $\mathcal{I}_{\mathsf{out}}$, is less than the integral of $e^{-\frac{2\epsilon}{5}}F$ over the region inside $\mathcal{C}$ which is denoted by $\mathcal{I}_{\mathsf{in}}$. Let $\tilde{\epsilon} = \frac{\epsilon}{20}$. First, observe that

$$\begin{aligned}
\mathcal{I}_{\mathsf{in}} &= d\omega\, p\|\widetilde{\theta}\|_2^p \int_0^1 e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(r\widetilde{\theta})} r^{p-1}dr \geq d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \int_0^1 e^{-\tilde{\epsilon}n(1-r)} r^{p-1}dr \\
&= d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \int_0^1 e^{-\tilde{\epsilon}nr}(1-r)^{p-1}dr \geq d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \int_0^{\frac{1}{\tilde{\epsilon}n+p}} (1-\tilde{\epsilon}nr)(1-pr)dr \\
&\geq d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \int_0^{\frac{1}{\tilde{\epsilon}n+p}} (1-(\tilde{\epsilon}n+p)\,r)\,dr = d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \frac{1}{2(\tilde{\epsilon}n+p)}
\end{aligned}$$

where the second inequality in the first line follows from the Lipschitz property of $\bar{\mathcal{L}}$ and the second inequality in the second line follows from the fact that $e^{-x} \geq 1-x$ and $(1-x)^{p-1} \geq 1-px$. On the other hand, we can upper bound $\mathcal{I}_{\mathsf{out}}$ as follows.

$$\begin{aligned}
\mathcal{I}_{\mathsf{out}} &\leq d\omega\, p\|\widetilde{\theta}\|_2^p \int_1^\infty e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(r\widetilde{\theta})} e^{-\alpha(r-1)} r^{p-1}dr \leq d\omega\, p\|\widetilde{\theta}\|_2^p e^{-\frac{\tilde{\epsilon}}{L\|\mathcal{C}\|_2}\bar{\mathcal{L}}(\widetilde{\theta})} \int_1^\infty e^{\tilde{\epsilon}n(r-1)} e^{-\alpha(r-1)} r^{p-1}dr \\
&\leq 2(\tilde{\epsilon}n+p)\mathcal{I}_{\mathsf{in}} \int_0^\infty e^{-(\alpha-(\tilde{\epsilon}n+p))r} \leq e^{-\frac{2\epsilon}{5}}\mathcal{I}_{\mathsf{in}} \qquad (3)
\end{aligned}$$

where the last inequality follows from the setting of α we made in Algorithm 5. Since this is true for any differential cone as described above, the proof of the claim is now complete. □

The previous claim together with the definition of the event Bad above concludes the proof of the first part of Lemma 6.4.

Next, let $\mu_{\mathcal{C}}$ denote the distribution induced by F on $\mathcal{C}$, that is, $\frac{F}{\int_{\theta\in\mathcal{C}} F(\theta)d\theta}$. We show that the conditional distribution $\hat{\mu}_{\mathsf{Good}}$ of θ^{priv} (the output of $\mathcal{A}_{\mathsf{eff-exp-samp}}$) conditioned on the event Good (the complement of

Bad) satisfies $\mathsf{Dist}_\infty(\hat{\mu}_{\mathsf{Good}}, \mu_{\mathcal{C}}) \leq \frac{2\epsilon}{5}$. Suppose that the event Good occurs. Let $i^* \in [n]$ denote the iteration in which $\mathcal{A}_{\mathsf{samp-A}}$ outputs a sample $\theta_{i^*} \in \mathcal{C}$. Note that i^* is a random variable. Moreover, observe that

$$\mathsf{Good} \iff i^* \in [n] \iff i^* \in [n] \wedge \theta_{i^*} \in \mathcal{C} \iff i^* \in [n] \wedge \theta_{i^*} \in \mathcal{C} \wedge \theta^{priv} = \theta_{i^*}$$

Thus, for any measurable subset $\mathcal{U} \subseteq \mathcal{C}$, we have

$$\begin{aligned}
\hat{\mu}_{\mathsf{Good}}(\mathcal{U}) &= \Pr\left[\theta^{priv} \in \mathcal{U} \mid \mathsf{Good}\right] = \Pr\left[\theta_{i^*} \in \mathcal{U} \mid \theta_{i^*} \in \mathcal{C} \wedge i^* \in [n]\right] \\
&= \sum_{j=1}^{n} \Pr\left[\theta_{i^*} \in \mathcal{U} \mid \theta_{i^*} \in \mathcal{C} \wedge i^* = j\right] \Pr[i^* = j | i^* \in [n]] \\
&= \sum_{j=1}^{n} \Pr\left[\theta_j \in \mathcal{U} \mid \theta_j \in \mathcal{C}\right] \Pr[i^* = j | i^* \in [n]] \\
&= \frac{\hat{\mu}_{\mathsf{A}}(\mathcal{U})}{\hat{\mu}_{\mathsf{A}}(\mathcal{C})} \sum_{j=1}^{n} \Pr[i^* = j | i^* \in [n]] = \frac{\hat{\mu}_{\mathsf{A}}(\mathcal{U})}{\hat{\mu}_{\mathsf{A}}(\mathcal{C})}
\end{aligned}$$

