Statistical Query Algorithms for Stochastic Convex Optimization

Vitaly Feldman
vitaly@post.harvard.edu
IBM Research - Almaden

Cristóbal Guzmán[*]
guzman@cwi.nl
Centrum Wiskunde & Informatica

Santosh Vempala
vempala@gatech.edu
School of Computer Science
Georgia Institute of Technology

Abstract

Stochastic convex optimization, where the objective is the expectation of a random convex function, is an important and widely used method with numerous applications in machine learning, statistics, operations research and other areas. We study the complexity of stochastic convex optimization given only *statistical query* (SQ) access to the objective function. We show that well-known and popular methods, including first-order iterative methods and polynomial-time methods, can be implemented using only statistical queries. For many cases of interest we derive nearly matching upper and lower bounds on the estimation (sample) complexity including linear optimization in the most general setting. We then present several consequences for machine learning, differential privacy and proving concrete lower bounds on the power of convex optimization based methods.

A new technical ingredient of our work is SQ algorithms for estimating the mean vector of a distribution over vectors in $\mathbb{R}^d$ with optimal estimation complexity. This is a natural problem and we show that our solutions can be used to get substantially improved SQ versions of Perceptron and other online algorithms for learning halfspaces.

[*] Part of this work was done during an internship at IBM Research - Almaden, and at a postdoctoral position of Núcleo Milenio Información y Coordinación en Redes (ICM/FIC P10-024F) at Universidad de Chile.

1 Introduction

In stochastic convex optimization the goal is to minimize a convex function $F(x) = \mathbf{E}_{\mathbf{w}}[f(x, \mathbf{w})]$ over a convex set $\mathcal{K} \subset \mathbb{R}^d$, where $\mathbf{w}$ is a random variable distributed according to some distribution D over domain $\mathcal{W}$ and each $f(x, w)$ is convex in x. The optimization is based on i.i.d. samples $w^1, w^2, \ldots, w^n$ of $\mathbf{w}$. Numerous central problems in machine learning and statistics are special cases of this general setting with a vast literature devoted to techniques for solving variants of this problem (*e.g.* [84, 79]). It is usually assumed that $\mathcal{K}$ is "known" to the algorithm (or in some cases given via a sufficiently strong oracle) and the key challenge is understanding how to cope with estimation errors arising from the stochastic nature of information about $F(x)$.

Here we consider the complexity of solving stochastic convex minimization problems by a restricted class of algorithms, referred to as *statistical (query) algorithms*. Statistical query (SQ) algorithms, defined by Kearns [55] in the context of PAC learning and by Feldman et al. [36] for general problems on inputs sampled i.i.d. from distributions, are algorithms that can be implemented using estimates of the expectation of any given function on a sample drawn randomly from the input distribution D instead of direct access to random samples. Such access is abstracted using a *statistical query oracle* that given a query function $\phi : \mathcal{W} \to [-1, 1]$ returns an estimate of $\mathbf{E}_{\mathbf{w}}[\phi(\mathbf{w})]$ within some tolerance τ. We will refer to the number of samples sufficient to estimate the expectation of each query of a SQ algorithm with some fixed constant confidence as its *estimation complexity* (often $1/\tau^2$) and the number of queries as its *query complexity*.

Reducing data access to estimation of simple expectations has a variety of useful properties. First, a SQ algorithm can be used to automatically derive an algorithm with additional useful properties such as noise-tolerance [55], differential-privacy [15, 54], distributed computation [20, 5], evolvability [33, 34] and generalization in adaptive data analysis [32]. This leads to the general question of which analyses can be decomposed in this way and what are the overheads of doing so (as compared to using the samples in an unrestricted way).

The second important property of statistical algorithms is that it is possible to prove information-theoretic lower bounds on the complexity of any statistical algorithm that solves a given problem. From this perspective, statistical algorithms for solving stochastic convex optimization allow one to convert an optimization algorithm into a lower bound on using convex optimization to solve the problem. For many problems in machine learning and computer science, convex optimization gives state-of-the-art results and therefore lower bounds against such techniques are a subject of significant research interest. Indeed, in recent years this area has been particularly active with major progress made on several long-standing problems (*e.g.* [38, 77, 66, 57]). It should be pointed out that the resulting lower bounds are *concrete* in the sense that they are structural results that do not rely on any oracles (see Section 6.4 for more details).

One of the most successful approaches for solving convex programs in theory and practice is iterative first-order methods, namely techniques that rely on updating the current point x^t using the gradient of F at x^t. It can be immediately observed that for every x, $\nabla F(x) = \mathbf{E}_{\mathbf{w}}[\nabla f(x, \mathbf{w})]$ and hence it is sufficient to estimate expected gradients to some sufficiently high accuracy in order to implement such algorithms (we are only seeking an approximate optimum anyway). The accuracy corresponds to the number of samples (or estimation complexity) and is the key measure of complexity for SQ algorithms. However, to the best of our knowledge, the estimation complexity for specific SQ implementations of first-order methods has not been previously addressed. This is in contrast to the rich and nuanced understanding of the sample and computational complexity of solving such problems given unrestricted access to samples.

1.1 Overview of Results

In this work we give SQ algorithms for a number of the commonly considered stochastic convex optimization problems. We also prove that in a range of settings our implementations achieve nearly optimal bounds.

The key new technical ingredients are algorithms for estimating the mean vector of a distribution over vectors in $\mathbb{R}^d$, a natural problem of independent interest. We then demonstrate several applications of our results to obtain new algorithms and lower bounds.

1.1.1 Linear optimization via mean estimation

We start with the linear optimization case which is a natural special case and also the basis of our implementations of first-order methods. In this setting $\mathcal{W} \subseteq \mathbb{R}^d$ and $f(x,w) = \langle x, w \rangle$. Hence $F(x) = \langle x, \bar{w} \rangle$, where $\bar{w} = \mathbf{E}_{\mathbf{w}}[\mathbf{w}]$. This reduces the problem to finding a sufficiently accurate estimate of $\bar{w}$. Specifically, for a given error parameter ε, it is sufficient to find a vector $\tilde{w}$, such that for every $x \in \mathcal{K}$, $|\langle x, \bar{w} \rangle - \langle x, \tilde{w} \rangle| \leq \varepsilon$. Given such an estimate $\tilde{w}$, we can solve the original problem with error of at most 2ε by solving $\min_{x \in \mathcal{K}} \langle x, \tilde{w} \rangle$.

An obvious way to estimate the high-dimensional mean using SQs is to simply estimate each of the coordinates of the mean vector using a separate SQ: that is $\mathbf{E}[\mathbf{w}_i/B_i]$, where $[-B_i, B_i]$ is the range of $\mathbf{w}_i$. Unfortunately, even in the most standard setting, where both $\mathcal{K}$ and $\mathcal{W}$ are ℓ_2 unit balls, this method requires accuracy that scales with $1/\sqrt{d}$ (or estimation complexity that scales linearly with d). In contrast, bounds obtained using samples are dimension-independent making this SQ implementation unsuitable for high-dimensional applications. Estimation of high-dimensional means for various distributions is (arguably) an even more basic question than stochastic optimization; yet we are not aware of any prior analysis of its statistical query complexity. In particular, SQ implementation of all algorithms for learning halfspaces (including the most basic Perceptron) require estimation of high-dimensional means but known analyses rely on inefficient coordinate-wise estimation (*e.g.* [17, 14, 4]).

Here we aim to address the high-dimensional mean estimation problem in detail and, specifically, to investigate whether the SQ estimation complexity is different from sample complexity of the problem. The first challenge here is that even the sample complexity of mean estimation depends in an involved way on the geometry of $\mathcal{K}$ and $\mathcal{W}$ and in this generality is not fully understood (*cf.* [71]). We therefore focus our attention on the most commonly studied setting, where $\mathcal{K}$ is a unit ball in ℓ_p norm and $\mathcal{W}$ is the unit ball in ℓ_q norm for $p \in [1, \infty]$ and $1/p + 1/q = 1$ (general radii can be reduced to this setting by scaling). This is equivalent to requiring that $\|\tilde{w} - \bar{w}\|_q \leq \varepsilon$ for a random variable $\mathbf{w}$ supported on the unit ℓ_q ball and we refer to it as ℓ_q mean estimation. The sample complexity of ℓ_q mean estimation depends both on q and the relationship between d and ε. We describe the known bounds in Table 1.1.1 (we are not aware of a reference stating the bounds in this form for all q. They are implicit in the literature and we provide the details in Appendix B.) These bounds are tight (up to constants) and are all achieved by using the empirical mean of the samples to estimate $\bar{w}$.

In a nutshell, we give tight (up to a polylogarithmic in d factor) bounds on the SQ complexity of ℓ_q mean estimation for all $q \in [1, \infty]$. These bounds match (up to a polylogarithmic in d factor) the sample complexity of the problem. These upper bounds are based on several different algorithms.

- For $q = \infty$ coordinate-wise estimation gives the desired guarantees.
- For $q = 2$ we show that Kashin's representation of vectors introduced by Lyubarskii and Vershynin [65] can be used to obtain optimal (up to a constant) estimation complexity of $O(1/\varepsilon^2)$ with just $2d$ non-adaptive queries. We also give a randomized algorithm based on estimating the truncated coefficients of the mean in a randomly rotated basis. The algorithm has slightly worse $O(\log(1/\varepsilon)/\varepsilon^2)$ estimation complexity but its analysis is simpler and self-contained.
- For $q \in (2, \infty)$ we use decomposition of the samples into $\log d$ "rings" in which non-zero coefficients have low dynamic range. For each ring we combine ℓ_2 and ℓ_∞ estimation to ensure low error in ℓ_q and optimal estimation complexity.

q	SQ estimation complexity		Sample
	Upper Bound	Lower bound	complexity
$[1,2)$	$O\left(\min\left\{\frac{d^{\frac{2}{q}-1}}{\varepsilon^2}, \left(\frac{\log d}{\varepsilon}\right)^p\right\}\right)$	$\tilde{\Omega}\left(\min\left\{\frac{d^{\frac{2}{q}-1}}{\varepsilon^2}, \frac{1}{\varepsilon^p \log d}\right\}\right)$	$\Theta\left(\min\left\{\frac{d^{\frac{2}{q}-1}}{\varepsilon^2}, \frac{1}{\varepsilon^p}\right\}\right)$
2	$O(1/\varepsilon^2)$	$\tilde{\Omega}(1/(\varepsilon^2 \log d))$	$\Theta(1/\varepsilon^2)$
$(2,\infty)$	$O((\log d/\varepsilon)^2)$	$\Omega(1/\varepsilon^2)$	$\Theta(1/\varepsilon^2)$
∞	$O(1/\varepsilon^2)$	$\Omega(1/\varepsilon^2)$	$\Theta(\log d/\varepsilon^2)$

Table 1: Bounds on ℓ_q mean estimation and linear optimization over ℓ_p ball. Upper bounds use at most $3d \log d$ queries. Lower bounds apply to all algorithms using poly(d/ε) queries.

- For $q \in [1,2)$ there are two regimes. One of the upper bounds is obtained via a reduction to ℓ_2 case (which introduces a d dependent factor). For the second regime we again use a decomposition into "rings" of low dynamic range. For each "ring" we use coordinate-wise estimation and then sparsify the estimate by removing small coefficients. The analysis of this algorithm is fairly delicate and requires using statistical queries in which accuracy takes into account the variance of the random variable (modeled by VSTAT oracle from [36]).

The nearly tight lower bounds are proved using the technique recently introduced in [37]. We prove it for the (potentially simpler) linear optimization problem. We remark that lower bounds on sample complexity do not imply lower bounds on estimation complexity since a SQ algorithm can use many adaptively chosen queries.

We then consider the case of general $\mathcal{K}$ with $\mathcal{W} = \text{conv}(\mathcal{K}^*, -\mathcal{K}^*)$ (which corresponds to normalizing the range of linear functions in the support of the distribution). Here we show that for any polytope $\mathcal{W}$ the estimation complexity is still $O(1/\varepsilon^2)$ but the number of queries grows linearly with the number of faces. More generally, the estimation complexity of $O(d/\varepsilon^2)$ can be achieved for any $\mathcal{K}$. The algorithm relies on knowing John's ellipsoid [49] for $\mathcal{W}$ and therefore depends on $\mathcal{K}$. Designing a single algorithm that given a sufficiently strong oracle for $\mathcal{K}$ (such as a separation oracle) can achieve the same estimation complexity for all $\mathcal{K}$ is an interesting open problem. This upper bound is nearly tight since even for $\mathcal{W}$ being the ℓ_1 ball we give a lower bound of $\tilde{\Omega}(d/\varepsilon^2)$.

1.1.2 Gradient descent and friends

The analysis of the linear case above gives us the basis for tackling first-order optimization methods for the general convex case. That is, we can now obtain an estimate of the expected gradient at each iteration but we still need to ensure that estimation errors from different iterations do not accumulate. Luckily, for this we can build on the study of the performance of first-order methods with inexact oracles. Methods of this type have a long history (*e.g.* [72, 82]), however some of our methods of choice have only been studied recently.

We study the traditional setups of convex optimization: non-smooth, smooth and strongly convex. For the two first classes of problems algorithms use global approximation of the gradient on the feasible domain, which is undesirable in general; however, for the strongly convex case we can show that an oracle introduced by Devolder et al. [25] only requires *local* approximation of the gradient, which leads to improved estimation complexity bounds. We note that smoothness and strong convexity are required only for the expected objective and not necessarily for each function in the support of the distribution.

For the non-smooth case we analyze and apply the classic mirror-descent method [68], for the smooth case we rely on the analysis by d'Aspremont [23] of an inexact variant of Nesterov's accelerated method [69], and for the strongly convex case we use the recent results by Devolder et al. [24] on the inexact dual gradient method. We summarize our results for the ℓ_2 norm in Table 1.1.2. Our results for the mirror-descent and Nesterov's algorithm apply in more general settings (e.g., ℓ_p norms): we refer the reader to Section 4 for the detailed statement of results. In Section 4.3 we also demonstrate and discuss the implications of our results for the well-studied generalized linear regression problems.

Objective	Inexact gradient method	Query complexity	Estimation complexity
Non-smooth	Mirror-descent	$O\left(d \cdot \left(\frac{L_0 R}{\varepsilon}\right)^2\right)$	$O\left(\left(\frac{L_0 R}{\varepsilon}\right)^2\right)$
Smooth	Nesterov	$O\left(d \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}}\right)$	$O\left(\left(\frac{L_0 R}{\varepsilon}\right)^2\right)$
Strongly convex non-smooth	Dual gradient	$O\left(d \cdot \frac{L_0^2}{\varepsilon\kappa} \log\left(\frac{L_0 R}{\varepsilon}\right)\right)$	$O\left(\frac{L_0^2}{\varepsilon\kappa}\right)$
Strongly convex smooth	Dual gradient	$O\left(d \cdot \frac{L_1}{\kappa} \log\left(\frac{L_1 R}{\varepsilon}\right)\right)$	$O\left(\frac{L_0^2}{\varepsilon\kappa}\right)$

Table 2: Upper bounds for inexact gradient methods in the stochastic ℓ_2-setup. Here R is the Euclidean radius of the domain, L_0 is the Lipschitz constant of all functions in the support of the distribution. L_1 is the Lipschitz constant of the gradient and κ is the strong convexity parameter for the expected objective.

1.1.3 Optimization of bounded-range functions

The estimation complexity bounds obtained for gradient descent-based methods depend polynomially on the norm of the gradient of each function in the support of $\mathcal{W}$ and the rad ius of $\mathcal{K}$ (unless the functions are strongly convex). In some cases such bounds are not explicitly available (or too large) and instead we have a bound on the range of $f(x, w)$ for all $w \in \mathcal{W}$ and $x \in \mathcal{K}$. This is a natural setting for stochastic optimization (and statistical algorithms, in particular) since even estimating the value of a given solution x with high probability and any desired accuracy from samples requires some assumptions about the range of most functions.

A bound on range, say $|f(x, w)| \leq 1$ for simplicity, implies that for every x, a single SQ for query function $f(x, w)$ with tolerance τ gives the value $\tilde{F}(x)$ such that $|F(x) - \tilde{F}(x)| \leq \tau$. This, by definition is the τ-approximate value (or zero-order) oracle for $F(x)$. It was proved by Nemirovsky and Yudin [68] and also by Grötschel et al. [42] (who refer to such oracle as *weak evaluation oracle*) that τ-approximate value oracle suffices to ε-minimize $F(x)$ over $\mathcal{K}$ with running time and $1/\tau$ being polynomial in $d, 1/\varepsilon, \log(R_1/R_0)$, where $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$.[1] The analysis in [68, 42] is relatively involved and does not provide explicit bounds on τ.

Nemirovsky and Yudin [68] also prove that even linear optimization over ℓ_2 ball of radius 1 with a τ-approximate value oracle requires $\tau = \tilde{\Omega}(\varepsilon/d)$ for any polynomial-time algorithm. Together with our results this implies that a τ-approximate value oracle is strictly weaker than STAT(τ).

Here we observe that a simple extension of the random walk approach of Kalai and Vempala [51] and Lovász and Vempala [62] can be used with any (ε/d)-approximate value oracle for $F(x)$ to ε-optimize in polynomial time. This approach was also (independently) used in a recent work of Belloni et al. [9] who provide a detailed analysis of the running time and query complexity.

[1] Naturally, assuming some conditions on access to $\mathcal{K}$ such as a membership or a separation oracle. See Thm. 2.2.15 in [60] for a discussion.

We are not constrained to the value information and we give a more efficient algorithm for this setting that is based on the center-of-gravity method and a generalization of our gradient estimation technique to asymmetric bodies. The algorithm uses $O(d^2 \log(1/\varepsilon))$ queries of estimation complexity $O(d^2/\varepsilon^2)$. The reason generalization to asymmetric bodies is necessary is that in the previous analysis the assumptions imply that gradients have bounded norm over $-\mathcal{K}$ and, in particular, over some symmetric body that contains $\mathcal{K}$. While the exact center-of-gravity method is not computationally efficient, we show that the approximate version introduced by Bertsimas and Vempala [12] suffices for our purposes.

1.2 Applications

We now highlight several applications of our results. Additional results can be easily derived in a variety of other contexts that rely on statistical queries (such as evolvability [90], adaptive data analysis [32, 31] and distributed data analysis [20]).

1.2.1 Online Learning of Halfspaces using SQs

Our high-dimensional mean estimation algorithms allow us to revisit SQ implementations of online algorithms for learning halfspaces, such as the classic Perceptron and Winnow algorithms. These algorithms are based on updating the weight vector iteratively using incorrectly classified examples. The convergence analysis of such algorithms relies on some notion of margin by which positive examples can be separated from the negative ones.

A natural way to implement such an algorithm using SQs is to use the mean vector of all positive (or negative) counterexamples to update the weight vector. By linearity of expectation, the true mean vector is still a positive (or correspondingly, negative) counterexample and it still satisfies the same margin condition. This approach was used by Bylander [17] and Blum et al. [14] to obtain algorithms tolerant to random classification noise for learning halfspaces and by Blum et al. [15] to obtain a private version of Perceptron. The analyses in these results use the simple coordinate-wise estimation of the mean and incur an additional factor d in their sample complexity. It is easy to see that to approximately preserve the margin γ it suffices to estimate the mean of some distribution over an ℓ_q ball with ℓ_q error of $\gamma/2$. We can therefore plug our mean estimation algorithms to eliminate the dependence on the dimension from these implementations (or in some cases have only logarithmic dependence). In particular, the estimation complexity of our algorithms is essentially the same as the sample complexity of PAC versions of these online algorithms. Note that such improvement is particularly important since Perceptron is usually used with a kernel (or in other high-dimensional space) and Winnow's main property is the logarithmic dependence of its sample complexity on the dimension.