Now, since $\mu_{\mathcal{C}}(\mathcal{U}) = \frac{\mu_{\mathsf{A}}(\mathcal{U})}{\mu_{\mathsf{A}}(\mathcal{C})}$, by Lemma 6.1, we have $\mathsf{Dist}_\infty\left(\frac{\hat{\mu}_{\mathsf{A}}}{\hat{\mu}_{\mathsf{A}}(\mathcal{C})}, \mu_{\mathcal{C}}\right) \leq \frac{2\epsilon}{5}$ which completes the proof of the second part of the lemma. □

Lemma 6.4 constitutes the central part of the proof of Theorem 3.4. The following lemma (from [20]) will be useful in finalizing the proof.

Lemma 6.6 (follows from Lemma A.1 in [20])**.** *Let $\epsilon, \tilde{\epsilon} > 0$. Let $\mathcal{Q} \subseteq \mathbb{R}^p$. For every dataset $\mathcal{D}$, let $\mu^{\mathcal{D}}$ denote the distribution (over $\mathcal{Q}$) of the output of an ϵ-differentially private algorithm $\mathcal{A}_1$ when run on the input dataset $\mathcal{D}$, and $\hat{\mu}^{\mathcal{D}}$ be the distribution (over $\mathcal{Q}$) of the output of some algorithm $\mathcal{A}_2$ when run on $\mathcal{D}$. Suppose that $\mathsf{Dist}_\infty(\hat{\mu}^{\mathcal{D}}, \mu^{\mathcal{D}}) \leq \tilde{\epsilon}$ for all $\mathcal{D}$. Then, $\mathcal{A}_2$ is $(2\tilde{\epsilon} + \epsilon)$-differentially private.*

Proof of Theorem 3.4: We start by proving differential privacy of Algorithm 5. Fix any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Let $\hat{\mu}^{\mathcal{D}}$ and $\hat{\mu}^{\mathcal{D}'}$ be the distributions of the output θ^{priv} of Algorithm 5 when the input datasets are $\mathcal{D}$ and $\mathcal{D}'$, respectively. Let $\mathsf{Good}^{\mathcal{D}}, \mathsf{Good}^{\mathcal{D}'}$ be the events analogous to the event Good of Lemma 6.4 when the input datasets are $\mathcal{D}$ and $\mathcal{D}'$, respectively. Similarly, we let $\mathsf{Bad}^{\mathcal{D}}$ and $\mathsf{Bad}^{\mathcal{D}'}$ be the events corresponding to Bad of Lemma 6.4. We denote the conditional distributions of the output θ^{priv} conditioned on the event $\mathsf{Good}^{\mathcal{D}}$ and $\mathsf{Good}^{\mathcal{D}'}$ by $\hat{\mu}^{\mathcal{D}}_{\mathsf{Good}}$ and $\hat{\mu}^{\mathcal{D}'}_{\mathsf{Good}}$, respectively. Note that, on the other hand, the conditional distribution of θ^{priv} conditioned on the Bad event of Lemma 6.4 is the same on $\mathcal{C}$ irrespective of the input dataset (namely, it is a degenrate distribution whose mass is located at $\theta^{priv} = \theta_0$). Let's denote it by $\hat{\mu}_{\mathsf{Bad}}$. Now, observe that for any $\theta^{priv} \in \mathcal{C}$

$$d\hat{\mu}^{\mathcal{D}}(\theta) = d\hat{\mu}^{\mathcal{D}}_{\mathsf{Good}} \Pr[\mathsf{Good}^{\mathcal{D}}] + \mathsf{d}\hat{\mu}_{\mathsf{Bad}} \Pr[\mathsf{Bad}^{\mathcal{D}}]$$

It follows from the first part of Lemma 6.4 that

$$e^{-\epsilon} \leq (1+\epsilon)^{-1} \leq \frac{\Pr[\mathsf{Bad}^{\mathcal{D}}]}{\Pr[\mathsf{Bad}^{\mathcal{D}'}]} \leq 1 + \epsilon \leq e^{\epsilon}$$

Thus, we also have

$$e^{-\frac{\epsilon}{10}} \leq 1 - \frac{\epsilon \Pr[\mathsf{Bad}^{\mathcal{D}}]}{1 - (1-\epsilon)\Pr[\mathsf{Bad}^{\mathcal{D}}]} \leq \frac{\Pr[\mathsf{Good}^{\mathcal{D}}]}{\Pr[\mathsf{Good}^{\mathcal{D}'}]} \leq \frac{\epsilon \Pr[\mathsf{Bad}^{\mathcal{D}'}]}{1 - \Pr[\mathsf{Bad}^{\mathcal{D}'}]} \leq e^{\frac{\epsilon}{10}}$$

where the first and last inequalities follow from the fact that $\Pr[\mathsf{Bad}^{\mathcal{D}}]$ (resp., $\Pr[\mathsf{Bad}^{\mathcal{D}'}]$) can be made sufficiently small. Moreover, from the second part of Lemma 6.4 and Lemma 6.6, it follows that $\mathsf{Dist}_\infty(\hat{\mu}^{\mathcal{D}}_{\mathsf{Good}}, \hat{\mu}^{\mathcal{D}'}_{\mathsf{Good}}) \leq \frac{9\epsilon}{10}$. Putting these together, we get $\mathsf{Dist}_\infty(\hat{\mu}^{\mathcal{D}}, \hat{\mu}^{\mathcal{D}'}) \leq \epsilon$. Hence, Algorithm 5 is ϵ-differentially private.

To show the utility guarantee of Algorithm 5, we first observe that except for an event that occurs with negligible probability, namely the event Bad of Lemma 6.4, the output θ^{priv} has distribution that is close with respect to Dist_∞ to (i.e., within a constant factor of) the distribution of the output of Algorithm 2, and hence, the utility analysis follows the same lines of Theorem 3.2.

Finally, observe that the running time is at most $O\left(nT_{\mathcal{A}_{\mathsf{samp}-\mathsf{A}}}\right)$ where $T_{\mathcal{A}_{\mathsf{samp}-\mathsf{A}}}$ is the running time of $\mathcal{A}_{\mathsf{samp}-\mathsf{A}}$. The running time of $\mathcal{A}_{\mathsf{samp}-\mathsf{A}}$ is given by Lemma 6.1 after substituting η with $\frac{\epsilon n}{20\|\mathcal{C}\|_2} + \alpha = O(\epsilon n)$ (where $\alpha = O(\epsilon n)$[5] is the gauge function parameter in Algorithm 5) and τ with $\|\mathcal{C}\|_2$. This completes the proof of Theorem 3.4.

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 funded by NSF awards #0747294 and #0941553. A.T. was funded in part by the Sloan Foundation.

References

[1] Alekh Agarwal, Peter L. Bartlett, Pradeep D. Ravikumar, and Martin 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] David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *STOC*. ACM, 1991.

[3] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: The SuLQ framework. In *PODS*, pages 128–138. ACM, 2005.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[5] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, 2014.

[6] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*. MIT Press, 2008.

[7] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *JMLR*, 12:1069–1109, 2011.

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

[5] Note that we use here the assumption that $n = \omega(p)$.

[9] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210. ACM, 2003.

[10] Irit Dinur, Cynthia Dwork, and Kobbi Nissim. Revealing information while preserving privacy, full version of [9], in preparation, 2010.

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

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

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

[14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[16] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, 2010.

[17] Martin E. Dyer, Alan M. Frieze, and Ravi Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[18] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[19] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, 2010.

[20] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. 2010.

[21] Juha Heinonen. Lectures on lipschitz analysis. *Lecture notes*, 2005.

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

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

[24] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24.1–24.34, 2012.

[25] Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In Sanjeev Khanna, editor, *SODA*, pages 1395–1414. SIAM, 2013. ISBN 978-1-61197-251-1, 978-1-61197-310-5.

[26] Shiva Prasad Kasiviswanathan and Adam Smith. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*, arXiv:0803.39461 [cs.CR], 2008.

[27] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *FOCS*, 2008.

[28] Shiva Prasad Kasiviswanathan, Mark Rudelson, and Adam Smith. The power of linear reconstruction attacks. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.

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

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

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

[32] Ben Morris and Yuval Peres. Evolving sets, mixing and heat kernel bounds. *Probability Theory and Related Fields*, 2005.

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

[34] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: The sparse and approximate cases. In *STOC*, 2013.

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

[36] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 71–79, 2013.