We note that a variant of the Perceptron algorithm referred to as Margin Perceptron outputs a halfspace that approximately maximizes the margin [3]. This allows it to be used in place of the SVM algorithm. Our SQ implementation of this algorithm gives an SVM-like algorithm with estimation complexity of $O(1/\gamma^2)$, where γ is the (normalized) margin. This is the same as the sample complexity of SVM (*cf.* [79]). Further details of this application are given in Sec. 6.1.

1.2.2 Lower Bounds

The statistical query framework provides a natural way to convert algorithms into lower bounds. For many problems over distributions it is possible to prove information-theoretic lower bounds against statistical algorithms that are much stronger than known computational lower bounds for the problem. A classical example of such problem is learning of parity functions with noise (or, equivalently, finding an assignment that maximizes the fraction of satisfied XOR constraints). This implies that any algorithm that can be implemented using statistical queries with complexity below the lower bound cannot solve the problem. If the algorithm relies solely on some structural property of the problem, such as approximation of functions by polynomials or computation by a certain type of circuit, then we can immediately conclude a lower bound for that structural property. This indirect argument exploits the power of the algorithm and hence can lead to results which are hard to derive directly.

One inspiring example of this approach comes from using the statistical query algorithm for learning halfspaces [14]. The structural property it relies on is linear separability. Combined with the exponential lower bound for learning parities [55] it immediately implies that there is no mapping from $\{-1,1\}^d$ to $\mathbb{R}^N$ which makes parity functions linearly separable for any $N \leq N_0 = 2^{\Omega(d)}$. Subsequently, and apparently unaware of this technique, Forster [39] proved a $2^{\Omega(d)}$ lower bound on the sign-rank (also known as the dimension complexity) of the Hadamard matrix which is exactly the same result (in [81] the connection between these two results is stated explicitly). His proof relies on a sophisticated and non-algorithmic technique and is considered a major breakthrough in proving lower bounds on the sign-rank of explicit matrices.

Convex optimization algorithms rely on existence of convex relaxations for problem instances that (approximately) preserve the value of the solution. Therefore, given a SQ lower bound for a problem, our algorithmic results can be directly translated into lower bounds for convex relaxations of the problem. At a high level, assume that we are dealing with a problem of (approximately) finding $\min_{z\in Z} \frac{1}{n}\sum_{i\leq n} v_i(z)$ given a sequence of real-valued functions $(v_i)_{i=1}^n$ from some collection of functions V over a domain Z. These functions are not restricted and could represent a loss of the solution given by z on a point represented by v_i or whether an assignment represented by z satisfies a constraint represented by v_i. Further, assume that we are given a lower bound on the SQ complexity of ε-approximating $\mathrm{Val}(D) \doteq \min_{z\in Z} \mathbf{E}_{v\sim D}[v(z)]$ for an unknown distribution D from some (known) collection of distributions $\mathcal{D}$ over V. Now, assume that for a set of convex functions $\mathcal{F}$ over $\mathcal{K} \subseteq \mathbb{R}^d$, stochastic optimization over $\mathcal{K}$ for distributions supported on $\mathcal{F}$ can be solved with accuracy $\varepsilon/2$ by a SQ algorithm with complexity below the given lower bound. This implies there does not exist a mapping $T : V \to \mathcal{F}$ such that for all $D \in \mathcal{D}$, $|\mathrm{Val}(D) - \min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)]| < \varepsilon/2$. Canonical LP/SDP relaxations of constraint satisfaction problems and *surrogate loss* convex relaxations used in machine learning are instances of mappings with such property (or other form of approximation). We defer the formal statement of this result and some concrete corollaries based on lower bounds from [37] to Section 6.4.

1.2.3 Differential Privacy

In local or *randomized-response* differential privacy the users provide the analyst with differentially private versions of their data points. Any analysis performed on such data is differentially private so, in effect, the data analyst need not be trusted. Such algorithms have been studied and applied for privacy preservation since at least the work of Warner [92]. While there exists a large and growing literature on mean estimation and convex optimization with (global) differential privacy (*e.g.* [19, 29, 7]), these questions have been only recently and partially addressed for the more stringent local privacy. Using simple estimation of statistical queries with local differential privacy by Kasiviswanathan et al. [54] we directly obtain a variety of corollaries for locally differentially private mean estimation and optimization. Some of them, including mean estimation for ℓ_2 and ℓ_∞ norms and their implications for gradient and mirror descent algorithms are known via specialized arguments [26, 27]. Our corollaries for mean estimation achieve the same bounds up to logarithmic in d factors. We also obtain corollaries for more general mean estimation problems and results for optimization that, to the best of our knowledge, were not previously known.

An additional implication in the context of differentially private data analysis is to the problem of releasing answers to multiple queries over a single dataset. A long line of research has considered this question for *linear* or *counting* queries which for a dataset $S \subseteq \mathcal{W}^n$ and function $\phi : \mathcal{W} \to [0,1]$ output an estimate of $\frac{1}{n}\sum_{w\in S}\phi(w)$ (see [29] for an overview). In particular, it is known that an exponential in n number of such queries can be answered differentially privately even when the queries are chosen adaptively [76, 46] (albeit the running time is linear in $|\mathcal{W}|$). Recently, Ullman [89] has considered the question of answering *convex minimization* queries which ask for an approximate minimum of a convex program taking a data point as an input averaged over the dataset. For several convex minimization problems he gives algorithms that can answer an exponential number of convex minimization queries. It is easy to see that the problem considered by Ullman [89] is a special case of our problem by taking the input distribution to be uniform over the points in S. A statistical query for this distribution is equivalent to a counting query and hence our algorithms effectively reduce answering of convex minimization queries to answering of counting queries. As a corollary we strengthen and substantially generalize the results in [89].

Details of these applications appear in Sections 6.2 and 6.3.

1.3 Related work

The SQ framework was introduced by Kearns [55], who showed how to derive PAC learning algorithms robust to random classification noise from SQ algorithms. Closely related concepts are linear statistical functionals studied in statistics (*e.g.* [93]) and the learning-by-distances model of Ben-David et al. [10]. Blum et al. [15] show how to implement a SQ algorithm with *differential privacy* [30] and Kasiviswanathan et al. [54] additionally show a simulation preserving more stringent *local* differential privacy. This connection has been used to get privacy-preserving algorithms in a number of additional contexts [5, 4].

Chu et al. [20] show that empirical estimation of expectations can be automatically parallelized on multicore architectures and give many examples of popular machine learning algorithms that can be sped up using this approach. SQ algorithms can be used to derive algorithms in Valiant's (2009) model of evolvability [33, 34]. In this context, Valiant [91] shows that the weak evaluation oracle from [42] can be implemented in the model of evolvability thereby obtaining polynomial-time evolution algorithms for stochastic convex optimization (albeit without any specific bounds). More recently, in a line of work initiated by Dwork et al. [32], SQs have been used as a basis for understanding generalization in adaptive data analysis [32, 47, 31, 86, 8].

The first lower bound for SQ algorithms was given by Kearns [55] for the problem of learning parity functions. Blum et al. [13] described a general technique for the analysis of the complexity of PAC learning using SQs based on the notion of SQ dimension. Subsequently, similar techniques were developed for more general learning settings and more recently for general problems over distributions [83, 35, 88, 36, 37]. Using these techniques, strong lower bounds for a number of fundamental problems in machine learning theory were obtained (such as PAC learning of juntas [13] and agnostic learning of monomials [35]) as well as for stochastic versions of several classical problems in computer science (including planted bi-clique [36] and planted satisfiability [37]).

There is long history of research on the complexity of convex optimization with access to some type of oracle (*e.g.* [68, 16, 45]) with a lot of renewed interest due to applications in machine learning (*e.g.* [73, 1]). In particular, a number of works study robustness of optimization methods to errors by considering oracles that provide approximate information about F and its (sub-)gradients [23, 25]. Our approach to getting statistical query algorithms for stochastic convex optimization is based in large part on implementing different approximate first-order oracles by a SQ oracle. This allows us to use known insights and results to derive SQ algorithms (and, naturally, the reduction can be used similarly to derive new algorithms).

A common way to model stochastic optimization is via a stochastic oracle for the objective function [68]. Such oracle is assumed to return a random variable whose expectation is equal to the exact value of the function and/or its gradient (most commonly the random variable is Gaussian or has bounded variance). Analyses of such algorithms (most notably the Stochastic Gradient Descent (SGD)) are rather different from ours although in both cases linearity and robustness properties of first-order methods are exploited. In most settings we consider, estimation complexity of our SQ agorithms is comparable to sample complexity of solving the same problem using an appropriate version of SGD (which is, in turn, often known to be optimal). On the other hand lower bounds for stochastic oracles (*e.g.* [1]) have a very different nature and it is impossible to obtain superpolynomial lower bounds on the number of oracle calls (such as those we prove in Section 3.2).

In a recent (and independent) work Steinhardt et al. [85] have established a number of relationships between learning with SQs and learning with several types of restrictions on memory and communication. Among other results, they proved an unexpected upper bound on memory-bounded sparse least-squares regression by giving a SQ algorithm for the problem. Their algorithm is based on inexact mirror-descent over the ℓ_1-ball and is a special case of our more general analysis (in optimization over ℓ_1 ball, ℓ_∞ estimation of gradients suffices bypassing the difficulties associated with other norms). Our results can be used to derive bounds of this type for other learning problems.

2 Preliminaries

For integer $n \geq 1$ let $[n] \doteq \{1, \ldots, n\}$. Typically, d will denote the ambient space dimension, and n will denote number of samples. Random variables are denoted by bold letters, e.g., $\mathbf{w}$, $\mathbf{U}$. We denote the indicator function of an event A (i.e., the function taking value zero outside of A, and one on A) by $\mathbf{1}_{\{A\}}$.

For $i \in [d]$ we denote by e_i the i-th basis vector in $\mathbb{R}^d$. Given a norm $\|\cdot\|$ on $\mathbb{R}^d$ we denote the ball of radius $R > 0$ by $\mathcal{B}^d_{\|\cdot\|}(R)$, and the unit ball by $\mathcal{B}^d_{\|\cdot\|}$. We also recall the definition of the norm dual to $\|\cdot\|$, $\|w\|_* \doteq \sup_{\|x\|\leq 1}\langle w, x\rangle$, where $\langle\cdot,\cdot\rangle$ is the standard inner product of $\mathbb{R}^d$.

For a convex body (i.e., compact convex set with nonempty interior) $\mathcal{K} \subseteq \mathbb{R}^d$ we define its polar as $\mathcal{K}_* = \{w \in \mathbb{R}^d : \langle w, x\rangle \leq 1 \ \forall x \in \mathcal{K}\}$, and we have that $(\mathcal{K}_*)_* = \mathcal{K}$. Any origin-symmetric convex body $\mathcal{K} \subset \mathbb{R}^d$ (i.e., $\mathcal{K} = -\mathcal{K}$) defines a norm $\|\cdot\|_\mathcal{K}$ as follows: $\|x\|_\mathcal{K} = \inf_{\alpha>0}\{\alpha \mid x/\alpha \in \mathcal{K}\}$, and $\mathcal{K}$ is the unit ball of $\|\cdot\|_\mathcal{K}$. It is easy to see that the norm dual to $\|\cdot\|_\mathcal{K}$ is $\|\cdot\|_{\mathcal{K}_*}$.

Our primary case of interest corresponds to ℓ_p-setups. Given $1 \leq p \leq \infty$, we consider the normed space $\ell_p^d \doteq (\mathbb{R}^d, \|\cdot\|_p)$, where for a vector $x \in \mathbb{R}^d$, $\|x\|_p \doteq \left(\sum_{i\in[d]}|x_i|^p\right)^{1/p}$. For $R \geq 0$, we denote by $\mathcal{B}^d_p(R) = \mathcal{B}^d_{\|\cdot\|_p}(R)$ and similarly for the unit ball, $\mathcal{B}^d_p = \mathcal{B}^d_p(1)$. We denote the conjugate exponent of p as q, meaning that $1/p + 1/q = 1$; with this, the norm dual to $\|\cdot\|_p$ is the norm $\|\cdot\|_q$. In all definitions above, when clear from context, we will omit the dependence on d.

We consider problems of the form

$$F^* \doteq \min_{x\in\mathcal{K}}\left\{F(x) \doteq \mathop{\mathbf{E}}_{\mathbf{w}}[f(x,\mathbf{w})]\right\}, \tag{1}$$

where $\mathcal{K}$ is a convex body in $\mathbb{R}^d$, $\mathbf{w}$ is a random variable defined over some domain $\mathcal{W}$, and for each $w \in \mathcal{W}$, $f(\cdot, w)$ is convex and subdifferentiable on $\mathcal{K}$. For an approximation parameter $\varepsilon > 0$ the goal is to find $x \in \mathcal{K}$ such that $F(x) \leq F^* + \varepsilon$, and we call any such x an *ε-optimal solution*. We denote the probability distribution of $\mathbf{w}$ by D and refer to it as the input distribution. For convenience we will also assume that $\mathcal{K}$ contains the origin.

Statistical Queries. The algorithms we consider here have access to a statistical query oracle for the input distribution. For most of our results a basic oracle introduced by Kearns [55] that gives an estimate of the mean with fixed tolerance will suffice. We will also rely on a stronger oracle that captures estimation of the mean of a random variable from samples more accurately and was introduced in [36].

Definition 2.1. *Let D be a distribution over a domain $\mathcal{W}$, $\tau > 0$ and n be an integer. A statistical query oracle STAT$_D(\tau)$ is an oracle that given as input any function $\phi : \mathcal{W} \rightarrow [-1,1]$, returns some value v such that $|v - \mathbf{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w})]| \leq \tau$. A statistical query oracle VSTAT$_D(n)$ is an oracle that given as input any function $\phi : \mathcal{W} \rightarrow [0,1]$ returns some value v such that $|v - p| \leq \max\left\{\frac{1}{n}, \sqrt{\frac{p(1-p)}{n}}\right\}$, where $p \doteq \mathbf{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w})]$. We say that an algorithm is* statistical query *(or, for brevity, just SQ) if it does not have direct access to n samples from the input distribution D, but instead makes calls to a statistical query oracle for the input distribution.*

Clearly VSTAT$_D(n)$ is at least as strong as STAT$_D(1/\sqrt{n})$ (but no stronger than STAT$_D(1/n)$). Query complexity of a statistical algorithm is the number of queries it uses. The *estimation complexity* of a statistical query algorithm using VSTAT$_D(n)$ is the value n and for an algorithm using STAT(τ) it is $n = 1/\tau^2$. Note that the estimation complexity corresponds to the number of i.i.d. samples sufficient to simulate the oracle for a single query with at least some positive constant probability of success. However it is not necessarily true that the whole algorithm can be simulated using $O(n)$ samples since answers to many queries need to be estimated. Answering m fixed (or non-adaptive) statistical queries can be done using $O(\log m \cdot n)$ samples but when queries depend on previous answers the best known bounds require $O(\sqrt{m} \cdot n)$ samples (see [32] for a detailed discussion). This also implies that a lower bound on sample complexity of solving a problem does not directly imply lower bounds on estimation complexity of a SQ algorithm for the problem.

Whenever that does not make a difference for our upper bounds on estimation complexity, we state results for STAT to ensure consistency with prior work in the SQ model. All our lower bounds are stated for the stronger VSTAT oracle. One useful property of VSTAT is that it only pays linearly when estimating expectations of functions conditioned on a rare event:

Lemma 2.2. *For any function $\phi : X \rightarrow [0,1]$, input distribution D and condition $A : X \rightarrow \{0,1\}$ such that $p_A \doteq \mathbf{Pr}_{x\sim D}[A(x) = 1] \geq \alpha$, let $p \doteq \mathbf{E}_{x\sim D}[\phi(x) \cdot A(x)]$. Then query $\phi(x) \cdot A(x)$ to VSTAT(n/α) returns a value v such that $|v - p| \leq \frac{p_A}{\sqrt{n}}$.*

Proof. The value v returned by VSTAT(n/α) on query $\phi(x)\cdot A(x)$ satisfies: $|v-p| \leq \min\left\{\frac{\alpha}{n}, \sqrt{\frac{p(1-p)\alpha}{n}}\right\}$. Note that $p = \mathbf{E}[\phi(x)A(x)] \leq \mathbf{Pr}[A(x) = 1] = p_A$. Hence $|v - p| \leq \frac{p_A}{\sqrt{n}}$. □

Note that one would need to use STAT$(\alpha/\sqrt{n})$ to obtain a value v with the same accuracy of $\frac{p_A}{\sqrt{n}}$ (since p_A can be as low as α). This corresponds to estimation complexity of n/α^2 vs. n/α for VSTAT.

3 Stochastic Linear Optimization and Vector Mean Estimation

We start by considering stochastic linear optimization, that is instances of the problem

$$\min_{x\in\mathcal{K}}\{\mathop{\mathbf{E}}_{\mathbf{w}}[f(x,\mathbf{w})]\}$$

in which $f(x,w) = \langle x, w\rangle$. From now on we will use the notation $\bar{w} \doteq \mathbf{E}_{\mathbf{w}}[\mathbf{w}]$.

For normalization purposes we will assume that the random variable $\mathbf{w}$ is supported on $\mathcal{W} = \{w \mid \forall x \in \mathcal{K},\ |\langle x, w\rangle| \leq 1\}$. Note that $\mathcal{W} = \text{conv}(\mathcal{K}_*, -\mathcal{K}_*)$ and if $\mathcal{K}$ is origin-symmetric then $\mathcal{W} = \mathcal{K}_*$. More generally, if $\mathbf{w}$ is supported on $\mathcal{W}$ and $B \doteq \sup_{x\in\mathcal{K},\ w\in\mathcal{W}}\{|\langle x, w\rangle|\}$ then optimization with error ε can be reduced to optimization with error ε/B over the normalized setting by scaling.

We first observe that for an origin-symmetric $\mathcal{K}$, stochastic linear optimization with error ε can be solved by estimating the mean vector $\mathbf{E}[\mathbf{w}]$ with error $\varepsilon/2$ measured in $\mathcal{K}_*$-norm and then optimizing a deterministic objective.

Observation 3.1. *Let $\mathcal{W}$ be an origin-symmetric convex body and $\mathcal{K} \subseteq \mathcal{W}_*$. Let $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E}[\langle x, \mathbf{w}\rangle]\}$ be an instance of stochastic linear optimization for $\mathbf{w}$ supported on $\mathcal{W}$. Let $\tilde{w}$ be a vector such that $\|\tilde{w} - \bar{w}\|_{\mathcal{W}} \leq \varepsilon/2$. Let $\tilde{x} \in \mathcal{K}$ be such that $F(\tilde{x}) \leq \min_{x\in\mathcal{K}}\langle \tilde{w}, x\rangle + \xi$. Then for all $x \in \mathcal{K}$, $F(\tilde{x}) \leq F(x) + \varepsilon + \xi$.*

Proof. Note that $F(x) = \langle x, \bar{w}\rangle$ and let $\bar{x} = \operatorname{argmin}_{x\in\mathcal{K}}\langle x, \bar{w}\rangle$. The condition $\|\tilde{w} - \bar{w}\|_{\mathcal{W}} \leq \varepsilon/2$ implies that for every $x \in \mathcal{W}_*$, $|\langle x, \tilde{w} - \bar{w}\rangle| \leq \varepsilon/2$. Therefore, for every $x \in \mathcal{K}$,

$$F(\tilde{x}) = \langle \tilde{x}, \bar{w}\rangle \leq \langle \tilde{x}, \tilde{w}\rangle + \varepsilon/2 \leq \langle \bar{x}, \tilde{w}\rangle + \varepsilon/2 + \xi \leq \langle \bar{x}, \bar{w}\rangle + \varepsilon + \xi \leq \langle x, \bar{w}\rangle + \varepsilon + \xi = F(x) + \varepsilon + \xi.$$

□

The mean estimation problem over $\mathcal{W}$ in norm $\|\cdot\|$ is the problem in which, given an error parameter ε and access to a distribution D supported over $\mathcal{W}$, the goal is to find a vector $\tilde{w}$ such that $\|\mathbf{E}_{\mathbf{w}\sim D}[\mathbf{w}] - \tilde{w}\| \leq \varepsilon$. We will be concerned primarily with the case when $\mathcal{W}$ is the unit ball of $\|\cdot\|$ in which case we refer to it as $\|\cdot\|$ mean estimation or mean estimation over $\mathcal{W}$.