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

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

[39] S. Song, K. Chaudhuri, and A.D. Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 Global Conference on Signal and Information Processing (GlobalSIP 2013)*, pages 245–248, December 2013. doi: 10.1109/GlobalSIP.2013.6736861.

[40] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing*, 2013.

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

A Inefficient Exponential Mechanism for Arbitrary Convex Bodies

In Algorithm 6 below we provide a computationally inefficient procedure to achieve an error of $\tilde{O}(p/\epsilon)$ for arbitrary convex sets. We only provide utility analysis for this algorithm, since the proof that this algorithm is ϵ-differentially private follows directly from the analysis of Theorem 3.1.

Algorithm 6 $\mathcal{A}_{\mathsf{net-samp}}$: Convex optimization via sampling from a γ-net

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

1: Define a net $\mathcal{M}$ that covers $\mathcal{C}$ with balls of radius $\frac{\|\mathcal{C}\|_2 p}{\epsilon n}$ and with $\Theta\left((\frac{\epsilon n}{p})^p\right)$ net points

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

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

Theorem A.1 (Utility guarantee)**.** *For θ^{priv} output by $\mathcal{A}_{\mathsf{net-samp}}$ (Algorithm 6) we have the following. (The expectation is over the randomness of the algorithm.)*

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

Proof. First, since ℓ is L-Lipschitz, $\mathcal{L}$ is nL-Lipschitz. Thus, there exist a net point $\hat{\theta}$ such that $\mathcal{L}(\hat{\theta};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D}) = nL \cdot \frac{\|\mathcal{C}\|_2 p}{\epsilon n} = L\|\mathcal{C}\|_2 p/\epsilon$. A standard analysis of the exponential mechanism (McSherry and Talwar [31]) shows that the additional expected loss due to sampling is at most $O(\frac{1}{\epsilon})$ times the sensitivity of the loss function (here at most $L\|\mathcal{C}\|_2$) times the logarithm of the size of the set being sampled from (here $O(\frac{\epsilon n}{p})^p)$). Thus the final expected excess risk is $\frac{L\|\mathcal{C}\|_2 p}{\epsilon} + O(\frac{L\|\mathcal{C}\|_2}{\epsilon} \cdot \log((\frac{\epsilon n}{p})^p)) = O(\frac{L\|\mathcal{C}\|_2 p}{\epsilon} \cdot \log(\frac{\epsilon n}{p}))$. □

B Missing Details from Section 4 (Localization and Exponential Mechanism)

B.1 Proof of Theorem 4.3

Proof. With Theorem 4.2 in hand, it suffices to show that Algorithm 2 (from Section 3.1) , when given the convex set $\mathcal{C}_0$ (the output of $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$) as input, it yields an output $\widetilde{\theta}$ that satisfies

$$E\left[\mathcal{L}(\widetilde{\theta};\mathcal{D}) - \mathcal{L}(\hat{\theta};\mathcal{D})\right] \leq O\left(\frac{pL\|\mathcal{C}_0\|_2}{\epsilon}\log\left(\frac{\epsilon n\|\mathcal{C}\|_2}{r}\right)\right)$$

where $\hat{\theta} = \arg\min_{\theta\in\mathcal{C}_0} \mathcal{L}(\theta;\mathcal{D})$. The rest of the proof will follow directly from Theorem 4.2.

By Lemma 3.3, there exists a ball $\mathcal{G} \subseteq \mathcal{C}$ of radius $r_0 = \frac{r}{\|\mathcal{C}\|_2}\|\mathcal{C}_0\|_2 \leq \frac{r}{\|\mathcal{C}\|_2}\frac{4\zeta Lp}{\Delta\epsilon n}$ such that for all $\theta \in \mathcal{G}$, we have $\|\theta - \theta_0\|_2 \leq \frac{4\zeta Lp}{\Delta\epsilon n}$ where $\theta_0 \in \mathcal{C}_0$ is the perturbed minimizer generated by Algorithm $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$. Hence, by definition of $\mathcal{C}_0$, we must have $\mathcal{G} \subseteq \mathcal{C}_0$. Thus, Algorithm 2 (from Section 3.1) is given a convex

set $\mathcal{C}_0$ that contains a ball of radius r_0. Since $\hat{\theta}$ is the minimizer of $\mathcal{L}(.;\mathcal{D})$ in $\mathcal{C}_0$, then Theorem 3.4 implies that Algorithm 5 (from Section 6) outputs $\widetilde{\theta}$ that satisfies

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\widetilde{\theta};\mathcal{D}) - \mathcal{L}(\hat{\theta};\mathcal{D})\right] &\leq O\left(\frac{pL\|\mathcal{C}_0\|_2}{\epsilon}\log\left(\frac{\epsilon n\|\mathcal{C}_0\|_2}{r_0}\right)\right)\\ &= O\left(\frac{pL\|\mathcal{C}_0\|_2}{\epsilon}\log\left(\frac{\epsilon n\|\mathcal{C}\|_2}{r}\right)\right)\end{aligned}$$

as desired. □

B.2 Localization and (ϵ, δ)-Differentially Private Algorithms for Lipschitz, Strongly Convex Loss

We use slightly different version of $\mathcal{A}^{\epsilon}_{\mathsf{out-pert}}$ (Algorithm 3) which we denote by $\mathcal{A}^{(\epsilon,\delta)}_{\mathsf{out-pert}}$ where the algorithm takes as input an extra privacy parameter δ, it samples the noise vector b from the Gaussian distribution $\mathcal{N}\left(0, \mathbb{I}_p\sigma_0^2\right)$ where $\sigma_0^2 = 4\frac{L^2\log\left(\frac{1}{\delta}\right)}{\Delta^2\epsilon^2n^2}$, and outputs $\mathcal{C}_0 = \{\theta \in \mathcal{C} : \|\theta - \theta_0\|_2 \leq \zeta\sigma_0\sqrt{p}\}$.

Let $\mathcal{A}^{(\epsilon,\delta)}_{\mathsf{gen-Lip}}$ denote any generic (ϵ, δ)-differentially private algorithm for optimizing a decomposable loss of convex Lipschitz functions over some arbitrary convex set $\tilde{\mathcal{C}} \subseteq \mathcal{C}$. Algorithm 1 from Section 2 is an example of $\mathcal{A}^{(\epsilon,\delta)}_{\mathsf{gen-Lip}}$. Now, we construct an algorithm $\mathcal{A}^{(\epsilon,\delta)}_{gen-str-convex}$ which is the (ϵ, δ) analog of $\mathcal{A}^{\epsilon}_{\mathsf{gen-str-convex}}$ (Algorithm 4). Namely, $\mathcal{A}^{(\epsilon,\delta)}_{gen-str-convex}$ runs in similar fashion to $\mathcal{A}^{\epsilon}_{\mathsf{gen-str-convex}}$ where the only difference is that it takes an extra privacy parameter δ as input and calls algorithms $\mathcal{A}^{(\frac{\epsilon}{2},\frac{\delta}{2})}_{\mathsf{out-pert}}$ and $\mathcal{A}^{(\frac{\epsilon}{2},\frac{\delta}{2})}_{\mathsf{gen-Lip}}$ instead of $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{out-pert}}$ and $\mathcal{A}^{\frac{\epsilon}{2}}_{\mathsf{gen-Lip}}$, respectively.

Theorem B.1 (Privacy guarantee)**.** *Algorithm $\mathcal{A}^{(\epsilon,\delta)}_{gen-str-convex}$ is (ϵ, δ)-differentially private.*

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

Theorem B.2 (Generic utility guarantee)**.** *Let $\widetilde{\theta}$ denote the output of Algorithm $\mathcal{A}^{(\epsilon,\delta)}_{\mathsf{gen-Lip}}$ on inputs $n, \mathcal{D}, \ell, \epsilon, \delta, \tilde{\mathcal{C}}$ (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*

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

for some function F, then the output θ^{priv} of $\mathcal{A}^{(\epsilon,\delta)}_{\mathsf{gen-str-convex}}$ satisfies

$$E\left[\mathcal{L}(\theta^{priv};\mathcal{D}) - \mathcal{L}(\theta^*;\mathcal{D})\right] \leq O\left(F\left(p, n, \epsilon, \delta, L, O\left(\frac{L\sqrt{p\log\left(\frac{1}{\delta}\right)\log(n)}}{\Delta\epsilon n}\right)\right)\right).$$

Proof. The proof follows the same lines of the proof of Theorem 4.2 except for the fact that, in Algorithm $\mathcal{A}^{(\frac{\epsilon}{2},\frac{\delta}{2})}_{\mathsf{out-pert}}$, the noise vector b is Gaussian and hence using the standard bounds on the norm of an i.i.d. Gaussian vector, we have

$$\Pr\left[\|b\|_2 \leq \zeta\sigma_0\sqrt{p}\right] = \Pr\left[\|b\|_2 \leq \zeta\frac{4L\sqrt{\log\left(\frac{2}{\delta}\right)}}{\Delta\epsilon n}\right] \geq 1 - e^{-\Omega(\zeta^2)}$$

We set $\zeta = \sqrt{3\log(n)}$ and the rest of the proof follows in the same way as the proof of Theorem 4.2. □