We also make a simple observation that if a norm $\|\cdot\|_A$ can be embedded via a linear map into a norm $\|\cdot\|_B$ (possibly with some distortion) then we can reduce mean estimation in $\|\cdot\|_A$ to mean estimation in $\|\cdot\|_B$.

Lemma 3.2. *Let $\|\cdot\|_A$ be a norm over $\mathbb{R}^{d_1}$ and $\|\cdot\|_B$ be a norm over $\mathbb{R}^{d_2}$ that for some linear map $T : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ satisfy: $\forall w \in \mathbb{R}^{d_1}$, $a \cdot \|Tw\|_B \leq \|w\|_A \leq b \cdot \|Tw\|_B$. Then mean estimation in $\|\cdot\|_A$ with error ε reduces to mean estimation in $\|\cdot\|_B$ with error $\frac{a}{2b}\varepsilon$ (or error $\frac{a}{b}\varepsilon$ when $d_1 = d_2$).*

Proof. Suppose there exists an statistical algorithm $\mathcal{A}$ that for any input distribution supported on $\mathcal{B}_{\|\cdot\|_B}$ computes $\tilde{z} \in \mathbb{R}^{d_2}$ satisfying $\|\tilde{z} - \mathbf{E}_{\mathbf{z}}[\mathbf{z}]\|_B \leq \frac{a}{2b}\varepsilon$.

Let D be the target distribution on $\mathbb{R}^{d_1}$, which is supported on $\mathcal{B}_{\|\cdot\|_A}$. We use $\mathcal{A}$ on the image of D by T, multiplied by a. That is, we replace each query $\phi : \mathbb{R}^{d_2} \to \mathbb{R}$ of $\mathcal{A}$ with query $\phi'(w) = \phi(a \cdot Tw)$. Notice that by our assumption, $\|a \cdot Tw\|_B \leq \|w\|_A \leq 1$. Let $\tilde{y}$ be the output of $\mathcal{A}$ divided by a. By linearity, we have that $\|\tilde{y} - T\bar{w}\|_B \leq \frac{1}{2b}\varepsilon$. Let $\tilde{w}$ be any vector such that $\|\tilde{y} - T\tilde{w}\|_B \leq \frac{1}{2b}\varepsilon$. Then,

$$\|\tilde{w} - \bar{w}\|_A \leq b\|T\tilde{w} - T\bar{w}\|_B \leq b\|\tilde{y} - T\tilde{w}\|_B + b\|\tilde{y} - T\bar{w}\|_B \leq \varepsilon.$$

Note that if $d_1 = d_2$ then T is invertible and we can use $\tilde{w} = T^{-1}\tilde{y}$. □

Remark 3.3. *The reduction of Lemma 3.2 is computationally efficient when the following two tasks can be performed efficiently: computing Tw for any input w, and given $z \in \mathbb{R}^{d_2}$ such that there exists $w' \in \mathbb{R}^{d_1}$ with $\|z - Tw'\|_B \leq \delta$, computing w such that $\|z - Tw\|_B \leq \delta + \xi$, for some precision $\xi = O(\delta)$.*

An immediate implication of this is that if the Banach-Mazur distance between unit balls of two norms $\mathcal{W}_1$ and $\mathcal{W}_2$ is r then mean estimation over $\mathcal{W}_1$ with error ε can be reduced to mean estimation over $\mathcal{W}_2$ with error ε/r.

3.1 ℓ_q Mean Estimation

We now consider stochastic linear optimization over $\mathcal{B}_p^d$ and the corresponding ℓ_q mean estimation problem. We first observe that for $q = \infty$ the problem can be solved by directly using coordinate-wise statistical queries with tolerance ε. This is true since each coordinate has range $[-1, 1]$ and for an estimate $\tilde{w}$ obtained in this way we have $\|\tilde{w} - \bar{w}\|_\infty = \max_i\{|\tilde{w}_i - \mathbf{E}[\mathbf{w}_i]\} \leq \varepsilon$.

Theorem 3.4. *ℓ_∞ mean estimation problem with error ε can be efficiently solved using d queries to STAT(ε).*

A simple application of Theorem 3.4 is to obtain an algorithm for ℓ_1 mean estimation. Assume that d is a power of two and let H be the orthonormal Hadamard transform matrix (if d is not a power of two we can first pad the input distribution to to $\mathbb{R}^{d'}$, where $d' = 2^{\lceil \log d \rceil} \leq 2d$). Then it is easy to verify that for every $w \in \mathbb{R}^d$, $\|Hw\|_\infty \leq \|w\|_1 \leq \sqrt{d}\|Hw\|_\infty$. By Lemma 3.2 this directly implies the following algorithm:

Theorem 3.5. *ℓ_1 mean estimation problem with error ε can be efficiently solved using $2d$ queries to STAT$(\varepsilon/\sqrt{2d})$.*

We next deal with an important case of ℓ_2 mean estimation. It is not hard to see that using statistical queries for direct coordinate-wise estimation will require estimation complexity of $\Omega(d/\varepsilon^2)$. We describe two algorithms for this problem with (nearly) optimal estimation complexity. The first one relies on so called Kashin's representations introduced by Lyubarskii and Vershynin [65]. The second is a simpler but slightly less efficient method based on truncated coordinate-wise estimation in a randomly rotated basis.

3.1.1 ℓ_2 Mean Estimation via Kashin's representation

A Kashin's representation is a representation of a vector in an overcomplete linear system such that the magnitude of each coefficient is small (more precisely, within a constant of the optimum) [65]. Such representations, also referred to as "democratic", have a variety of applications including vector quantization and peak-to-average power ratio reduction in communication systems (*cf.* [87]). We show that existence of such representation leads directly to SQ algorithms for ℓ_2 mean estimation.

We start with some requisite definitions.

Definition 3.6. *A sequence $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ is a* tight frame[2] *if for all $w \in \mathbb{R}^d$,*

$$\|w\|_2^2 = \sum_{j=1}^N |\langle w, u_i \rangle|^2.$$

The redundancy of a frame is defined as $\lambda \doteq N/d \geq 1$.

An easy to prove property of a tight frame (see Obs. 2.1 in [65]) is that for every frame representation $w = \sum_{j=1}^N a_i u_i$ it holds that $\sum_{j=1}^N a_i^2 \leq \|w\|_2^2$.

Definition 3.7. *Consider a sequence $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ and $w \in \mathbb{R}^d$. An expansion $w = \sum_{i=1}^N a_i u_i$ such that $\|a\|_\infty \leq \frac{K}{\sqrt{N}}\|w\|_2$ is referred to as a Kashin's representation of w with level K.*

Theorem 3.8 ([65])**.** *For all $\lambda = N/d > 1$ there exists a tight frame $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ in which every $w \in \mathbb{R}^d$ has a Kashin's representation of w with level K for some constant K depending only on λ. Moreover, such a frame can be computed in (randomized) polynomial time.*

The existence of such frames follows from Kashin's theorem [53]. Lyubarskii and Vershynin [65] show that any frame that satisfies a certain uncertainty principle (which itself is implied by the well-studied Restricted Isometry Property) yields a Kashin's representation for all $w \in \mathbb{R}^d$. In particular, various random choices of u_j's have this property with high probability. Given a vector w, a Kashin's representation of w for level K can be computed efficiently (whenever it exists) by solving a convex program. For frames that satisfy the above mentioned uncertainty principle a Kashin's representation can also be found using a simple algorithm that involves $\log(N)$ multiplications of a vector by each of u_j's. Other algorithms for the task are discussed in [87].

[2] In [65] complex vector spaces are considered but the results also hold in the real case.

Theorem 3.9. *For every d there is an efficient algorithm that solves ℓ_2 mean estimation problem (over $\mathcal{B}_2^d$) with error ε using $2d$ queries to STAT$(\Omega(\varepsilon))$.*

Proof. For $N = 2d$ let $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ be a frame in which every $w \in \mathbb{R}^d$ has a Kashin's representation of w with level $K = O(1)$ (as implied by Theorem 3.8). For a vector $w \in \mathbb{R}^d$ let $a(w) \in \mathbb{R}^N$ denote the coefficient vector of some specific Kashin's representation of w (*e.g.* that computed by the algorithm in [65]). Let $\mathbf{w}$ be a random variable supported on $\mathcal{B}_2^d$ and let $\bar{a}_j \doteq \mathbf{E}[a(\mathbf{w})_j]$. By linearity of expectation, $\bar{w} = \mathbf{E}[\mathbf{w}] = \sum_{j=1}^N \bar{a}_j u_j$.

For each $j \in [N]$, let $\phi_j(w) \doteq \frac{\sqrt{N}}{K} \cdot a(w)_j$. Let $\tilde{a}_j$ denote the answer of STAT(ε/K) to query ϕ_j multiplied by $\frac{K}{\sqrt{N}}$. By the definition of Kashin's representation with level K, the range of ϕ_j is $[-1, 1]$ and, by the definition of STAT(ε/K), we have that $|\bar{a}_j - \tilde{a}_j| \leq \frac{\varepsilon}{\sqrt{N}}$ for every $j \in [N]$. Let $\tilde{w} \doteq \sum_{j=1}^N \tilde{a}_j u_j$.

Then by the property of tight frames mentioned above,

$$\|\bar{w} - \tilde{w}\|_2 = \left\| \sum_{j=1}^N (\bar{a}_j - \tilde{a}_j) u_j \right\|_2 \leq \sqrt{\sum_{j=1}^N (\bar{a}_j - \tilde{a}_j)^2} \leq \varepsilon.$$

□

3.1.2 ℓ_2 Mean Estimation using a Random Basis

We now show a simple to analyze randomized algorithm that achieves dimension independent estimation complexity for ℓ_2 mean estimation. The algorithm will use coordinate-wise estimation in a randomly and uniformly chosen basis. We show that for such a basis simply truncating coefficients that are too large will, with high probability, have only a small effect on the estimation error.

More formally, we define the truncation operation as follows. For a real value z and $a \in \mathbb{R}^+$, let

$$m_a(z) := \begin{cases} z & \text{if } |z| \leq a \\ a & \text{if } z > a \\ -a & \text{if } z < -a. \end{cases}$$

For a vector $w \in \mathbb{R}^d$ we define $m_a(w)$ as the coordinate-wise application of m_a to w. For a $d \times d$ matrix U we define $m_{U,a}(w) \doteq U^{-1} m_a(Uw)$ and define $r_{U,a}(w) \doteq w - m_{U,a}(w)$. The key step of the analysis is the following lemma:

Lemma 3.10. *Let $\mathbf{U}$ be an orthogonal matrix chosen uniformly at random and $a > 0$. For every w, with $\|w\|_2 = 1$, $\mathbf{E}[\|r_{\mathbf{U},a}(w)\|_2^2] \leq 4e^{-da^2/2}$.*

Proof. Notice that $\|r_{\mathbf{U},a}(w)\|_2 = \|\mathbf{U}w - m_a(\mathbf{U}w)\|_2$. It is therefore sufficient to analyze $\|\mathbf{u} - m_a(\mathbf{u})\|_2$ for $\mathbf{u}$ a random uniform vector of length 1. Let $\mathbf{r} \doteq \mathbf{u} - m_a(\mathbf{u})$. For each i,

$$\begin{aligned} \mathbf{E}[\mathbf{r}_i^2] &= \int_0^\infty 2t\ \mathbf{Pr}[|\mathbf{r}_i| > t]\, dt = \int_0^\infty 2t \left\{ \mathbf{Pr}[\mathbf{r}_i > t] + \mathbf{Pr}[\mathbf{r}_i < -t] \right\} dt \\ &= \int_0^\infty 4t\ \mathbf{Pr}[\mathbf{r}_i > t]\, dt = \int_0^\infty 4t\ \mathbf{Pr}[\mathbf{u}_i - a > t]\, dt \\ &= 4 \left\{ \int_0^\infty (t+a)\, \mathbf{Pr}[\mathbf{u}_i > t + a]\, dt - a \int_0^\infty \mathbf{Pr}[\mathbf{u}_i > t + a]\, dt \right\} \\ &\leq 4 \frac{e^{-da^2/2}}{d}, \end{aligned}$$

where we have used the symmetry of $\mathbf{r}_i$ and concentration on the unit sphere. From this we obtain $\mathbf{E}[\|\mathbf{r}\|_2^2] \leq 4e^{-da^2/2}$, as claimed.

□

From this lemma is easy to obtain the following algorithm.

Theorem 3.11. *There is an efficient randomized algorithm that solves the ℓ_2 mean estimation problem with error ε and success probability $1-\delta$ using $O(d\log(1/\delta))$ queries to STAT$(\Omega(\varepsilon/\log(1/\varepsilon)))$.*

Proof. Let $\mathbf{w}$ be a random variable supported on $\mathcal{B}_2^d$. For an orthonormal $d \times d$ matrix U, and for $i \in [d]$, let $\phi_{U,i}(w) = (m_a(Uw))_i/a$ (for some a to be fixed later). Let v_i be the output of STAT$(\varepsilon/[2\sqrt{d}a])$ for query $\phi_{U,i} : \mathcal{W} \to [-1,1]$, multiplied by a. Now, let $\tilde{w}_{U,a} \doteq U^{-1}v$, and let $\bar{w}_{U,a} \doteq \mathbf{E}[m_{U,a}(\mathbf{w})]$. This way,

$$\begin{aligned}
\|\bar{w} - \tilde{w}_{U,a}\|_2 &\leq \|\bar{w} - \bar{w}_{U,a}\|_2 + \|\bar{w}_{U,a} - \tilde{w}_{U,a}\|_2 \\
&\leq \|\bar{w} - \bar{w}_{U,a}\|_2 + \|\,\mathbf{E}[m_a(U\mathbf{w})] - v\|_2 \\
&\leq \|\bar{w} - \bar{w}_{U,a}\|_2 + \varepsilon/2.
\end{aligned}$$

Let us now bound the norm of $\mathbf{v} \doteq \bar{w} - \bar{w}_{\mathbf{U},a}$ where $\mathbf{U}$ is a randomly and uniformly chosen orthonormal $d \times d$ matrix. By Chebyshev's inequality:

$$\mathbf{Pr}[\|\mathbf{v}\|_2 \geq \varepsilon/2] \leq 4\frac{\mathbf{E}[\|\mathbf{v}\|_2^2]}{\varepsilon^2} \leq \frac{16\exp(-da^2/2)}{\varepsilon^2}.$$

Notice that to bound the probability above by δ we may choose $a = \sqrt{2\ln(16/(\delta\varepsilon^2))/d}$. Therefore, the queries above require querying STAT$(\varepsilon/[2\sqrt{2\ln(16/\delta\varepsilon^2)}])$, and they guarantee to solve the ℓ_2 mean estimation problem with probability at least $1-\delta$.

Finally, we can remove the dependence on δ in STAT queries by confidence boosting. Let $\varepsilon' = \varepsilon/3$ and $\delta' = 1/8$, and run the algorithm above with error ε' and success probability $1-\delta'$ for $\mathbf{U}_1, \ldots, \mathbf{U}_k$ i.i.d. random orthogonal matrices. If we define $\tilde{w}^1, \ldots, \tilde{w}^k$ the outputs of the algorithm, we can compute the (high-dimensional) median $\tilde{w}$, namely the point $\tilde{w}^j$ whose median ℓ_2 distance to all the other points is the smallest. It is easy to see that (*e.g.* [68, 48])

$$\mathbf{Pr}[\|\tilde{w} - \bar{w}\|_2 > \varepsilon] \leq e^{-Ck},$$

where $C > 0$ is an absolute constant.

Hence, as claimed, it suffices to choose $k = O(\log(1/\delta))$, which means using $O(d\log(1/\delta))$ queries to STAT$(\Omega(\varepsilon/\log(1/\varepsilon))$, to obtain success probability $1-\delta$. □

3.1.3 ℓ_q Mean Estimation for $q > 2$

We now demonstrate that by using the results for ℓ_∞ and ℓ_2 mean estimation we can get algorithms for ℓ_q mean estimation with nearly optimal estimation complexity.

The idea of our approach is to decompose each point into a sum of at most $\log d$ points each of which has a small "dynamic range" of non-zero coordinates. This property ensures a very tight relationship between the ℓ_∞, ℓ_2 and ℓ_q norms of these points allowing us to estimate their mean with nearly optimal estimation complexity. More formally we will rely on the following simple lemma.

Lemma 3.12. *For any $x \in \mathbb{R}^d$ and any two $0 < p < r$:*

1. $\|x\|_r \leq \|x\|_\infty^{1-p/r} \cdot \|x\|_p^{p/r}$;

2. *Let $a = \min_{i\in[d]}\{x_i \mid x_i \neq 0\}$. Then $\|x\|_p \leq a^{1-r/p} \cdot \|x\|_r^{r/p}$.*

Proof. 1.

$$\|x\|_r^r = \sum_{i=1}^d |x_i|^r \leq \sum_{i=1}^d \|x\|_\infty^{r-p} \cdot |x_i|^p = \|x\|_\infty^{r-p} \cdot \|x\|_p^p$$

2.

$$\|x\|_r^r = \sum_{i=1}^d |x_i|^r \geq \sum_{i=1}^d a^{r-p} \cdot |x_i|^p = a^{r-p} \cdot \|x\|_p^p.$$

□

Theorem 3.13. *For any $q \in (2, \infty)$ and $\varepsilon > 0$, ℓ_q mean estimation with error ε can be solved using $3d \log d$ queries to STAT$(\varepsilon / \log(d))$.*

Proof. Let $k \doteq \lfloor \log(d)/q \rfloor - 2$. For $w \in \mathbb{R}^d$, and $j = 0, \ldots, k$ we define

$$R_j(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{2^{-(j+1)} < |w_i| \leq 2^{-j}\}},$$

and $R_\infty(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{|w_i| \leq 2^{-(k+1)}\}}$. It is easy to see that if $w \in \mathcal{B}_q$ then $w = \sum_{j=0}^k R_j(w) + R_\infty(w)$. Furthermore, observe that $\|R_j(w)\|_\infty \leq 2^{-j}$, and by Lemma 3.12, $\|R_j(w)\|_2 \leq 2^{-(j+1)(1-q/2)}$. Finally, let $\bar{w}^j = \mathbf{E}[R_j(\mathbf{w})]$, and $\bar{w}^\infty = \mathbf{E}[R_\infty(\mathbf{w})]$.

Let $\varepsilon' \doteq 2^{2/q-3}\varepsilon/(k+1)$. For each level $j = 0, \ldots, k$, we perform the following queries:

- By using $2d$ queries to STAT$(\Omega(\varepsilon'))$ we obtain a vector $\tilde{w}^{2,j}$ such that $\|\tilde{w}^{2,j} - \bar{w}^j\|_2 \leq 2^{(\frac{q}{2}-1)(j+1)}\varepsilon'$. For this, simply observe that $R_j(\mathbf{w})/[2^{(\frac{q}{2}-1)(j+1)}]$ is supported on $\mathcal{B}_2^d$, so our claim follows from Theorem 3.9.