C Converting Excess Risk Bounds in Expectation to High-probability Bounds

In this paper all of our utility guarantees are in terms of the expectation over the randomness of the algorithm. Although all the utility analysis except for the gradient descent based algorithm (Algorithm 1) provide high-probability guarantees directly, in this section we provide a generic approach for obtaining high-probability guarantee based on the expected risk bounds. The idea is to run the underlying differentially private algorithm k-times, with the privacy parameters ϵ/k and δ/k for each run. Let $\theta_1^{priv}, \cdots \theta_k^{priv}$ be the vectors output by the k-runs. First notice that the vector $\theta_1^{priv}, \cdots, \theta_k^{priv}$ is (ϵ, δ)-differentially private. Moreover if the algorithm has expected excess risk of $F(\epsilon, \delta)$ (where F is the specific excess risk function of ϵ and δ), then by Markov's inequality there exist an execution of the algorithm $i \in [k]$ for which the excess risk is $2F(\epsilon/k, \delta/k)$ with probability at least $1 - 1/2^k$.

One can now use the exponential mechanism from Algorithm 2, to pick the best θ_i^{priv} from the list. By the same analysis of Theorem 3.2, one can show that with probability at least $1 - \rho/2$, the exponential mechanism will output a vector θ^{priv} that has excess risk of $\max_i \mathsf{Excess\_risk}(\theta_i^{priv}) - O\left(\frac{L\|\mathcal{C}\|_2}{\epsilon}\log(k/\rho)\right)$. Setting $k = \log(2/\rho)$, we have that with probability at lest $1 - \rho$, the excess risk for θ^{priv} is at most $O(F(\frac{\epsilon}{\log(1/\rho)}, \frac{\delta}{\log(1/\rho)}))$. Placing this bound in context of the paper, the high probability bounds are only a poly $\log(1/\rho)$ factor off from the expectation bounds.

D Excess Risk Bounds for Smooth Functions

In this section we present the scenario where each of the loss function $\ell(\theta; d)$ (for all d in the domain) is β-smooth in addition to being L-Lipschitz (for $\theta \in \mathcal{C}$). It turns out that both for ϵ and (ϵ, δ)-differential privacy, objective perturbation algorithm (see (4)) [7, 30] achieves the best possible error guarantees, where the random variable b is either sampled i) from the Gamma distribution with the kernel $\propto e^{-\frac{\epsilon\|b\|_2}{2L}}$. or ii) from the Normal distribution $\mathcal{N}\left(0, \mathbb{I}_p \frac{8L^2\log(1/\delta)}{\epsilon^2}\right)$. In terms of privacy, when the noise vector b is from Gamma distribution, the algorithm is ϵ-differentially private. And when the noise is from Normal distribution, it is (ϵ, δ)-differentially private. For completeness purposes, we also state the error bounds from [30] (translated to the context of this paper).

$$\theta^{priv} = \arg\min_{\theta\in\mathcal{C}} \mathcal{L}(\theta; \mathcal{D}) + \frac{\Delta}{2}\|\theta\|_2^2 + \langle b, \theta\rangle \tag{4}$$

Theorem D.1 (Lipschitz and smooth function)**.** *The excess risk bounds are as follows:*

1. *[7] With Gamma density ν_1, setting $\Delta = \Theta\left(\frac{Lp}{\epsilon\|\mathcal{C}\|_2}\right)$ and assuming $\Delta \geq \frac{\beta}{2\epsilon}$, we have $\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{L\|\mathcal{C}\|_2 p}{\epsilon}\right)$.*
2. *[30] With Gaussian density, setting $\Delta = \Theta\left(\frac{\sqrt{L^2 p\log(1/\delta)}}{\epsilon\|\mathcal{C}\|_2}\right)$ and assuming $\Delta \geq \frac{\beta}{2\epsilon}$, we have $\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{L\|\mathcal{C}\|_2\sqrt{p\ln(1/\delta)}}{\epsilon}\right)$.*

Additionally when the loss function $\ell(\theta; d)$ is Δ-strongly convex (for $\theta \in \mathcal{C}$) for all d in the domain with the condition that $\Delta \geq \frac{\beta}{2\epsilon}$, one can essentially recover the tight error guarantees for the ϵ and (ϵ, δ) case of differential privacy respectively. The main observation is that for the privacy guarantee to be achieved one need not add the additional regularizer. Although not in its explicit form, a variant of this observation appears in the work of [30]. We state the error guarantee from [30, Theorem 31] translated to our setting. Notice that unlike Theorem D.1, the error guarantee in Theorem D.2 does not depend on the diameter of the convex set $\mathcal{C}$.