- By using d queries to STAT(ε') we obtain a vector $\tilde{w}^{\infty,j}$ such that $\|\tilde{w}^{\infty,j} - \bar{w}^j\|_\infty \leq 2^{-j}\varepsilon'$. For this, notice that $R_j(\mathbf{w})/[2^{-j}]$ is supported on $\mathcal{B}_\infty^d$ and appeal to Theorem 3.4.

We consider the following feasibility problem, which is always solvable (e.g., by $\bar{w}^j$)

$$\|\tilde{w}^{\infty,j} - w\|_\infty \leq 2^{-j}\varepsilon', \quad \|\tilde{w}^{2,j} - w\|_2 \leq 2^{(\frac{q}{2}-1)(j+1)}\varepsilon'.$$

Notice that this problem can be solved easily (we can minimize ℓ_2 distance to $\tilde{w}^{2,j}$ with the ℓ_∞ constraint above, and this minimization problem can be solved coordinate-wise), so let $\tilde{w}^j$ be a solution. By the triangle inequality, $\tilde{w}^j$ satisfies $\|\tilde{w}^j - \bar{w}^j\|_\infty \leq 2^{-j}(2\varepsilon')$, and $\|\tilde{w}^j - \bar{w}^j\|_2 \leq 2^{(\frac{q}{2}-1)(j+1)}(2\varepsilon')$.

By Lemma 3.12,

$$\|\tilde{w}^j - \bar{w}^j\|_q \leq \|\tilde{w}^j - \bar{w}^j\|_2^{2/q} \cdot \|\tilde{w}^j - \bar{w}^j\|_\infty^{1-2/q} \leq 2^{(1-2/q)(j+1)}\, 2^{-j(1-2/q)}(2\varepsilon') = \varepsilon/[2(k+1)].$$

Next we estimate $\bar{w}^\infty$. Since $2^{-(k+1)} = 2^{-\lfloor \ln d/q \rfloor + 1} \leq 4d^{-1/q}$, by using d queries to STAT$(\varepsilon/8)$ we can estimate each coordinate of $\bar{w}^\infty$ with accuracy $\varepsilon/[2d^{1/q}]$ and obtain $\tilde{w}^\infty$ satisfying $\|\tilde{w}^\infty - \bar{w}^\infty\|_q \leq d^{1/q}\|\tilde{w}^\infty - \bar{w}^\infty\|_\infty \leq \varepsilon/2$. Let now $\tilde{w} = [\sum_{j=0}^k \tilde{w}^j] + \tilde{w}^\infty$. We have,

$$\|\tilde{w} - \bar{w}\|_q \;\leq\; \sum_{j=0}^k \|\tilde{w}^j - \bar{w}^j\|_q + \|\tilde{w}^\infty - \bar{w}^\infty\|_q \;\leq\; (k+1)\frac{\varepsilon}{2(k+1)} + \frac{\varepsilon}{2} \;=\; \varepsilon.$$

□

3.1.4 ℓ_q Mean Estimation for $q \in (1,2)$

Finally, we consider the case when $q \in (1,2)$. Here we get the nearly optimal estimation complexity via two bounds.

The first bound follows from the simple fact that for all $w \in \mathbb{R}^d$, $\|w\|_2 \leq \|w\|_q \leq d^{1/q-1/2}\|w\|_2$. Therefore we can reduce ℓ_q mean estimation with error ε to ℓ_2 mean estimation with error $\varepsilon/d^{1/q-1/2}$ (this is a special case of Lemma 3.2 with the identity embedding). Using Theorem 3.9 we then get the following theorem.

Theorem 3.14. *For $q \in (1,2)$ and every d there is an efficient algorithm that solves ℓ_q mean estimation problem with error ε using $2d$ queries to STAT$(\Omega(d^{1/2-1/q}\varepsilon))$.*

It turns out that for large ε better sample complexity can be achieved using a different algorithm. Achieving (nearly) optimal estimation complexity in this case requires the use of VSTAT oracle. (The estimation complexity for STAT is quadratically worse. That still gives an improvement over Theorem 3.14 for some range of values of ε.) In in the case of $q > 2$, our algorithm decompose each point into a sum of at most $\log d$ points each of which has a small "dynamic range" of non-zero coordinates. For each component we can then use coordinate-wise estimation with an additional zeroing of coordinates that are too small. Such zeroing ensures that the estimate does not accumulate large error from the coordinates where the mean of the component itself is close to 0.

Theorem 3.15. *For any $q \in (1,2)$ and $\varepsilon > 0$, the ℓ_q mean estimation problem can be solved with error ε using $2d\log d$ queries to VSTAT$((16\log(d)/\varepsilon)^p)$.*

Proof. Given $w \in \mathcal{B}_q$ we consider its positive and negative parts: $w = w^+ - w^-$, where $w^+ \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{w_i \geq 0\}}$ and $w^- \doteq -\sum_{i=1}^d e_i w_i \mathbf{1}_{\{w_i<0\}}$. We again rely on the decomposition of w into "rings" of dynamic range 2, but now for its positive and negative parts. Namely, $w = \sum_{j=0}^k [R_j(w^+) - R_j(w^-)] + [R_\infty(w^+) - R_\infty(w^-)]$, where $k \doteq \lfloor \log(d)/q \rfloor - 2$, $R_j(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{2^{-(j+1)}<|w_i|\leq 2^{-j}\}}$ and $R_\infty(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{|w_i|\leq 2^{-k-1}\}}$.

Let $\mathbf{w}$ be a random variable supported on $\mathcal{B}_q^d$. Let $\varepsilon' \doteq \varepsilon/(2k+3)$. For each level $j = 0,\ldots,k$, we now describe how to estimate $\overline{w^{+,j}} = \mathbf{E}[R_j(\mathbf{w}^+)]$ with accuracy ε'. The estimation is essentially just coordinate-wise use of VSTAT with zeroing of coordinates that are too small. Let v_i' be the value returned by VSTAT(n) for query $\phi_i(w) = 2^j \cdot (R_j(w^+))_i$, where $n = (\varepsilon'/8)^{-p} \leq (16\log(d)/\varepsilon)^p$. Note that $2^j \cdot (R_j(w^+))_i \in [0,1]$ for all w and j. Further, let $v_i = v_i' \cdot \mathbf{1}_{\{|v_i'|\geq 2/n\}}$. We start by proving the following decomposition of the error of v.

Lemma 3.16. *Let $u \doteq 2^j \cdot \overline{w^{+,j}}$, and $z \doteq u - v$. Then $\|z\|_q^q \leq \|u^<\|_q^q + n^{-q/2} \cdot \|u^>\|_{q/2}^{q/2}$, where $u_i^< = u_i \cdot \mathbf{1}_{\{u_i<4/n\}}$ and $u_i^> = u_i \cdot \mathbf{1}_{\{u_i \geq 1/n\}}$ and for all i.*

Proof. For every index $i \in [d]$ we consider two cases. The first case is when $v_i = 0$. By the definition of v_i, we know that $v_i' < 2/n$. This implies that $u_i = 2^j\,\mathbf{E}[(R_j(\mathbf{w}^+))_i] < 4/n$. This is true since, otherwise (when $u_i \geq 4/n$), by the guarantees of VSTAT(n), we would have $|v_i' - u_i| \leq \sqrt{\frac{u_i}{n}}$ and $v_i' \geq u_i - \sqrt{\frac{u_i}{n}} \geq 2/n$. Therefore in this case, $u_i = u_i^<$ and $z_i = u_i - v_i = u_i^<$.

In the second case $v_i \neq 0$. In this case we have that $v_i' \geq 2/n$. This implies that $u_i \geq 1/n$. This is true since, otherwise (when $u_i < 1/n$), by the guarantees of VSTAT(n), we would have $|v_i' - u_i| \leq \sqrt{\frac{u_i}{n}}$ and $v_i' \leq u_i + \frac{1}{n} < 2/n$. Therefore in this case, $u_i = u_i^>$ and $z_i = u_i - v_i'$. By the guarantees of VSTAT(n), $|z_i| = |u_i^> - v_i'| \leq \max\left\{\frac{1}{n}, \sqrt{\frac{u_i^>}{n}}\right\} = \sqrt{\frac{u_i^>}{n}}$.

The claim now follows since by combining these two cases we get $|z_i|^q \leq (u_i^<)^q + \left(\frac{u_i^>}{n}\right)^{q/2}$. □

We next observe that by Lemma 3.12, for every $w \in \mathcal{B}_q^d$,

$$\|R_j(w^+)\|_1 \leq (2^{-j-1})^{1-q} \|R_j(w^+)\|_q^q \leq (2^{-j-1})^{1-q}.$$

This implies that

$$\|u\|_1 = 2^j \cdot \left\|\overline{w^{+,j}}\right\|_1 = 2^j \cdot \left\|\mathbf{E}[R_j(\mathbf{w}^+)]\right\|_1 \leq 2^j \cdot (2^{-j-1})^{1-q} = 2^{(j+1)q-1}. \tag{2}$$

Now by Lemma 3.12 and eq.(2), we have

$$\|u^<\|_q^q \leq \left(\frac{4}{n}\right)^{q-1} \cdot \|u^<\|_1 = n^{1-q} \cdot 2^{(j+3)q-3}. \tag{3}$$

Also by Lemma 3.12 and eq.(2), we have

$$\|u^>\|_{q/2}^{q/2} \leq \left(\frac{1}{n}\right)^{q/2-1} \cdot \|u^>\|_1 \leq n^{1-q/2} \cdot 2^{(j+1)q-1}. \tag{4}$$

Substituting eq. (3) and eq. (4) into Lemma 3.16 we get

$$\|z\|_q^q \leq \|u^<\|_q^q + n^{-q/2} \cdot \|u^>\|_{q/2}^{q/2} \leq n^{1-q} \cdot \left(2^{(j+3)q-3} + 2^{(j+1)q-1}\right) \leq n^{1-q} \cdot 2^{(j+3)q}.$$

Let $\tilde{w}^{+,j} \doteq 2^{-j} v$. We have

$$\left\|\overline{w^{+,j}} - 2^{-j} v\right\|_q = 2^{-j} \cdot \|z\|_q \leq 2^3 \cdot n^{1/q-1} = \varepsilon'.$$

We obtain an estimate of $\overline{w^{-,j}}$ in an analogous way. Finally, to estimate, $\bar{w}^\infty \doteq \mathbf{E}[R_\infty(\mathbf{w})]$ we observe that $2^{-k-1} \leq 2^{1-\lfloor \log(d)/q \rfloor} \leq 4d^{-1/q}$. Now using VSTAT$(1/(4\varepsilon')^2)$ we can obtain an estimate of each coordinate of $\bar{w}^\infty$ with accuracy $\varepsilon' \cdot d^{-1/q}$. In particular, the estimate $\tilde{w}^\infty$ obtained in this way satisfies $\|\bar{w}^\infty - \tilde{w}^\infty\|_q \leq \varepsilon'$.

Now let $\tilde{w} = \sum_{j=0}^k (\tilde{w}^{+,j} - \tilde{w}^{-,j}) + \tilde{w}^\infty$. Each of the estimates has ℓ_q error of at most $\varepsilon' = \varepsilon/(2k+3)$ and therefore the total error is at most ε. □

3.1.5 General Convex Bodies

Next we consider mean estimation and stochastic linear optimization for convex bodies beyond ℓ_p-balls. A first observation is that Theorem 3.4 can be easily generalized to origin-symmetric polytopes. The easiest way to see the result is to use the standard embedding of the origin-symmetric polytope norm into ℓ_∞ and appeal to Lemma 3.2.

Corollary 3.17. *Let $\mathcal{W}$ be an origin-symmetric polytope with $2m$ facets. Then mean estimation over $\mathcal{W}$ with error ε can be efficiently solved using m queries to STAT$(\varepsilon/2)$.*

In the case of an arbitrary origin-symmetric convex body $\mathcal{W} \subseteq \mathbb{R}^d$, we can reduce mean estimation over $\mathcal{W}$ to ℓ_2 mean estimation using the John ellipsoid. Such an ellipsoid $\mathcal{E}$ satisfies the inclusions $\frac{1}{\sqrt{d}}\mathcal{E} \subseteq \mathcal{W} \subseteq \mathcal{E}$ and any ellipsoid is linearly isomorphic to a unit ℓ_2 ball. Therefore appealing to Lemma 3.2 and Theorem 3.9 we have the following.

Theorem 3.18. *Let $\mathcal{W} \subseteq \mathbb{R}^d$ an origin-symmetric convex body. Then the mean estimation problem over $\mathcal{W}$ can be solved using $2d$ queries to STAT$(\Omega(\varepsilon/\sqrt{d}))$.*

By Observation 3.1, for an arbitrary convex body $\mathcal{K}$, the stochastic linear optimization problem over $\mathcal{K}$ reduces to mean estimation over $\mathcal{W} \doteq \mathrm{conv}(\mathcal{K}_*, -\mathcal{K}_*)$. This leads to a nearly-optimal (in terms of worst-case dimension dependence) estimation complexity. A matching lower bound for this task will be proved in Corollary 3.22.

A drawback of this approach is that it depends on knowledge of the John ellipsoid for $\mathcal{W}$, which is, in general, cannot be computed efficiently (*e.g.* [11]). However, if $\mathcal{K}$ is a polytope with a polynomial number of facets, then $\mathcal{W}$ is an origin-symmetric polytope with a polynomial number of vertices, and the John ellipsoid can be computed in polynomial time [56]. From this, we conclude that

Corollary 3.19. *Then there exists an efficient algorithm that given as input the vertices of an origin-symmetric polytope $\mathcal{W} \subseteq \mathbb{R}^d$ solves the mean estimation problem over $\mathcal{W}$ using $2d$ queries to STAT$(\Omega(\varepsilon/\sqrt{d}))$. The algorithm runs in time polynomial in the number of vertices.*

3.2 Lower Bounds

We now prove lower bounds for stochastic linear optimization over the ℓ_p unit ball and consequently also for ℓ_q mean estimation. We do this using the technique from [37] that is based on bounding the statistical dimension with discrimination norm. The *discrimination norm* of a set of distributions $\mathcal{D}'$ relative to a distribution D is denoted by $\kappa_2(\mathcal{D}', D)$ and defined as follows:

$$\kappa_2(\mathcal{D}', D) \doteq \max_{h:X\to\mathbb{R}, \|h\|_D=1} \left\{ \mathop{\mathbf{E}}_{D'\sim\mathcal{D}'} \left[\left| \mathop{\mathbf{E}}_{D'}[h] - \mathop{\mathbf{E}}_{D}[h] \right| \right] \right\},$$

where the norm of h over D is $\|h\|_D = \sqrt{\mathbf{E}_D[h^2(x)]}$ and $D' \sim \mathcal{D}'$ refers to choosing D' randomly and uniformly from the set $\mathcal{D}'$.

Let $\mathcal{B}(\mathcal{D}, D)$ denote the decision problem in which given samples from an unknown input distribution $D' \in \mathcal{D} \cup \{D\}$ the goal is to output 1 if $D' \in \mathcal{D}$ and 0 if $D' = D$.

Definition 3.20 ([36]). *For $\kappa > 0$, domain X and a decision problem $\mathcal{B}(\mathcal{D}, D)$, let t be the largest integer such that there exists a finite set of distributions $\mathcal{D}_D \subseteq \mathcal{D}$ with the following property: for any subset $\mathcal{D}' \subseteq \mathcal{D}_D$, where $|\mathcal{D}'| \geq |\mathcal{D}_D|/t$, $\kappa_2(\mathcal{D}', D) \leq \kappa$. The **statistical dimension** with discrimination norm κ of $\mathcal{B}(\mathcal{D}, D)$ is t and denoted by* $\mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$.

The statistical dimension with discrimination norm κ of a problem over distributions gives a lower bound on the complexity of any statistical algorithm.

Theorem 3.1 ([36]). *Let X be a domain and $\mathcal{B}(\mathcal{D}, D)$ be a decision problem over a class of distributions $\mathcal{D}$ on X and reference distribution D. For $\kappa > 0$, let $t = \mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$. Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\geq 2/3$ requires $t/3$ calls to VSTAT$(1/(3 \cdot \kappa^2))$.*

We now reduce a simple decision problem to stochastic linear optimization over the ℓ_p unit ball. Let $E = \{e_i \mid i \in [d]\} \cup \{-e_i \mid i \in [d]\}$. Let the reference distribution D be the uniform distribution over E. For a vector $v \in [-1,1]^d$, let D_v denote the following distribution: pick $i \in [d]$ randomly and uniformly, then pick $b \in \{-1, 1\}$ randomly subject to the expectation being equal to v_i and output $b \cdot e_i$. By definition, $\mathbf{E}_{\mathbf{w}\sim D_v}[\mathbf{w}] = \frac{1}{d}v$. Further D_v is supported on $E \subset \mathcal{B}_q^d$.

For $q \in [1,2]$, $\alpha \in [0,1]$ and every $v \in \{-1,1\}^d$, $d^{1/q-1} \cdot v \in \mathcal{B}_p^d$ and $\langle d^{1/q-1}v, \mathbf{E}_{\mathbf{w}\sim D_{\alpha v}}[\mathbf{w}]\rangle = \alpha \cdot d^{1/q-1}$. At the same time for the reference distribution D and every $x \in \mathcal{B}_p^d$, we have that $\langle x, \mathbf{E}_{\mathbf{w}\sim D}[\mathbf{w}]\rangle = 0$. Therefore to optimize with accuracy $\varepsilon = \alpha d^{1/q-1}/2$ it is necessary distinguish every distribution in $\mathcal{D}_\alpha$ from D, in other words to solve the decision problem $\mathcal{B}(\mathcal{D}_\alpha, D)$.

Lemma 3.21. *For any $r > 0$, $2^{\Omega(r)}$ queries to VSTAT$(d/(r\alpha^2))$ are necessary to solve the decision problem $\mathcal{B}(\mathcal{D}_\alpha, D)$ with success probability at least $2/3$.*

Proof. We first observe that for any function $h : \mathcal{B}_1^d \to \mathbb{R}$,

$$\mathop{\mathbf{E}}_{D_{\alpha v}}[h] - \mathop{\mathbf{E}}_{D}[h] = \frac{\alpha}{2d} \sum_{i \in [d]} v_i \cdot (h(e_i) - h(-e_i)). \tag{5}$$

Let $\beta = \sqrt{\sum_{i\in[d]}(h(e_i) - h(-e_i))^2}$. By Hoeffding's inequality we have that for every $r > 0$,

$$\mathop{\mathbf{Pr}}_{v\sim\{-1,1\}^d}\left[\left|\sum_{i\in[d]} v_i \cdot (h(e_i) - h(-e_i))\right| \geq r \cdot \beta\right] \leq 2e^{-r^2/2}.$$

This implies that for every set $\mathcal{V} \subseteq \{-1,1\}^d$ such that $|\mathcal{V}| \geq 2^d/t$ we have that

$$\mathop{\mathbf{Pr}}_{v\sim\mathcal{V}}\left[\left|\sum_{i\in[d]} v_i \cdot (h(e_i) - h(-e_i))\right| \geq r \cdot \beta\right] \leq t \cdot 2e^{-r^2/2}.$$

From here a simple manipulation (see Lemma A.4 in [79]) implies that

$$\mathop{\mathbf{E}}_{v\sim\mathcal{V}}\left[\left|\sum_{i\in[d]} v_i \cdot (h(e_i) - h(-e_i))\right|\right] \leq \sqrt{2}(2 + \sqrt{\ln t}) \cdot \beta \leq \sqrt{2\log t} \cdot \beta.$$

Note that

$$\beta \leq \sqrt{\sum_{i\in[d]} 2h(e_i)^2 + 2h(-e_i)^2} = \sqrt{2d} \cdot \|h\|_D.$$

For a set of distributions $\mathcal{D}' \subseteq \mathcal{D}_\alpha$ of size at least $2^d/t$, let $\mathcal{V} \subseteq \{-1,1\}^d$ be the set of vectors in $\{-1,1\}^d$ associated with $\mathcal{D}'$. By eq.(5) we have that

$$\begin{aligned}\mathop{\mathbf{E}}_{D'\sim\mathcal{D}'}\left[\left|\mathop{\mathbf{E}}_{D'}[h] - \mathop{\mathbf{E}}_{D}[h]\right|\right] &= \frac{\alpha}{2d}\mathop{\mathbf{E}}_{v\sim\mathcal{V}}\left[\left|\sum_{i\in[d]} v_i \cdot (h(e_i) - h(-e_i))\right|\right] \\ &\leq \frac{\alpha}{2d} 2\sqrt{d\log t} \cdot \|h\|_D = \alpha\sqrt{\log t/d} \cdot \|h\|_D.\end{aligned}$$

By Definition 3.20, this implies that for every $t > 0$, $\mathrm{SDN}(\mathcal{B}(\mathcal{D}_\alpha, D), \alpha\sqrt{\log t/d}) \geq t$. By Theorem 3.1 that for any $r > 0$, $2^{\Omega(r)}$ queries to VSTAT$(d/(r\alpha^2))$ are necessary to solve the decision problem $\mathcal{B}(\mathcal{D}_\alpha, D)$ with success probability at least $2/3$. □

To apply this lemma with our reduction we set $\alpha = 2\varepsilon d^{1-1/q}$. Note that α must be in the range $[0,1]$ so this is possible only if $\varepsilon < d^{1/q-1}/2$. Hence the lemma gives the following corollary:

Corollary 3.22. *For any $\varepsilon \leq d^{1/q-1}/2$ and $r > 0$, $2^{\Omega(r)}$ queries to VSTAT$(d^{2/q-1}/(r\varepsilon^2))$ are necessary to find an ε-optimal solution to the stochastic linear optimization problem over $\mathcal{B}_p^d$ with success probability at least $2/3$. The same lower bound holds for ℓ_q mean estimation with error ε.*

Observe that this lemma does not cover the regime when $q > 1$ and $\varepsilon \geq d^{1/q-1}/2 = d^{-1/p}/2$. We analyze this case via a simple observation that for every $d' \in [d]$, $\mathcal{B}_p^{d'}$ and $\mathcal{B}_q^{d'}$ can be embedded into $\mathcal{B}_p^d$ and $\mathcal{B}_q^d$ respectively in a trivial way: by adding $d - d'$ zero coordinates. Also the mean of the distribution supported on such an embedding of $\mathcal{B}_q^{d'}$ certainly lies inside the embedding. In particular, a d-dimensional solution x can be converted back to a d'-dimensional solution x' without increasing the value achieved by the solution. Hence lower bounds for optimization over $\mathcal{B}_p^{d'}$ imply lower bounds for optimization over $\mathcal{B}_p^d$. Therefore for any $\varepsilon \geq d^{-1/p}/2$, let $d' = (2\varepsilon)^{-p}$ (ignoring for simplicity the minor issues with rounding). Now Corollary 3.22 applied to d' implies that $2^{\Omega(r)}$ queries to VSTAT$((d')^{2/q-1}/(r\varepsilon^2))$ are necessary for stochastic linear optimization. Substituting the value of $d' = (2\varepsilon)^{-p}$ we get $(d')^{2/q-1}/(r\varepsilon^2) = 2^{2-p}/(r\varepsilon^p)$ and hence we get the following corollary.

Corollary 3.23. *For any $q > 1$, $\varepsilon \geq d^{1/q-1}/2$ and $r > 0$, $2^{\Omega(r)}$ queries to VSTAT$(1/(r\varepsilon^p))$ are necessary to find an ε-optimal solution to the stochastic linear optimization problem over $\mathcal{B}_p^d$ with success probability at least $2/3$. The same lower bound holds for ℓ_q mean estimation with error ε.*

These lower bounds are not tight when $q > 2$. In this case a lower bound of $\Omega(1/\varepsilon^2)$ (irrespective of the number of queries) follows from a basic property of VSTAT: no query to VSTAT(n) can distinguish between two input distributions D_1 and D_2 if the total variation distance between D_1^n and D_2^n is smaller than some (universal) positive constant [36].

4 Gradient Descent and Friends

We now describe approaches for solving convex programs by SQ algorithms that are based on the broad literature of inexact gradient methods. We will show that some of the standard oracles proposed in these works can be implemented by SQs; more precisely, by estimation of the mean gradient. This reduces the task of solving a stochastic convex program to a polynomial number of calls to the algorithms for mean estimation from Section 3.

For the rest of the section we use the following notation. Let $\mathcal{K}$ be a convex body in a normed space $(\mathbb{R}^d, \|\cdot\|)$, and let $\mathcal{W}$ be a parameter space (notice we make no assumptions on this set). Unless we explicitly state it, $\mathcal{K}$ is not assumed to be origin-symmetric. Let $R \doteq \max_{x,y\in\mathcal{K}} \|x-y\|/2$, which is the $\|\cdot\|$-radius of $\mathcal{K}$. For a random variable $\mathbf{w}$ supported on $\mathcal{W}$ we consider the stochastic convex optimization problem $\min_{x\in\mathcal{K}} \{F(x) \doteq \mathbf{E_w}[f(x,\mathbf{w})]\}$, where for all $w \in \mathcal{W}$, $f(\cdot, w)$ is convex and subdifferentiable on $\mathcal{K}$. Given $x \in \mathcal{K}$, we denote $\nabla f(x,w) \in \partial f(x,w)$ an arbitrary selection of a subgradient;[3] similarly for F, $\nabla F(x) \in \partial F(x)$ is arbitrary.

Let us make a brief reminder of some important classes of convex functions. We say a subdifferentiable convex function $f : \mathcal{K} \to \mathbb{R}$ is in the class

- $\mathcal{F}(\mathcal{K}, B)$ of B-bounded-range functions if for all $x \in \mathcal{K}$, $|f(x)| \leq B$.
- $\mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$ of L_0-Lipschitz continuous functions w.r.t. $\|\cdot\|$, if for all $x, y \in \mathcal{K}$, $|f(x) - f(y)| \leq L_0\|x-y\|$; this implies
$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + L_0\|y-x\|. \tag{6}$$
- $\mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$ of functions with L_1-Lipschitz continuous gradient w.r.t. $\|\cdot\|$, if for all $x, y \in \mathcal{K}$, $\|\nabla f(x) - \nabla f(y)\|_* \leq L_1\|x-y\|$; this implies
$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L_1}{2}\|y-x\|^2. \tag{7}$$

[3] We omit some necessary technical conditions, *e.g.* measurability, for the gradient selection in the stochastic setting. We refer the reader to [74] for a detailed discussion.

- $\mathcal{S}_{\|\cdot\|}(\mathcal{K},\kappa)$ of κ-strongly convex functions w.r.t. $\|\cdot\|$, if for all $x, y \in \mathcal{K}$

$$f(y) \geq f(x) + \langle \nabla f(x), y-x\rangle + \frac{\kappa}{2}\|y-x\|^2. \tag{8}$$

4.1 SQ Implementation of Approximate Gradient Oracles

Here we present two classes of oracles previously studied in the literature, together with SQ algorithms for implementing them.

Definition 4.1 (Approximate gradient [23]). *Let $F : \mathcal{K} \to \mathbb{R}$ be a convex subdifferentiable function. We say that $\tilde{g} : \mathcal{K} \to \mathbb{R}^d$ is an η-*approximate gradient *of F over $\mathcal{K}$ if for all $u, x, y \in \mathcal{K}$*

$$|\langle \tilde{g}(x) - \nabla F(x), y-u\rangle| \leq \eta. \tag{9}$$

Observation 4.2. *Let $\mathcal{K}_0 \doteq \{x-y \mid x,y \in \mathcal{K}\}$ (which is origin-symmetric by construction), let furthermore $\|\cdot\|_{\mathcal{K}_0}$ be the norm induced by $\mathcal{K}_0$ and $\|\cdot\|_{\mathcal{K}_{0*}}$ its dual norm. Notice that under this notation, (9) is equivalent to $\|\tilde{g}(x) - \nabla F(x)\|_{\mathcal{K}_{0*}} \leq \eta$. Therefore, if $F(x) = \mathbf{E}_{\mathbf{w}}[f(x,\mathbf{w})]$ satisfies for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}^0_{\|\cdot\|_{\mathcal{K}_0}}(\mathcal{K}, L_0)$ then implementing a η-approximate gradient reduces to mean estimation in $\|\cdot\|_{\mathcal{K}_{0*}}$ with error η/L_0.*

Definition 4.3 (Inexact Oracle [25, 24]). *Let $F : \mathcal{K} \to \mathbb{R}$ be a convex subdifferentiable function. We say that $(\tilde{F}(\cdot), \tilde{g}(\cdot)) : \mathcal{K} \to \mathbb{R} \times \mathbb{R}^d$ is a* first-order (η, M, μ)-oracle *of F over $\mathcal{K}$ if for all $x, y \in \mathcal{K}$*

$$\frac{\mu}{2}\|y-x\|^2 \leq F(y) - [\tilde{F}(x) - \langle \tilde{g}(x), y-x\rangle] \leq \frac{M}{2}\|y-x\|^2 + \eta. \tag{10}$$

An important feature of this oracle is that the error for approximating the gradient is *independent of the radius*. This observation was established by Devolder et al. [24], and the consequences for statistical algorithms are made precise in the following lemma.

Lemma 4.4. *Let $\eta > 0$, $0 < \kappa \leq L_1$ and assume that for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B) \cap \mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$ and $F(\cdot) = \mathbf{E}_{\mathbf{w}}[f(\cdot, \mathbf{w})] \in \mathcal{S}_{\|\cdot\|}(\mathcal{K},\kappa) \cap \mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$. Then implementing a first-order (η, M, μ)-oracle (where $\mu = \kappa/2$ and $M = 2L_1$) for F reduces to mean estimation in $\|\cdot\|_*$ with error $\sqrt{\eta\kappa}/[2L_0]$, plus a single query to STAT($\Omega(\eta/B)$). Furthermore, for a first-order method that does not require values of F, the latter query can be omitted.*

If we remove the assumption $F \in \mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$ we can instead use the upper bound $M = 2L_0^2/\eta$.

Proof. We first observe that we can obtain an approximate zero-order oracle for F with error η by a single query to STAT($\Omega(\eta/B)$). In particular, we can obtain a value $\hat{F}(x)$ such that $|\hat{F}(x) - F(x)| \leq \eta/4$, and then use as approximation

$$\tilde{F}(x) = \hat{F}(x) - \eta/2.$$

This way $|F(x) - \tilde{F}(x)| \leq |F(x) - \hat{F}(x)| + |\hat{F}(x) - \tilde{F}(x)| \leq 3\eta/4$, and also $F(x) - \tilde{F}(x) = F(x) - \hat{F}(x) + \eta/2 \geq \eta/4$. Finally, observe that for any gradient method that does not require access to the function value we can skip the estimation of $\tilde{F}(x)$, and simply replace it by $F(x) - \eta/2$ in what comes next.

Next, we prove that an approximate gradient $\tilde{g}(x)$ satisfying

$$\|\nabla F(x) - \tilde{g}(x)\|_* \leq \sqrt{\eta\kappa}/2 \leq \sqrt{\eta L_1}/2, \tag{11}$$

suffices for a (η, μ, M)-oracle, where, $\mu = \kappa/2$, $M = 2L_1$. For convenience, we refer to the first inequality in (10) as the *lower bound* and the second as the *upper bound*.

Lower bound. Since F is κ-strongly convex, and by the lower bound on $F(x)-\tilde{F}(x)$

$$\begin{array}{rcl} F(y) & \geq & F(x)+\langle \nabla F(x), y-x\rangle + \frac{\kappa}{2}\|x-y\|^2 \\ & \geq & \tilde{F}(x)+\eta/4+\langle \tilde{g}(x), y-x\rangle + \langle \nabla F(x)-\tilde{g}(x), y-x\rangle + \frac{\kappa}{2}\|x-y\|^2. \end{array}$$

Thus to obtain the lower bound it suffices prove that for all $y\in\mathbb{R}^d$,

$$\frac{\eta}{4}+\langle \nabla F(x)-\tilde{g}(x), y-x\rangle + \frac{\mu}{2}\|x-y\|^2 \geq 0. \tag{12}$$

In order to prove this inequality, notice that among all y's such that $\|y-x\|=t$, the minimum of the expression above is attained when $\langle \nabla F(x)-\tilde{g}(x), y-x\rangle = -t\|\nabla F(x)-\tilde{g}(x)\|_*$. This leads to the one dimensional inequality

$$\frac{\eta}{4}-t\|\nabla F(x)-\tilde{g}(x)\|_* + \frac{\mu}{2}t^2 \geq 0,$$

whose minimum is attained at $t=\frac{\|\nabla F(x)-\tilde{g}(x)\|_*}{\mu}$, and thus has minimum value $\eta/4-\|\nabla F(x)-\tilde{g}(x)\|_*^2/(2\mu)$. Finally, this value is nonnegative by assumption, proving the lower bound.

Upper bound. Since F has L_1-Lipschitz continuous gradient, and by the bound on $|F(x)-\tilde{F}(x)|$

$$\begin{array}{rcl} F(y) & \leq & F(x)+\langle \nabla F(x), y-x\rangle + \frac{L_1}{2}\|y-x\|^2 \\ & \leq & \tilde{F}(x)+\frac{3\eta}{4}+\langle \tilde{g}(x), y-x\rangle + \langle \nabla F(x)-\tilde{g}(x), y-x\rangle + \frac{L_1}{2}\|x-y\|^2. \end{array}$$

Now we show that for all $y\in\mathbb{R}^d$

$$\frac{L_1}{2}\|y-x\|^2 - \langle \nabla F(x)-\tilde{g}(x), y-x\rangle + \frac{\eta}{4} \geq 0.$$

Indeed, minimizing the expression above in y shows that it suffices to have $\|\nabla F(x)-\tilde{g}(x)\|_*^2 \leq \eta L_1/2$, which is true by assumption.

Finally, combining the two bounds above we get that for all $y\in\mathcal{K}$

$$F(y) \leq [\tilde{F}(x)+\langle \tilde{g}(x), y-x\rangle] + \frac{M}{2}\|y-x\|^2+\eta,$$

which is precisely the upper bound.

As a conclusion, we proved that in order to obtain $\tilde{g}$ for a (η, M, μ)-oracle it suffices to obtain an approximate gradient satisfying (11), which can be obtained by solving a mean estimation problem in $\|\cdot\|_*$ with error $\sqrt{\eta\kappa}/[2L_0]$. This together with our analysis of the zero-order oracle proves the result.

Finally, if we remove the assumption $F\in\mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$ then from (6) we can prove that for all $x,y\in\mathcal{K}$

$$F(y)-[F(x)+\langle \nabla F(x), y-x\rangle] \leq \frac{L_0^2}{\eta}\|x-y\|^2+\frac{\eta}{4},$$

where $M=2L_0^2/\eta$. This is sufficient for carrying out the proof above, and the result follows. □

4.2 Classes of Convex Minimization Problems

We now use known inexact convex minimization algorithms together with our SQ implementation of approximate gradient oracles to solve several classes of stochastic optimization problems. We will see that in terms of estimation complexity there is no significant gain from the non-smooth to the smooth case; however, we can significantly reduce the number of queries by acceleration techniques.

On the other hand, strong convexity leads to improved estimation complexity bounds: The key insight here is that only a local approximation of the gradient around the current query point suffices for methods, as a first order (η, M, μ)-oracle is robust to crude approximation of the gradient at far away points from the query (see Lemma 4.4). We note that both smoothness and strong convexity are required only for the objective function and not for each function in the support of the distribution. This opens up the possibility of applying this algorithm without the need of adding a strongly convex term pointwise –*e.g.* in regularized linear regression– as long as the expectation is strongly convex.

4.2.1 Non-smooth Case: The Mirror-Descent Method

Before presenting the mirror-descent method we give some necessary background on prox-functions. We assume the existence of a subdifferentiable r-uniformly convex function (where $2 \leq r < \infty$) $\Psi : \mathcal{K} \to \mathbb{R}_+$ w.r.t. the norm $\|\cdot\|$, i.e., that satisfies[4] for all $x, y \in \mathcal{K}$

$$\Psi(y) \geq \Psi(x) + \langle \nabla \Psi(x), y - x \rangle + \frac{1}{r}\|y - x\|^r. \tag{13}$$

We will assume w.l.o.g. that $\inf_{x \in \mathcal{K}} \Psi(x) = 0$.

The existence of r-strongly convex functions holds in rather general situations [71], and, in particular, for finite-dimensional ℓ_p^d spaces we have explicit constructions for $r = \min\{2, p\}$ (see Appendix A for details). Let $D_\Psi(\mathcal{K}) \doteq \sup_{x \in \mathcal{K}} \Psi(x)$ be the *prox-diameter of* $\mathcal{K}$ w.r.t. Ψ.

We define the prox-function (a.k.a. Bregman distance) at $x \in \text{int}(\mathcal{K})$ as $V_x(y) = \Psi(y) - \Psi(x) - \langle \nabla \Psi(x), y - x \rangle$. In this case we say the prox-function is based on Ψ proximal setup. Finally, notice that by (13) we have $V_x(y) \geq \frac{1}{r}\|y - x\|^r$.

For the first-order methods in this section we will assume $\mathcal{K}$ is such that for any vector $x \in \mathcal{K}$ and $g \in \mathbb{R}^d$ the *proximal problem* $\min\{\langle g, y - x \rangle + V_x(y) \, : \, y \in \mathcal{K}\}$ can be solved efficiently. For the case $\Psi(\cdot) = \|\cdot\|_2^2$ this corresponds to Euclidean projection, but this type of problems can be efficiently solved in more general situations [68].

The first class of functions we study is $\mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$. We propose to solve problems in this class by the mirror-descent method [68]. This is a classic method for minimization of non-smooth functions, with various applications to stochastic and online learning. Although simple and folklore, we are not aware of a reference on the analysis of the inexact version with proximal setup based on a r-uniformly convex function. Therefore we include its analysis here.

Mirror-descent uses a prox function $V_x(\cdot)$ based on Ψ proximal setup. The method starts querying a gradient at point $x^0 = \arg\min_{x \in \mathcal{K}} \Psi(x)$, and given a response $\tilde{g}^t \doteq \tilde{g}(x^t)$ to the gradient query at point x^t it will compute its next query point as

$$x^{t+1} = \arg\min_{y \in \mathcal{K}} \{\alpha \langle \tilde{g}^t, y - x^t \rangle + V_{x^t}(y)\}, \tag{14}$$

which corresponds to a proximal problem. The output of the method is the average of iterates $\bar{x}^T \doteq \frac{1}{T}\sum_{t=1}^T x^t$.

[4] We have normalized the function so that the constant of r-uniform convexity is 1.