Theorem D.2 (Lipschitz, smooth and strongly convex function)**.** *The excess risk bounds are as follows:*

1. *With Gamma density ν_1, if $\Delta \geq \frac{\beta}{2\epsilon}$, we have $\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{L^2 p^2}{n\Delta\epsilon^2}\right)$.*
2. *With Gaussian density, if $\Delta \geq \frac{\beta}{2\epsilon}$, we have $\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = O\left(\frac{L^2 p \ln(1/\delta)}{n\Delta\epsilon^2}\right)$.*

E Straightforward Smoothing Does Not Yield Optimal Algorithms

In Section D we saw that the objective perturbation algorithm (4) of [7, 30] already matches the optimal excess risk bounds for Lipschitz, and Lipschitz and strongly convex functions when the loss function ℓ is twice-continuously differentiable with a bounded double derivative β. A natural question that arises, *is it possible to smoothen out a non-smooth loss function by convolving with a smooth kernel (like the Gaussian kernel) or by Huberization, and still achieve the optimal excess risk bound?* In this section we look at a simple loss functions (the hinge loss) and a very popular Huberization method (quadratic smoothing) argue that there is an inherent cost due to smoothing which will not allow one to get the optimal excess risk bounds.

Consider the loss function $\ell(\theta; d) = (y - x\theta)^+$, where the data point $d = (x, y)$, and $x, y \in [-1, 1]$ and $\theta \in \mathbb{R}$. Here the function $f(z) = (z)^+$ is equal to z when $z > 0$ and zero otherwise. Clearly $f(z)$ has a point of non-differentiability at zero. We can modify the function f, in the following way, to ensure that the resulting function $\hat{f}$ is smooth (or twice-continuously differentiable). Define $\hat{f}(z) = f(z)$, when $z < -h$ or when $z > h$. In the range $[-h, h]$, we set $\hat{f}(z) = \frac{z^2}{4h} + \frac{z}{2} + \frac{h}{4}$. It is not hard to verify that the function $\hat{f}(z)$ is twice-continuously differentiable everywhere. This form of smoothing is commonly called Huberization. Let the smoothed version of $\ell(\theta; d)$ be defined as $\hat{\ell}(\theta; d) = \hat{f}((y - x\theta))$ for $d = (x, y)$.

With the choice of loss function $\hat{\ell}$, the objective perturbation algorithm is as below. (The regularization coefficient is chosen to ensure that it is at least $\frac{\beta}{2\epsilon}$, where β is the smoothness parameter of $\hat{\ell}$.):

$$\theta^{priv} = \arg\min_{\theta \in [-2,2]} \sum_{i=1}^{n} \hat{\ell}(\theta; d_i) + \frac{\theta^2}{8\epsilon h} + b\theta \tag{5}$$

In (5) the noise $b \sim \mathcal{N}(0, \frac{8\log(1/\delta)}{\epsilon^2})$. In the results to follow, we show that for any choice of the Huberization parameter h, there exists data sets of size n from the domain above where the excess risk for objective perturbation will be provably worse than our results in this paper. We present the results for the (ϵ, δ)-differential privacy case, but the same conclusions hold for the pure ϵ-differential privacy case.

Theorem E.1. *For every $h > 0$, there exists $\mathcal{D}$ such the excess risk for the objective perturbation algorithm in* (5) *satisfies:*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})\right] = \Omega\left(\min\left\{n, \max\{nh, \tfrac{1}{h}\}\right\}\right) = \Omega(\sqrt{n})\,.$$

Here the loss function $\mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{n} \ell(\theta; d_i)$ (where $\mathcal{D} = \{d_1, \cdots, d_n\}$) and $\theta^* = \arg\min_{\theta \in [-2,2]} \sum_{i=1}^{n} \ell(\theta; d_i)$.

Proof. Consider the data set $\mathcal{D}_1$ with $\frac{n}{3}$ entries being $(x = -1, y = 1)$ and $\frac{2n}{3}$ entries being $(x = 1, y = -1)$. In the following lemma we lower bound the excess risk on $\mathcal{D}_1$ for a given huberization parameter h.

Lemma E.2. *Let ϵ, δ be the privacy parameters with ϵ being a constant (< 1) and $\delta = \Omega\left(\frac{1}{n^4}\right)$. For the data set $\mathcal{D}_1$ mentioned above, the excess risk for objective perturbation* (5) *is as follows. For all $h > 0$, we have*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}_1) - \mathcal{L}(\theta; \mathcal{D}_1)\right] = \Omega\left(n \cdot \min\{1, h\}\right).$$

Here the loss function $\mathcal{L}(\theta; \mathcal{D}_1) = \sum_{i=1}^{n} \ell(\theta; d_i)$ (where $\mathcal{D}_1 = \{d_1, \cdots, d_n\}$) and $\theta^* = \arg\min_{\theta \in [-2,2]} \sum_{i=1}^{n} \ell(\theta; d_i)$.