Theorem 4.5. *Let $F \in \mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$ and $\Psi : \mathcal{K} \to \mathbb{R}$ be an r-uniformly convex function. Then the inexact mirror-descent method with Ψ proximal setup, step size $\alpha = \frac{1}{L_0}[rD_\Psi(\mathcal{K})/T]^{1-1/r}$, and an η-approximate gradient for F over $\mathcal{K}$, guarantees after T steps an accuracy*

$$F(\bar{x}^T) - F^* \leq L_0 \left(\frac{rD_\Psi(\mathcal{K})}{T}\right)^{1/r} + \eta.$$

Proof. We first state without proof the following identity for prox-functions (for example, see (5.3.20) in [11]): for all x, x' and u in $\mathcal{K}$

$$V_x(u) - V_{x'}(u) - V_x(x') = \langle \nabla V_x(x'), u - x' \rangle.$$

On the other hand, the optimality conditions of problem (14) are

$$\langle \alpha \tilde{g}^t + \nabla V_{x^t}(x^{t+1}), u - x^{t+1} \rangle \geq 0, \quad \forall u \in \mathcal{K}.$$

Let $u \in \mathcal{K}$ be an arbitrary vector, and let s be such that $1/r + 1/s = 1$. Since $\tilde{g}^t$ is a η-approximate gradient,

$$\begin{array}{rcl}
\alpha[F(x^t) - F(u)] & \leq & \alpha\langle \nabla F(x^t), x^t - u\rangle \\
& \leq & \alpha\langle \tilde{g}^t, x^t - u\rangle + \alpha\eta \\
& = & \alpha\langle \tilde{g}^t, x^t - x^{t+1}\rangle + \alpha\langle \tilde{g}^t, x^{t+1} - u\rangle + \alpha\eta \\
& \leq & \alpha\langle \tilde{g}^t, x^t - x^{t+1}\rangle - \langle \nabla V_{x^t}(x^{t+1}), x^{t+1} - u\rangle + \alpha\eta \\
& = & \alpha\langle \tilde{g}^t, x^t - x^{t+1}\rangle + V_{x^t}(u) - V_{x^{t+1}}(u) - V_{x^t}(x^{t+1}) + \alpha\eta \\
& \leq & [\alpha\langle \tilde{g}^t, x^t - x^{t+1}\rangle - \dfrac{1}{r}\|x^t - x^{t+1}\|^r] + V_{x^t}(u) - V_{x^{t+1}}(u) + \alpha\eta \\
& \leq & \dfrac{1}{s}\|\alpha\tilde{g}^t\|_*^s + V_{x^t}(u) - V_{x^{t+1}}(u) + \alpha\eta,
\end{array}$$

where we have used all the observations above, and the last step holds by Fenchel's inequality.

Let us choose u such that $F(u) = F^*$, thus by definition of $\bar{x}^T$ and by convexity of f

$$\alpha T[F(\bar{x}^T) - F^*] \;\leq\; \sum_{t=1}^{T} \alpha[F(x^t) - F^*] \;\leq\; \frac{(\alpha L_0)^s}{s} T + D_\Psi(\mathcal{K}) + \alpha T\eta.$$

and since $\alpha = \frac{1}{L_0}\left(\frac{rD_\Psi(\mathcal{K})}{T}\right)^{1/s}$ we obtain $F(\bar{x}^T) - F^* \leq L_0 \left(\frac{rD_\Psi(\mathcal{K})}{T}\right)^{1/r} + \eta$. □

We can readily apply the result above to stochastic convex programs in non-smooth ℓ_p settings.

Definition 4.6 (ℓ_p-setup)**.** *Let $1 \leq p \leq \infty$, $L_0, R > 0$, and $\mathcal{K} \subseteq \mathcal{B}_p^d(R)$ be a convex body. We define as the (non-smooth) ℓ_p-setup the family of problems $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E}_{\mathbf{w}}[f(x, \mathbf{w})]\}$, where for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{K}, L_0)$.*

In the smooth ℓ_p-setup we additionally assume that $F \in \mathcal{F}^1_{\|\cdot\|_p}(\mathcal{K}, L_1)$.

From constructions of r-uniformly convex functions for ℓ_p spaces, with $r = \min\{2, p\}$ (see Appendix A), we know that there exists an efficiently computable Prox function Ψ (*i.e.* whose value and gradient can be computed exactly, and thus problem (14) is solvable for simple enough $\mathcal{K}$). The consequences in terms of estimation complexity are summarized in the following corollary, and proved in Appendix C.

Corollary 4.7. *The stochastic optimization problem in the non-smooth ℓ_p-setup can be solved with accuracy ε by:*

- *If $p = 1$, using $O\left(d \log d \cdot \left(\frac{L_0 R}{\varepsilon}\right)^2\right)$ queries to STAT $\left(\frac{\varepsilon}{4L_0 R}\right)$;*
- *If $1 < p < 2$, using $O\left(d \log d \cdot \frac{1}{(p-1)} \left(\frac{L_0 R}{\varepsilon}\right)^2\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{[\log d] L_0 R}\right)\right)$;*
- *If $p = 2$, using $O\left(d \cdot \left(\frac{L_0 R}{\varepsilon}\right)^2\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{L_0 R}\right)\right)$;*
- *If $2 < p < \infty$, using $O\left(d \log d \cdot 4^p \left(\frac{L_0 R}{\varepsilon}\right)^p\right)$ queries to VSTAT $\left(\left(\frac{64 L_0 R \log d}{\varepsilon}\right)^p\right)$.*

4.2.2 Smooth Case: Nesterov Accelerated Method

Now we focus on the class of functions whose expectation has Lipschitz continuous gradient. For simplicity, we will restrict the analysis to the case where the Prox function is obtained from a strongly convex function, i.e., r-uniform convexity with $r = 2$. We utilize a known inexact variant of Nesterov's accelerated method [69].

Theorem 4.8 ([23])**.** *Let $F \in \mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$, and let $\Psi : \mathcal{K} \to \mathbb{R}_+$ be a 1-strongly convex function w.r.t. $\|\cdot\|$. Let (x^t, y^t, z^t) be the iterates of the accelerated method with Ψ proximal setup, and where the algorithm has access to an η-approximate gradient oracle for F over $\mathcal{K}$. Then,*

$$F(y^T) - F^* \leq \frac{L_1 D_\Psi(\mathcal{K})}{T^2} + 3\eta.$$

The consequences for the smooth ℓ_p-setup, which are straightforward from the theorem above and Observation 4.2, are summarized below, and proved in Appendix D.

Corollary 4.9. *Any stochastic convex optimization problem in the smooth ℓ_p-setup can be solved with accuracy ε by:*

- *If $p = 1$, using $O\left(d\sqrt{\log d} \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\frac{\varepsilon}{12 L_0 R}\right)$;*
- *If $1 < p < 2$, using $O\left(d \log d \cdot \frac{1}{\sqrt{p-1}} \sqrt{\frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{[\log d] L_0 R}\right)\right)$;*
- *If $p = 2$, using $O\left(d \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{L_0 R}\right)\right)$.*

4.2.3 Strongly Convex Case

Finally, we consider the class $\mathcal{S}_{\|\cdot\|}(\mathcal{K}, \kappa)$ of strongly convex functions. We further restrict our attention to the Euclidean case, i.e., $\|\cdot\| = \|\cdot\|_2$. There are two main advantages of having a strongly convex objective: On the one hand, gradient methods in this case achieve linear convergence rate, on the other hand we will see that estimation complexity is independent of the radius. Let us first make precise the first statement: It turns out that with a (η, M, μ)-oracle we can implement the inexact dual gradient method [24] achieving linear convergence rate. The result is as follows

Theorem 4.10 ([24]). *Let $F : \mathcal{K} \to \mathbb{R}$ be a subdifferentiable convex function endowed with a (η, M, μ)-oracle over $\mathcal{K}$. Let y^t be the sequence of averages of the inexact dual gradient method, then*

$$F(y^T) - F^* \leq \frac{MR^2}{2} \exp\left(-\frac{\mu}{M}(T+1)\right) + \eta.$$

The results in [24] indicate that the accelerated method can also be applied in this situation, and it does not suffer from noise accumulation. However, the accuracy requirement is more restrictive than for the primal and dual gradient methods. In fact, the required accuracy for the approximate gradient is $\eta = O(\varepsilon\sqrt{\mu/M})$; although this is still independent of the radius, it makes estimation complexity much more sensitive to condition number, which is undesirable.

An important observation of the dual gradient algorithm is that it does not require function values (as opposed to its primal version). This together with Lemma 4.4.

Corollary 4.11. *The stochastic convex optimization problem $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E_w}[f(x,w)]\}$, where $F \in \mathcal{S}_{\|\cdot\|_2}(\mathcal{K},\kappa) \cap \mathcal{F}^1_{\|\cdot\|_2}(\mathcal{K}, L_1)$, and for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}^0_{\|\cdot\|_2}(\mathcal{K}, L_0)$, can be solved to accuracy $\varepsilon > 0$ using $O\left(d \cdot \frac{L_1}{\kappa} \log\left(\frac{L_1 R}{\varepsilon}\right)\right)$ queries to STAT$(\Omega(\sqrt{\varepsilon\kappa}/L_0))$.*

Without the assumption $F \in \mathcal{F}^1_{\|\cdot\|_2}(\mathcal{K}, L_1)$ the problem can be solved to accuracy $\varepsilon > 0$ by using $O\left(d \cdot \frac{L_0^2}{\varepsilon\kappa} \log\left(\frac{L_0 R}{\varepsilon}\right)\right)$ queries to STAT$(\Omega(\sqrt{\varepsilon\kappa}/L_0))$.

4.3 Applications to Generalized Linear Regression

We conclude this section with a comparison of the bounds obtained by statistical query inexact first-order methods with some state-of-the-art error bounds for linear regression problems. To be precise, we compare sample complexity of obtaining excess error ε (with constant success probability or in expectation) with the estimation complexity of the SQ oracle for achieving ε accuracy. It is worth noticing though that these two quantities are not directly comparable, as an SQ algorithm performs a (polynomial) number of queries to the oracle. However, this comparison shows that our results roughly match what can be achieved via samples.

We consider the *generalized linear regression* problem: Given a normed space $(\mathbb{R}^d, \|\cdot\|)$, let $\mathcal{W} \subseteq \mathbb{R}^d$ be the input space, and $\mathbb{R}$ be the output space. Let $(\mathbf{w}, \mathbf{z}) \sim D$, where D is an unknown target distribution supported on $\mathcal{W} \times \mathbb{R}$. The objective is to obtain a linear predictor $x \in \mathcal{K}$ that predicts the outputs as a function of the inputs coming from D. Typically, $\mathcal{K}$ is prescribed by desirable structural properties of the predictor, *e.g.* sparsity or low norm. The parameters determining complexity are given by bounds on the predictor and input space: $\mathcal{K} \subseteq \mathcal{B}_{\|\cdot\|}(R)$ and $\mathcal{W} \subseteq \mathcal{B}_{\|\cdot\|_*}(W)$. Under these assumptions we may restrict the output space to $[-M, M]$, where $M = RW$.

The prediction error is measured using a *loss function*. For a function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, letting $f(x,(w,z)) = \ell(\langle w, x\rangle, z)$, we seek to solve the stochastic convex program $\min_{x\in\mathcal{K}}\{F(x) = \mathbf{E}_{(\mathbf{w},\mathbf{z})\sim D}[f(x,(\mathbf{w},\mathbf{z}))]\}$. We assume that $\ell(\cdot, z)$ is convex for every z in the support of D. A common example of this problem is the (random design) least squares linear regression, where $\ell(z', z) = (z' - z)^2$.

Non-smooth case: We assume that for every z in the support of D, $\ell(\cdot, z) \in \mathcal{F}^0_{|\cdot|}([-M, M], L_{\ell,0})$. To make the discussion concrete, let us consider the ℓ_p-setup, *i.e.* $\|\cdot\| = \|\cdot\|_p$. Hence the Lipschitz constant of our stochastic objective $f(\cdot,(w,z)) = \ell(\langle w, \cdot\rangle, z)$ can be upper bounded as $L_0 \leq L_{\ell,0} \cdot W$. For this setting Kakade et al. [50] show that the sample complexity of achieving excess error $\varepsilon > 0$ with constant success probability is $n = O\left(\left(\frac{L_{\ell,0}WR}{\varepsilon}\right)^2 \ln d\right)$ when $p = 1$; and $n = O\left(\left(\frac{L_{\ell,0}WR}{\varepsilon}\right)^2 (q-1)\right)$ for

$1 < p \leq 2$. Using Corollary 4.7 we obtain that the estimation complexity of solving this problem using our SQ implementation of the mirror-descent method gives the same up to (at most) a logarithmic in d factor.

Kakade et al. [50] do not provide sample complexity bounds for $p > 2$, however since their approach is based on Rademacher complexity (see Appendix B for the precise bounds), the bounds in this case should be similar to ours as well.

Strongly convex case: Let us now consider a generalized linear regression with regularization. Here

$$f(x,(w,z)) = \ell(\langle w,x\rangle, z) + \lambda \cdot \Phi(x),$$

where $\Phi : \mathcal{K} \to \mathbb{R}$ is a 1-strongly convex function and $\lambda > 0$. This model has a variety of applications in machine learning, such as ridge regression and soft-margin SVM. For the non-smooth linear regression in ℓ_2 setup (as described above), Shalev-Shwartz et al. [80] provide a sample complexity bound of $O\left(\frac{(L_{\ell,0}W)^2}{\lambda\varepsilon}\right)$ (with constant success probability). Note that the expected objective is 2λ-strongly convex and therefore, applying Corollary 4.11, we get the same (up to constant factors) bounds on estimation complexity of solving this problem by SQ algorithms.

5 Optimization of Bounded-Range Functions

The estimation complexity bounds obtained for gradient descent-based methods depend polynomially either on the the Lipschitz constant L_0 and the radius R of $\mathcal{K}$ (unless the functions are strongly convex). In some cases such bounds are not explicitly available (or too large) and instead we know that the range of functions in the support of the distribution is bounded, that is, $\max_{(x,y\in\mathcal{K},\ v,w\in\mathcal{W})}(f(x,v) - f(y,w)) \leq 2B$ for some B. Without loss of generality we may assume that for all $w \in \mathcal{W}$, $f(\cdot,w) \in \mathcal{F}(\mathcal{K},B)$.

5.1 Random walks

We first show that a simple extension of the random walk approach of Kalai and Vempala [51] and Lovász and Vempala [62] can be used to address this setting. One advantage of this approach is that to optimize F it requires only access to approximate values of F (such an oracle is also referred to as approximate zero-order oracle). Namely, a τ-approximate value oracle for a function F is the oracle that for every x in the domain of F, returns value v such that $|v - F(x)| \leq \tau$.

We note that the random walk based approach was also (independently[5]) used in a recent work of Belloni et al. [9]. Their work includes an optimized and detailed analysis of this approach and hence we only give a brief outline of the proof here.

Theorem 5.1. *There is an algorithm that with probability at least $2/3$, given any convex program $\min_{x\in\mathcal{K}} F(x)$ in $\mathbb{R}^d$ where $\forall x \in \mathcal{K}$, $|F(x)| \leq 1$ and $\mathcal{K}$ is given by a membership oracle with the guarantee that $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$, outputs an ε-optimal solution in time poly$(d, \frac{1}{\varepsilon}, \log(R_1/R_0))$ using poly$(d, \frac{1}{\varepsilon})$ queries to (ε/d)-approximate value oracle.*

Proof. Let $x^* = \mathrm{argmin}_{x\in\mathcal{K}} F(x)$ and $F^* = F(x^*)$. The basic idea is to sample from a distribution that has most of its measure on points with $F(x) \leq F^* + \varepsilon$. To do this, we use the random walk approach as in [51, 62] with a minor extension. The algorithm performs a random walk whose stationary distribution is proportional to $g_\alpha(x) = e^{-\alpha F(x)}$, with $g(x) = e^{-F(x)}$. Each step of the walk is a function evaluation. Noting that $e^{-\alpha F(x)}$ is a logconcave function, the number of steps is poly$(d, \log\alpha, \beta)$ to get a point from a distribution within total variation distance β of the target distribution. Applying Lemma 5.1 from [62]

[5] The statement of our result and proof sketch were included by the authors for completeness in the appendix of [37, v2].

(which is based on Lemma 5.16 from [64]) with $B = 2$ to g_α with $\alpha = 4(d + \ln(1/\delta)/\varepsilon$, we have (note that α corresponds to $a_m = \frac{1}{B}(1 + 1/\sqrt{n})^m$ in that statement).

$$\mathbf{Pr}[g(\mathbf{x}) < e^{-\varepsilon} \cdot g(x^*)] \leq \delta \left(\frac{2}{e}\right)^{d-1}. \tag{15}$$

Therefore, the probability that a random point $\mathbf{x}$ sampled proportionately to $g_\alpha(x)$ does not satisfy $F(\mathbf{x}) < F^* + \varepsilon$ is at most $\delta(2/e)^{d-1}$.

Now we turn to the extension, which arises because we can only evaluate $F(x)$ approximately through the oracle. We assume w.l.o.g. that the value oracle is consistent in its answers (i.e., returns the same value on the same point). The value returned by the oracle $\tilde{F}(x)$ satisfies $|F(x) - \tilde{F}(x)| \leq \varepsilon/d$. The stationary distribution is now proportional to $\tilde{g}_\alpha(x) = e^{-\alpha\tilde{F}(x)}$ and satisfies

$$\frac{\tilde{g}_\alpha(x)}{g_\alpha(x)} = e^{-\alpha(\tilde{F}(x) - F(x))} \leq e^{\alpha\frac{\varepsilon}{d}} \leq e^5. \tag{16}$$

We now argue that with large probability, the random walk with the approximate evaluation oracle will visit a point x where F has of value at most $F^* + \varepsilon$. Assuming that a random walk gives samples from a distribution (sufficiently close to being) proportional to $\tilde{g}_\alpha$, from property (16), the probability of the set $\{x : g(x) > e^{-\varepsilon} \cdot g(x^*)\}$ is at most a factor of e^{10} higher than for the distribution proportional to g_α (given in eq. (15)). Therefore with a small increase in the number of steps a random point from the walk will visit the set where F has value of at most $F^* + \varepsilon$ with high probability. Thus the minimum function value that can be achieved is at most $F^* + \varepsilon + 2\varepsilon/d$.

Finally, we need the random walk to mix rapidly for the extension. Note that $\tilde{F}(x)$ is approximately convex, *i.e.* for any $x, y \in \mathcal{K}$ and any $\lambda \in [0, 1]$, we have

$$\tilde{F}(\lambda x + (1 - \lambda)y) \leq \lambda\tilde{F}(x) + (1 - \lambda)\tilde{F}(y) + 2\varepsilon/d. \tag{17}$$

and therefore $\tilde{g}_\alpha$ is a near-logconcave function that satisfies, for any $x, y \in \mathcal{K}$ and $\lambda \in [0, 1]$,

$$\tilde{g}_\alpha(\lambda x + (1 - \lambda)y) \geq e^{-2\alpha\varepsilon/d} \cdot \tilde{g}_\alpha(x)^\lambda \tilde{g}_\alpha(x)^{1-\lambda} \geq e^{-10} \cdot \tilde{g}_\alpha(x)^\lambda \tilde{g}_\alpha(x)^{1-\lambda}.$$

As a result, as shown by Applegate and Kannan [2], it admits an isoperimetric inequality that is weaker than that for logconcave functions by a factor of e^{10}. For the grid walk, as analyzed by them, this increases the convergence time by a factor of at most e^{20}. The grid walk's convergence also depends (logarithmically) on the Lipshitz constant of $\tilde{g}_\alpha$. This dependence is avoided by the ball walk, whose convergence is again based on the isoperimetric inequality, as well as on local properties, namely on the 1-step distribution of the walk. It can be verified that the analysis of the ball walk (e.g., as in [64]) can be adapted to near-logconcave functions with an additional factor of $O(1)$ in the mixing time. □

Going back to the stochastic setting, let $F(x) = \mathbf{E}_D[f(x, \mathbf{w})]$. If $\forall w$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$ then a single query $f(x, w)$ to STAT(τ/B) is equivalent to a query to a τ-approximate value oracle for $F(x)$.