Proof. For the ease of notation, let $\hat{\mathcal{L}}(\theta; \mathcal{D}_1) = \sum_{i=1}^{n} \hat{\ell}(\theta; d_i)$. First notice two properties of $\hat{\mathcal{L}}$: i) the minimizer (call it $\hat{\theta}$ within the set $[-2, 2]$ is at $\max\{-2, 1-h\}$, and ii) $\hat{\mathcal{L}}$ is quadratic within the range $[1-h, 1+h]$ with strong convexity parameter at least $\frac{n}{6h}$. Additionally notice that $\theta^* = 1$ and the regularizer $\frac{\theta^2}{8\epsilon h}$ in (5) is centered at zero. Also by Markov's inequality, w.p. $\geq 2/3$, we have $|b| \leq \frac{8\sqrt{\log(1/\delta)}}{\epsilon}$. Now to satisfy optimality, $\frac{n|\theta^{priv} - \hat{\theta}|}{3h} \leq |b|$. This suggests that $|\theta^{priv} - \hat{\theta}| \leq \frac{3h|b|}{n}$.Therefore, the difference $\left(\theta^* - \theta^{priv}\right)$ is at least $\min\left\{1, h\left(1 - \frac{3|b|}{n}\right)\right\}$. Therefore the excess risk with probability at least $2/3$ is $\Omega\left(n \cdot \min\{1, h\}\right)$, which concludes the proof.

□

Consider a data set $\mathcal{D}_2$ which has exactly $\max\{\frac{n}{2} - \frac{1}{32h}, 0\}$ entries with $(x = -1, y = 1)$ and $\min\{\frac{n}{2} + \frac{1}{32h}, n\}$ entries with $(x = 1, y = 1)$. In the following lemma we lower bound the excess risk on $\mathcal{D}_2$ for a given huberization parameter h.

Lemma E.3. *Let ϵ, δ be the privacy parameters with ϵ being a constant (< 1) and $\delta = \Omega\left(\frac{1}{n^4}\right)$. Let $h < \frac{1}{\log n}$ be a fixed Huberization parameter. Then for the data set $\mathcal{D}_2$ mentioned above, the excess risk for objective perturbation* (5) *is as follows.*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}; \mathcal{D}_2) - \mathcal{L}(\theta^*; \mathcal{D}_2)\right] = \Omega\left(\min\left\{\frac{1}{h}, n\right\}\right).$$

Here the loss function $\mathcal{L}(\theta; \mathcal{D}_2) = \sum_{i=1}^{n} \ell(\theta; d_i)$ and $\theta^* = \arg\min_{\theta \in [-2,2]} \sum_{i=1}^{n} \ell(\theta; d_i)$.

Proof. For the ease of notation, let $\hat{\mathcal{L}}(\theta; \mathcal{D}_2) = \sum_{i=1}^{n} \hat{\ell}(\theta; d_i)$. Notice that within the range $[-1+h, 1-h]$, the slope of $\hat{\mathcal{L}}(\theta; \mathcal{D}_2)$ is $\max\{\frac{-1}{16h}, -n\}$. By the optimality condition of θ^{priv}, we have the following.

$$\frac{\theta^{priv}}{4\epsilon h} + b - \min\{\frac{1}{16h}, n\} = 0 \tag{6}$$

Solving for θ^{priv}, we have $\theta^{priv} = \min\left\{\frac{\epsilon}{4}, 4\epsilon n h\right\} + 4b\epsilon h$. By assumption $h < 1/\log n$ and w.p. $\geq 2/3$ we have $|b| \leq \frac{8\sqrt{\log(1/\delta)}}{\epsilon}$. Therefore, w.p. $\geq 2/3$, we have $\theta^{priv} \leq \epsilon$.

Now notice that with the original loss function ℓ, $\arg\min_{\theta \in [-2,2]} \sum_{i=1}^{n} \ell(\theta; d_i) = 1$. Since the loss function $\mathcal{L}(\theta; \mathcal{D}_2)$ has a slope of $\max\{\frac{-1}{16h}, -n\}$ in the range $[-1, 1]$, the excess risk is $\Omega((1-\epsilon)\min\left\{\frac{1}{h}, n\right\})$ which concludes the proof. □

Finally combining Lemmas E.3 and E.2 completes the proof of Theorem E.1. □