Corollary 5.1. *There is an algorithm that for any distribution D over $\mathcal{W}$ and convex program $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E}_{\mathbf{w}\sim D}[f(x, \mathbf{w})]\}$ in $\mathbb{R}^d$ where $\forall w$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$ and $\mathcal{K}$ is given by a membership oracle with the guarantee that $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$, with probability at least $2/3$, outputs an ε-optimal solution in time poly($d, \frac{B}{\varepsilon}, \log(R_1/R_0)$) using poly($d, \frac{B}{\varepsilon}$) queries to STAT($\varepsilon/(dB)$).*

We point out that τ-approximate value oracle is strictly weaker than STAT(τ). This follows from a simple result of Nemirovsky and Yudin [68, p.360] who show that linear optimization over $\mathcal{B}_2^d$ with τ-approximate value oracle requires $\tau = \Omega(\sqrt{\log q} \cdot \varepsilon/d)$ for any algorithm using q queries. Together with our upper bounds in Section 3 this implies that approximate value oracle is weaker than STAT.

5.2 Center-of-Gravity

An alternative and simpler technique to establish the $O(d^2B^2/\varepsilon^2)$ upper bound on the estimation complexity for B-bounded-range functions is to use cutting-plane methods, more specifically, the classic center-of-gravity method, originally proposed by Levin [58].

We introduce some notation. Given a convex body $\mathcal{K}$, let $\mathbf{x}$ be a uniformly and randomly chosen point from $\mathcal{K}$. Let $z(\mathcal{K}) \doteq \mathbf{E}[\mathbf{x}]$ and $A(\mathcal{K}) \doteq \mathbf{E}[(\mathbf{x} - z(\mathcal{K}))(\mathbf{x} - z(\mathcal{K}))^T]$ be the center of gravity and co-variance matrix of $\mathcal{K}$ respectively. We define the (origin-centered) inertial ellipsoid of $\mathcal{K}$ as $\mathcal{E}_{\mathcal{K}} \doteq \{y : y^T A(\mathcal{K})^{-1} y \leq 1\}$.

The classic center-of-gravity method starts with $G^0 \doteq \mathcal{K}$ and iteratively computes a progressively smaller body containing the optimum of the convex program. We call such a body a *localizer*. Given a localizer G^{t-1}, for $t \geq 1$, the algorithm computes $x^t = z(G^{t-1})$ and defines the new localizer to be

$$G^t \doteq G^{t-1} \cap \{y \in \mathbb{R}^d \mid \langle \nabla F(x^t), y - x^t \rangle \leq 0\}.$$

It is known that that any halfspace containing the center of gravity of a convex body contains at least $1/e$ of its volume [44], that is $\mathrm{vol}(G^t) \leq \gamma \cdot \mathrm{vol}(G^{t-1})$, where $\gamma = 1 - 1/e$. We call this property the *volumetric guarantee* with parameter γ.

The first and well-known issue we will deal with is that the exact center of gravity of G^{t-1} is hard to compute. Instead, following the approach in [12], we will let x^t be an approximate center-of-gravity. For such an approximate center we will have a volumetric guarantee with somewhat larger parameter γ.

The more significant issue is that we do not have access to the exact value of $\nabla F(x^t)$. Instead will show how to compute an approximate gradient $\tilde{g}(x^t)$ satisfying for all $y \in G^t$,

$$|\langle \tilde{g}(x^t) - \nabla F(x^t), y - x^t \rangle| \leq \eta. \tag{18}$$

Notice that this is a weaker condition than the one required by (9): first, we only impose the approximation on the localizer; second, the gradient approximation is only at x^t. These two features are crucial for our results.

Condition (18) implies that for all $y \in G^{t-1} \setminus G^t$,

$$F(y) \geq F(x^t) + \langle \nabla F(x^t), y - x^t \rangle \geq F(x^t) + \langle \tilde{g}(x^t), y - x^t \rangle - \eta > F(x^t) - \eta.$$

Therefore we will lose at most η by discarding points in $G^{t-1} \setminus G^t$.

Plugging this observation into the standard analysis of the center-of-gravity method (see, *e.g.* [67, Chapter 2]) yields the following result.

Theorem 5.2. *For $B > 0$, let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex body, and $F \in \mathcal{F}(\mathcal{K}, B)$. Let $x^1, x^2, \ldots$ and $\tilde{g}(x^1), \tilde{g}(x^2), \ldots$ be a sequence of points and gradient estimates such that for $G_0 \doteq \mathcal{K}$ and $G^t \doteq G^{t-1} \cap \{y \in \mathbb{R}^d \mid \langle \tilde{g}(x^t), y - x^t \rangle \leq 0\}$ for all $t \geq 1$, we have a volumetric guarantee with parameter $\gamma < 1$ and condition* (18) *is satisfied for some fixed $\eta > 0$. Let $\hat{x}^T \doteq \mathrm{argmin}_{t \in [T]} F(x^t)$, then*

$$F(\hat{x}^T) - \min_{x \in \mathcal{K}} F(x) \leq \gamma^{T/d} \cdot 2B + \eta\ .$$

In particular, choosing $\eta = \varepsilon/2$, and $T = \lceil d \log(\frac{1}{\gamma}) \log(\frac{4B}{\varepsilon}) \rceil$ gives $F(\hat{x}^T) - \min_{x \in \mathcal{K}} F(x) \leq \varepsilon$.

We now describe how to compute an approximate gradient satisfying condition (18). We show that it suffices to find an ellipsoid $\mathcal{E}$ centered at x^t such that $x^t + \mathcal{E}$ is included in G^t and G^t is included in $x^t + R \cdot \mathcal{E}$. The first condition, together with the bound on the range of functions in the support of the distribution, implies a bound on the ellipsoidal norm of the gradients. This allows us to use Theorem 3.9 to estimate $\nabla F(x^t)$ in the ellipsoidal norm. The second condition can be used to translate the error in the ellipsoidal norm to the error η over G^t as required by condition (18). Formally we prove the following lemma:

Lemma 5.3. *Let $G \subseteq \mathbb{R}^d$ be a convex body, $x \in G$, and $\mathcal{E} \subseteq \mathbb{R}^d$ be an origin-centered ellipsoid that satisfies*

$$R_0 \cdot \mathcal{E} \subseteq (G - x) \subseteq R_1 \cdot \mathcal{E}.$$

Given $F(x) = \mathbf{E}_{\mathbf{w}}[f(x, \mathbf{w})]$ a convex function on G such that for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$, we can compute a vector $\tilde{g}(x)$ satisfying (18) *in polynomial time using $2d$ queries to STAT $\left(\Omega\left(\frac{\eta}{[R_1/R_0]B}\right)\right)$.*

Proof. Let us first bound the norm of the gradients, using the norm induced by the ellipsoid $\mathcal{E}$.

$$\begin{aligned} \|\nabla f(x,w)\|_{\mathcal{E}} &= \sup_{y \in \mathcal{E}} \langle \nabla f(x,w), y \rangle \leq \frac{1}{R_0} \sup_{y \in G} \langle \nabla f(x,w), y - x \rangle \\ &\leq \frac{1}{R_0} \sup_{y \in G} [f(y,w) - f(x,w)] \leq \frac{2B}{R_0}. \end{aligned}$$

Next we observe that for any vector $\tilde{g}$,

$$\begin{aligned} \sup_{y \in G} \langle \nabla F(x) - \tilde{g}, y - x \rangle &= R_1 \sup_{y \in G} \left\langle \nabla F(x) - \tilde{g}, \frac{y - x}{R_1} \right\rangle \leq R_1 \sup_{y \in \mathcal{E}} \langle \nabla F(x) - \tilde{g}, y \rangle \\ &= R_1 \|\nabla F(x) - \tilde{g}\|_{\mathcal{E}}. \end{aligned}$$

From this we reduce obtaining $\tilde{g}(x)$ satisfying (18) to a mean estimation problem in an ellipsoidal norm with error $R_0\eta/[2R_1B]$, which by Theorem 3.9 (with Lemma 3.2) can be done using $2d$ queries to STAT $\left(\Omega\left(\frac{\eta}{[R_1/R_0]B}\right)\right)$. □

It is known that if $x^t = z(G^t)$ then the inertial ellipsoid of G^t has the desired property with the ratio of the radii being d.

Theorem 5.4. *[52] For any convex body $G \subseteq \mathbb{R}^d$, $\mathcal{E}_G$ (the inertial ellipsoid of G) satisfies*

$$\sqrt{\frac{d+2}{d}} \cdot \mathcal{E}_G \subseteq (G - z(G)) \subseteq \sqrt{d(d+2)} \cdot \mathcal{E}_G.$$

This means that estimates of the gradients sufficient for executing the exact center-of-gravity method can be obtained using SQs with estimation complexity of $O(d^2B^2/\varepsilon^2)$.

Finally, before we can apply Theorem 5.2, we note that instead of $\hat{x}^T \doteq \operatorname{argmin}_{t \in [T]} F(x^t)$ we can compute $\tilde{x}^T = \operatorname{argmin}_{t \in [T]} \tilde{F}(x^t)$ such that $F(\tilde{x}^T) \leq F(\hat{x}^T) + \varepsilon/2$. This can be done by using T queries to STAT$(\varepsilon/[4B])$ to obtain $\tilde{F}(x^t)$ such that $|\tilde{F}(x^t) - F(x^t)| \leq \varepsilon/4$ for all $t \in [T]$. Plugging this into Theorem 5.2 we get the following (inefficient) SQ version of the center-of-gravity method.

Theorem 5.5. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex body, and assume that for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$. Then there is an algorithm that for every distribution D over $\mathcal{W}$ finds an ε-optimal solution for the stochastic convex optimization problem $\min_{x \in \mathcal{K}} \{\mathbf{E}_{\mathbf{w} \sim D}[f(x, \mathbf{w})]\}$ using $O(d^2 \log(B/\varepsilon))$ queries to STAT$(\Omega(\varepsilon/[Bd]))$.*

5.2.1 Computational Efficiency

The algorithm described in Theorem 5.5 relies on the computation of the exact center of gravity and inertial ellipsoid for each localizer. Such computation is #P-hard in general. We now describe a computationally efficient version of the center-of-gravity method that is based on computation of approximate center of gravity and inertial ellipsoid via random walks, an approach was first proposed by Bertsimas and Vempala [12].

We first observe describe the volumetric guarantee that is satisfied by any cut through an approximate center of gravity.

Lemma 5.6. *[12] For a convex body $G \subseteq \mathbb{R}^d$, let z be any point s.t. $\|z - z(G)\|_{\mathcal{E}_G} = t$. Then, for any halfspace H containing z,*

$$\text{Vol}(G \cap H) \geq \left(\frac{1}{e} - t\right) \text{Vol}(G).$$

From this result, we know that it suffices to approximate the center of gravity in the inertial ellipsoid norm in order to obtain the volumetric guarantee.

Lovász and Vempala [63] show that for any convex body G given by a membership oracle, a point $x \in G$ and R_0, R_1 s.t. $R_0 \cdot \mathcal{B}_2^d \subseteq (G - x) \subseteq R_1 \cdot \mathcal{B}_2^d$, there is a sampling algorithm based on a random walk that outputs points that are within statistical distance α of the uniform distribution in time polynomial in $d, \log(1/\alpha), \log(R_1/R_0)$. The current best dependence on d is d^4 for the first random point and d^3 for all subsequent points [61]. Samples from such a random walk can be directly used to estimate the center of gravity and the inertial ellipsoid of G.

Theorem 5.7. *[63] There is a randomized algorithm that for any $\varepsilon > 0, 1 > \delta > 0$, for a convex body G given by a membership oracle and a point x s.t. $R_0 \cdot \mathcal{B}_2^d \subseteq (G - x) \subseteq R_1 \cdot \mathcal{B}_2^d$, finds a point z and an origin-centered ellipsoid $\mathcal{E}$ s.t. with probability at least $1 - \delta$, $\|z - z(G)\|_{\mathcal{E}_G} \leq \varepsilon$ and $\mathcal{E} \subset \mathcal{E}_G \subset (1 + \varepsilon)\mathcal{E}$. The algorithm uses $\tilde{O}(d^4 \log(R_1/R_0) \log(1/\delta)/\varepsilon^2)$ calls to the membership oracle.*

We now show that an algorithm having the guarantees given in Theorem 5.5 can be implemented in time poly$(d, B/\varepsilon, \log(R_1/R_0))$. More formally,

Theorem 5.8. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex body given by a membership oracle and a point x s.t. $R_0 \cdot \mathcal{B}_2^d \subseteq (G - x) \subseteq R_1 \cdot \mathcal{B}_2^d$, and assume that for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$. Then there is an algorithm that for every distribution D over $\mathcal{W}$ finds an ε-optimal solution for the stochastic convex optimization problem $\min_{x \in \mathcal{K}}\{\mathbf{E}_{\mathbf{w} \sim D}[f(x, \mathbf{w})]\}$ using $O(d^2 \log(B/\varepsilon))$ queries to STAT$(\Omega(\varepsilon/[Bd]))$. The algorithm succeeds with probability $\geq 2/3$ and runs in poly$(d, B/\varepsilon, \log(R_1/R_0))$ time.*

Proof. Let the initial localizer be $G = \mathcal{K}$. We will prove the following by induction: For every step of the method, if G is the current localizer then a membership oracle for G can be implemented efficiently given a membership oracle for $\mathcal{K}$ and we can efficiently compute $x \in G$ such that, with probability at least $1 - \delta$,

$$R_0' \cdot \mathcal{B}_2^d \subseteq G - x \subseteq R_1' \cdot \mathcal{B}_2^d, \tag{19}$$

where $R_1'/R_0' \leq \max\{R_1/R_0, 4d\}$. We first note that the basis of the induction holds by the assumptions of the theorem. We next show that the assumption of the induction allows us to compute the desired approximations to the center of gravity and the inertial ellipsoid which in turn will allow us to prove the inductive step.

Since G satisfies the assumptions of Theorem 5.7, we can obtain in polynomial time (with probability $1 - \delta$) an approximate center z and ellipsoid $\mathcal{E}$ satisfying $\|z - z(G)\|_{\mathcal{E}_G} \leq \chi$ and $\mathcal{E} \subseteq \mathcal{E}_G \subseteq (1 + \chi)\mathcal{E}$, where $\chi \doteq 1/e - 1/3$. By Lemma 5.6 and $\|z - z(G)\|_{\mathcal{E}_G} \leq \chi$, we get that volumetric guarantee holds for the next localizer G' with parameter $\gamma = 2/3$.

Let us now observe that

$$(\sqrt{(d+2)/d} - \chi) \cdot \mathcal{E} + z \subseteq \sqrt{(d+2)/d} \cdot \mathcal{E}_G + z(G) \subseteq G.$$

We only prove the first inclusion, as the second one holds by Theorem 5.4. Let $y \in \alpha\mathcal{E} + z$ (where $\alpha = \sqrt{(d+2)/d - \chi)}$). Now we have $\|y - z(G)\|_{\mathcal{E}_G} \leq \|y - z\|_{\mathcal{E}_G} + \|z - z(G)\|_{\mathcal{E}_G} \leq \|y - z\|_{\mathcal{E}} + \chi \leq \alpha + \chi = \sqrt{(d+2)/d}$. Similarly, we can prove that

$$G - z \subseteq \sqrt{d(d+2)} \cdot \mathcal{E}_G + (z(G) - z) \subseteq (\sqrt{d(d+2)} + \chi) \cdot \mathcal{E}_G \subseteq (1 + \chi)(\sqrt{d(d+2)} + \chi) \cdot \mathcal{E}.$$

Denoting $r_0 \doteq \sqrt{(d+2)/d} - \chi$ and $r_1 \doteq (1+\chi)(\sqrt{d(d+2)} + \chi)$ we obtain that $r_0 \cdot \mathcal{E} \subseteq G - z \subseteq r_1 \cdot \mathcal{E}$, where $\frac{r_1}{r_0} = \frac{(1+\chi)(\sqrt{d(d+2)}+\chi)}{\sqrt{(d+2)/d}-\chi} \leq \frac{3}{2}d$. By Lemma 5.3 this implies that using $2d$ queries to STAT$(\Omega(\varepsilon/[Bd]))$ we can obtain an estimate $\tilde{g}$ of $\nabla F(z)$ that suffices for executing the approximate center-of-gravity method.

We finish the proof by establishing the inductive step. Let the new localizer G' be defined as G after removing the cut through z given by $\tilde{g}$ and transformed by the affine transformation induced by z and $\mathcal{E}$ (that is mapping z to the origin and $\mathcal{E}$ to $\mathcal{B}_2^d$). Notice that after the transformation $r_0 \cdot \mathcal{B}_2^d \subseteq \tilde{G} \subseteq r_1 \cdot \mathcal{B}_2^d$, where $\tilde{G}$ denotes G after the affine transformation. G' is obtained from $\tilde{G}$ by a cut though the origin. This implies that G' contains a ball of radius $r_0/2$ which is inscribed in the half of $r_0 \cdot \mathcal{B}_2^d$ that is contained in G'. Let x' denote the center of this contained ball (which can be easily computed from $\tilde{g}$, z and $\mathcal{E}$). It is also easy to see that a ball of radius $r_0/2 + r_1$ centered at x' contains G'. Hence $G' - x'$ is sandwiched by two Euclidean balls with the ratio of radii being $(r_1 + r_0/2)/(r_0/2) \leq 4d$. Also notice that since a membership oracle for $\mathcal{K}$ is given and the number of iterations of this method is $O(d \log(4B/\varepsilon))$ then a membership oracle for G' can be efficiently computed.

Finally, choosing the confidence parameter δ inversely proportional to the number of iterations of the method guarantees a constant success probability. □

6 Applications

In this section we describe several applications of our results. We start by giving SQ implementation of algorithms for learning halfspaces that eliminate the linear dependence on the dimension in previous work. Then we obtain algorithms for high-dimensional mean estimation with local differential privacy that re-derive and generalize existing bounds. We also give the first algorithm for solving general stochastic convex programs with local differential privacy. Another immediate corollary of our results is a strengthening and generalization of algorithms for answering sequences of convex minimization queries differentially privately given in [89]. Finally, we show that our algorithms together with lower bounds for SQ algorithms give lower bounds against convex programs.

Additional applications in settings where SQ algorithms are used can be derived easily. For example, our results immediately imply that an algorithm for answering a sequence of adaptively chosen SQs (such as those given in [32, 31, 8] can be used to solve a sequence of adaptively chosen stochastic convex minimization problems. This question that has been recently studied by Bassily et al. [8] and our bounds can be easily seen to strengthen and generalize some of their results (see Sec. 6.3 for an analogous comparison).

6.1 Learning Halfspaces

We now use our high-dimensional mean estimation algorithms to address the efficiency of SQ versions of online algorithms for learning halfspaces (also known as linear threshold functions). A linear threshold function is a Boolean function over $\mathbb{R}^d$ described by a weight vector $w \in \mathbb{R}^d$ together with a threshold $\theta \in \mathbb{R}$ and defined as $f_{w,\theta}(x) \doteq \mathsf{sign}(\langle w, x\rangle - \theta)$.

Margin Perceptron: We start with the classic Perceptron algorithm [75, 70]. For simplicity, and without loss of generality we only consider the case of $\theta = 0$. We describe a slightly more general version of the Perceptron algorithm that approximately maximizes the margin and is referred to as Margin Perceptron [3]. The Margin Perceptron with parameter η works as follows. Initialize the weights $w^0 = 0^d$. At round $t \geq 1$, given a vector x^t and correct prediction $y^t \in \{-1, 1\}$, if $y^t \cdot \langle w^{t-1}, x^t\rangle \geq \eta$, then we let $w^t = w^{t-1}$. Otherwise, we update $w^t = w^{t-1} + y^t x^t$. The Perceptron algorithm corresponds to using this algorithm with $\eta = 0$. This update rule has the following guarantee:

Theorem 6.1 ([3]). *Let $(x^1, y^1), \ldots, (x^t, y^t)$ be any sequence of examples in $\mathcal{B}_2^d(R) \times \{-1, 1\}$ and assume that there exists a vector $w^* \in \mathcal{B}_2^d(W)$ such that for all t, $y^t\langle w^*, x^t\rangle \geq \gamma > 0$. Let M be the number of rounds in which the Margin Perceptron with parameter η updates the weights on this sequence of examples. Then $M \leq R^2W^2/(\gamma - \eta)^2$.*

The advantage of this version over the standard Perceptron is that it can be used to ensure that the final vector w^t separates the positive examples from the negative ones with margin η (as opposed to the plain Percetron which does not guarantee any margin). For example, by choosing $\eta = \gamma/2$ one can approximately maximize the margin while only paying a factor 4 in the upper bound on the number of updates. This means that the halfspace produced by Margin-Perceptron has essentially the same properties as that produced by the SVM algorithm.

In PAC learning of halfspaces with margin assumption we are given random examples from a distribution D over $\mathcal{B}_2^d(R) \times \{-1, 1\}$. The distribution is assumed to be supported only on examples (x, y) that for some vector w^* satisfy $y\langle w^*, x\rangle \geq \gamma$. It has long been observed that a natural way to convert the Perceptron algorithm to the SQ setting is to use the mean vector of all counterexamples with Perceptron updates [17, 14]. Namely, update using the example $(\bar{x}^t, 1)$, where $\bar{x}^t = \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y} \cdot \mathbf{x} \mid \mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta]$. Naturally, by linearity of the expectation, we have that $\langle w^{t-1}, \bar{x}^t\rangle < \eta$ and $\langle w^*, \bar{x}^t\rangle \geq \gamma$, and also, by convexity, that $\bar{x}^t \in \mathcal{B}_2^d(R)$. This implies that exactly the same analysis can be used for updates based on the mean counterexample vector. Naturally, we can only estimate $\bar{x}^t$ and hence our goal is to find an estimate that still allows the analysis to go through. In other words, we need to use statistical queries to find a vector $\tilde{x}$ which satisfies the conditions above (at least approximately). The main difficulty here is preserving the condition $\langle w^*, \tilde{x}\rangle \geq \gamma$, since we do not know w^*. However, by finding a vector $\tilde{x}$ such that $\|\tilde{x} - \bar{x}^t\|_2 \leq \gamma/(3W)$ we can ensure that

$$\langle w^*, \tilde{x}\rangle = \langle w^*, \bar{x}^t\rangle - \langle w^*, \bar{x}^t - \tilde{x}\rangle \geq \gamma - \|\tilde{x} - \bar{x}^t\|_2 \cdot \|w^*\|_2 \geq 2\gamma/3.$$

We next note that conditions $\langle w^{t-1}, \tilde{x}\rangle < \eta$ and $\tilde{x} \in \mathcal{B}_2^d(R)$ are easy to preserve. These are known and convex constraints so we can always project $\tilde{x}$ to the (convex) intersection of these two closed convex sets. This can only decrease the distance to $\bar{x}^t$. This implies that, given an estimate $\tilde{x}$, such that $\|\tilde{x} - \bar{x}^t\|_2 \leq \gamma/(3W)$ we can use Thm. 6.1 with $\gamma' = 2\gamma/3$ to obtain an upper bound of $M \leq R^2W^2/(2\gamma/3 - \eta)^2$ on the number of updates.

Now, by definition,

$$\mathop{\mathbf{E}}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y} \cdot \mathbf{x} \mid \mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta] = \frac{\mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y} \cdot \mathbf{x} \cdot \mathbf{1}_{\{\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta\}}]}{\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta]}.$$

In PAC learning with error ε we can assume that $\alpha \doteq \mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta] \geq \varepsilon$ since otherwise the halfspace $f_{w^{t-1}}$ is a sufficiently accurate hypothesis (that is classifies at least a $1 - \varepsilon$ fraction of examples with margin at least η). This implies that it is sufficient to find a vector $\tilde{z}$ such that $\|\tilde{z} - \bar{z}\|_2 \leq \alpha\gamma/(3W)$, where $\bar{z} = \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y} \cdot \mathbf{x} \cdot \mathbf{1}_{\{\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta\}}]$.

Now the distribution on $\mathbf{y} \cdot \mathbf{x} \cdot \mathbf{1}_{\{\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta\}}$ is supported on $\mathcal{B}_2^d(R)$ and therefore using Theorem 3.9 we can get the desired estimate using $2d$ queries to STAT($\Omega(\varepsilon\gamma/(RW))$). In other words, the estimation complexity of this implementation of Margin Perceptron is $O(RW/(\varepsilon\gamma)^2)$. We make a further observation that the dependence of estimation complexity on ε can be reduced from $1/\varepsilon^2$ to $1/\varepsilon$ by using VSTAT in place of STAT. This follows from Lemma 2.2 which implies that we need to pay only linearly for conditioning on $\mathbf{1}_{\{\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta\}}$. Altogether we get the following result which we for simplicity state for $\eta = \gamma/2$:

Theorem 6.2. *There exists an efficient algorithm* Margin-Perceptron-SQ *that for every $\varepsilon > 0$ and distribution D over $\mathcal{B}_2^d(R) \times \{-1, 1\}$ that is supported on examples (x, y) such that for some vector $w^* \in \mathcal{B}_2^d(W)$ satisfy $y\langle w^*, x\rangle \geq \gamma$, outputs a halfspace w such that $\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\langle w, \mathbf{x}\rangle < \gamma/2] \leq \varepsilon$.* Margin-Perceptron-SQ *uses $O(d(WR/\gamma)^2)$ queries to VSTAT($O((WR/\gamma)^2/\varepsilon)$).*

The estimation complexity of our algorithm is the same as the sample complexity of the PAC learning algorithm for learning large-margin halfspaces obtained via a standard online-to-batch conversion (*e.g.* [18]). SQ implementation of Perceptron were used to establish learnability of large-margin halfspaces with random classification noise [17] and to give a private version of Perceptron [15]. Perceptron is also the basis of SQ algorithms for learning halfspaces that do not require a margin assumption [14, 28]. All previous analyses that we are aware of used coordinate-wise estimation of $\bar{x}$ and resulted in estimation complexity bound of $O(d(WR/(\gamma\varepsilon)^2)$. Perceptron and SVM algorithms are most commonly applied over a very large number of variables (such as when using a kernel) and the dependence of estimation complexity on d would be prohibitive in such settings.

Online p-norm algorithms: The Perceptron algorithm can be seen as a member in the family of online p-norm algorithms [43] with $p = 2$. The other famous member of this family is the Winnow algorithm [59] which corresponds to $p = \infty$. For $p \in [2, \infty]$, a p-norm algorithm is based on p-margin assumption: there exists $w^* \in \mathcal{B}_q^d(R)$ such that for each example $(x, y) \in \mathcal{B}_p^d(R) \times \{-1, 1\}$ we have $y\langle w^*, x\rangle \geq \gamma$. Under this assumption the upper bound on the number of updates is $O((WR/\gamma)^2)$ for $p \in [2, \infty)$ and $O(\log d \cdot (WR/\gamma)^2)$ for $p = \infty$. Our ℓ_p mean estimation algorithms can be used in exactly the same way to (approximately) preserve the margin in this case giving us the following extension of Theorem 6.2.

Theorem 6.3. *For every $p \in [2, \infty]$, there exists an efficient algorithm p-norm-SQ that for every $\varepsilon > 0$ and distribution D over $\mathcal{B}_p^d(R) \times \{-1, 1\}$ that is supported on examples (x, y) that for some vector $w^* \in \mathcal{B}_q^d(W)$ satisfy $y\langle w^*, x\rangle \geq \gamma$, outputs a halfspace w such that $\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\langle w, \mathbf{x}\rangle < 0] \leq \varepsilon$. For $p \in [2, \infty)$ p-norm-SQ uses $O(d \log d (WR/\gamma)^2)$ queries to VSTAT$(O(\log d (WR/\gamma)^2/\varepsilon))$ and for $p = \infty$ p-norm-SQ uses $O(d \log d (WR/\gamma)^2)$ queries to VSTAT$(O((WR/\gamma)^2/\varepsilon))$.*

It is not hard to prove that margin can also be approximately maximized for these more general algorithms but we are not aware of an explicit statement of this in the literature. We remark that to implement the Winnow algorithm, the update vector can be estimated via straightforward coordinate-wise statistical queries.

Many variants of the Perceptron and Winnow algorithms have been studied in the literature and applied in a variety of settings (*e.g.* [40, 78, 22]). The analysis inevitably relies on a margin assumption (and its relaxations) and hence, we believe, can be implemented using SQs in a similar manner.

6.2 Local Differential Privacy

We now exploit the simulation of SQ algorithms by locally differentially private (LDP) algorithms [54] to obtain new LDP mean estimation and optimization algorithms.

We first recall the definition of local differential privacy. In this model it is assumed that each data sample obtained by an analyst is randomized in a differentially private way.

Definition 6.4. *An α-local randomizer $R : \mathcal{W} \to \mathcal{Z}$ is a randomized algorithm that satisfies $\forall w \in \mathcal{W}$ and $z_1, z_2 \in \mathcal{Z}$, $\mathbf{Pr}[R(w) = z_1] \leq e^\alpha\, \mathbf{Pr}[R(w) = z_2]$. An LR_D oracle for distribution D over $\mathcal{W}$ takes as an input a local randomizer R and outputs a random value z obtained by first choosing a random sample w from D and then outputting $R(w)$. An algorithm is α-local if it uses access only to LR_D oracle. Further, if the algorithm uses n samples such that sample i is obtained from α_i-randomizer R_i then $\sum_{i\in[n]} \alpha_i \leq \alpha$.*

The composition properties of differential privacy imply that an α-local algorithm is α-differentially private [30].

Kasiviswanathan et al. [54] show that one can simulate $\mathrm{STAT}_D(\tau)$ oracle with success probability $1 - \delta$ by an α-local algorithm using $n = O(\log(1/\delta)/(\alpha\tau)^2)$ samples from LR_D oracle. This has the following implication for simulating SQ algorithms.

Theorem 6.5 ([54])**.** *Let $\mathcal{A}_{SQ}$ be an algorithm that makes at most t queries to STAT$_D(\tau)$. Then for every $\alpha > 0$ and $\delta > 0$ there is an α-local algorithm $\mathcal{A}$ that uses $n = O(t \log(t/\delta)/(\alpha\tau^2))$ samples from LR_D oracle and produces the same output as $\mathcal{A}_{SQ}$ (for some answers of STAT$_D(\tau)$) with probability at least $1 - \delta$.*

Kasiviswanathan et al. [54] also prove a converse of this theorem that uses n queries to STAT$(\Theta(e^{2\alpha}\delta/n))$ to simulate n samples of an α-local algorithm with probability $1 - \delta$. The high accuracy requirement of this simulation implies that it is unlikely to give a useful SQ algorithm from an LDP algorithm.

Mean estimation: Duchi et al. [26] give α-local algorithms for ℓ_2 mean estimation using $O(d/(\varepsilon\alpha)^2)$ samples ℓ_∞ mean estimation using $O(d \log d/(\varepsilon\alpha)^2)$ samples (their bounds are for the expected error ε but we can equivalently treat them as ensuring error ε with probability at least $2/3$). They also prove that these bounds are tight. We observe that a direct combination of Thm. 6.5 with our mean estimation algorithms implies algorithms with nearly the same sample complexity (up to constants for $q = \infty$ and up to a $O(\log d)$ factor for $q = 2$). In addition, we can as easily obtain mean estimation results for other norms. For example we can fill the $q \in (2, \infty)$ regime easily.

Corollary 6.6. *For every α and $q \in [2, \infty]$ there is an α-local algorithm for ℓ_q mean estimation with error ε and success probability of at least $2/3$ that uses n samples from LR_D where:*

- *For $q = 2$ and $q = \infty$, $n = O(d \log d/(\alpha\varepsilon)^2)$.*
- *For $q \in (2, \infty)$, $n = O(d \log^2 d/(\alpha\varepsilon)^2)$.*

Convex optimization: Duchi et al. [27] give locally private versions of the mirror-descent algorithm for ℓ_1 setup and gradient descent for ℓ_2 setup. Their algorithms achieve the guarantees of the (non-private) stochastic versions of these algorithms at the expense of using $O(d/\alpha^2)$ times more samples. For example for the mirror-descent over the $\mathcal{B}_1^d$ the bound is $O(d \log d(RW/\varepsilon\alpha)^2)$ samples. α-local simulation of our algorithms from Sec. 4 can be used to obtain α-local algorithms for these problems. However such simulation leads to an additional factor corresponding to the number of iterations of the algorithm. For example for mirror-descent in ℓ_1 setup we will obtain and $O(d \log d/\alpha^2 \cdot (RW/\varepsilon)^4)$ bound. At the same time our results in Sec. 4 and Sec. 5 are substantially more general. In particular, our center-of-gravity-based algorithm (Thm. 5.8) gives the first α-local algorithm for stochastic convex bounded-range programs.

Corollary 6.7. *Let $\alpha > 0, \varepsilon > 0$. There is an α-local algorithm that for any convex body $\mathcal{K}$ given by a membership oracle with the guarantee that $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$ and any convex program $\min_{x\in\mathcal{K}} \mathbf{E}_{\mathbf{w}\sim D}[f(x, \mathbf{w})]$ in $\mathbb{R}^d$, where $\forall w$, $f(\cdot, w) \in \mathcal{F}(\mathcal{K}, B)$, with probability at least $2/3$, outputs an ε-optimal solution to the program in time poly$(d, \frac{B}{\alpha\varepsilon}, \log(R_1/R_0))$ and using $n = \tilde{O}(d^4 B^2/(\varepsilon^2\alpha^2))$ samples from LR_D.*

We note that a closely related application is also discussed in [9]. It relies on the random walk-based approximate value oracle optimization algorithm similar to the one we outlined in Sec. 5.1. Known optimization algorithms that use only the approximate value oracle require a substantially larger number of queries than our algorithm in Thm. 5.8 and hence need a substantially larger number of samples to implement (specifically, for the setting in Cor. 6.7, $n = \tilde{O}(d^{6.5} B^2/(\varepsilon^2\alpha^2))$ is implied by the algorithm given in [9]).

6.3 Differentially Private Answering of Convex Minimization Queries

An additional implication in the context of differentially private data analysis is to the problem of releasing answers to convex minimization queries over a single dataset that was recently studied by Ullman [89]. For a dataset $S = (w^i)_{i=1}^n \in \mathcal{W}^n$, a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and a family of convex functions $\mathcal{F} = \{f(\cdot, w)\}_{w \in \mathcal{W}}$ over $\mathcal{K}$, let $q_f(S) \doteq \mathrm{argmin}_{x \in \mathcal{K}} \frac{1}{n} \sum_{i \in [n]} f(x, w^i)$. Ullman [89] considers the question of how to answer sequences of such queries ε-approximately (that is by a point $\tilde{x}$ such that $\frac{1}{n} \sum_{i \in [n]} f(\tilde{x}, w^i) \leq q_f(S) + \varepsilon$).

We make a simple observation that our algorithms can be used to reduce answering of such queries to answering of counting queries. A *counting* query for a data set S, query function $\phi : \mathcal{W} \to [0, 1]$ and accuracy τ returns a value v such that $|v - \frac{1}{n} \sum_{i \in [n]} \phi(w^i)| \leq \tau$. A long line of research in differential privacy has considered the question of answering counting queries (see [29] for an overview). In particular, Hardt and Rothblum [46] prove that given a dataset of size $n \geq n_0 = O(\sqrt{\log(|\mathcal{W}|) \log(1/\beta)} \cdot \log t/(\alpha \tau^2)$ it is possible to (α, β)-differentially privately answer any sequence of t counting queries with accuracy τ (and success probability $\geq 2/3$).

Note that a convex minimization query is equivalent to a stochastic optimization problem when D is the uniform distribution over the elements of S (denote it by U_S). Further, a τ-accurate counting query is exactly a statistical query for $D = U_S$. Therefore our SQ algorithms can be seen as reductions from convex minimization queries to counting queries. Thus to answer t convex minimization queries with accuracy ε we can use the algorithm for answering $t' = tm(\varepsilon)$ counting queries with accuracy $\tau(\varepsilon)$, where $m(\varepsilon)$ is the number of queries to STAT$(\tau(\varepsilon))$ needed to solve the corresponding stochastic convex minimization problems with accuracy ε. The sample complexity of the algorithm for answering counting queries in [46] depends only logarithmically on t. As a result, the additional price for such implementation is relatively small since such algorithms are usually considered in the setting where t is large and $\log |\mathcal{W}| = \Theta(d)$. Hence the counting query algorithm in [46] together with the results in Corollary 4.7 immediately imply an algorithm for answering such queries that strengthens quantitatively and generalizes results in [89].

Corollary 6.8. *Let $p \in [1, 2]$, $L_0, R > 0$, $\mathcal{K} \subseteq \mathcal{B}_p^d(R)$ be a convex body and let $\mathcal{F} = \{f(\cdot, w)\}_{w \in \mathcal{W}} \subset \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{K}, L_0)$ be a finite family of convex functions. Let $\mathcal{Q}_\mathcal{F}$ be the set of convex minimization queries corresponding to $\mathcal{F}$. For any $\alpha, \beta, \varepsilon, \delta > 0$, there exists an (α, β)-differentially private algorithm that, with probability at least $1 - \delta$ answers any sequence of t queries from $\mathcal{Q}_\mathcal{F}$ with accuracy ε on datasets of size n for*

$$n \geq n_0 = \tilde{O}\left(\frac{(L_0 R)^2 \sqrt{\log(|\mathcal{W}|)} \cdot \log t}{\varepsilon^2 \alpha} \cdot \mathrm{polylog}\left(\frac{d}{\beta \delta} \right) \right).$$

For comparison, the results in [89] only consider the $p = 2$ case and the stated upper bound is

$$n \geq n_0 = \tilde{O}\left(\frac{(L_0 R)^2 \sqrt{\log(|\mathcal{W}|)} \cdot \max\{\log t, \sqrt{d}\}}{\varepsilon^2 \alpha} \cdot \mathrm{polylog}\left(\frac{1}{\beta \delta} \right) \right).$$

Our bound is a significant generalization and an improvement by a factor of at least $\tilde{O}(\sqrt{d}/\log t)$. Ullman [89] also shows that for generalized linear regression one can replace the $\sqrt{d}$ in the maximum by $L_0 R/\varepsilon$. The bound in Corollary 6.8 also subsumes this improved bound (in most parameter regimes of interest).

Finally, in the κ-strongly convex case (with $p = 2$), plugging our bounds from Corollary 4.11 into the algorithm in [46] we obtain that it suffices to use a dataset of size

$$n \geq n_0 = \tilde{O}\left(\frac{L_0^2 \sqrt{\log(|\mathcal{W}|)} \cdot \log(t \cdot d \cdot \log R)}{\varepsilon \alpha \kappa} \cdot \mathrm{polylog}\left(\frac{1}{\beta \delta} \right) \right).$$

The bound obtained by Ullman [89] for the same function class is

$$n_0 = \tilde{O}\left(\frac{L_0^2 R \sqrt{\log(|\mathcal{W}|)}}{\varepsilon \alpha} \cdot \max\left\{ \frac{\sqrt{d}}{\sqrt{\kappa \varepsilon}}, \frac{R \log t}{\varepsilon} \right\} \mathrm{polylog}\left(\frac{1}{\beta \delta} \right) \right).$$

Here our improvement over [89] is two-fold: We eliminate the $\sqrt{d}$ factor and we essentially eliminate the dependence on R (as in the non-private setting). We remark that our bound might appear incomparable to that in [89] but is, in fact, stronger since it can be assumed that $\kappa \geq \varepsilon/R^2$ (otherwise, bounds that do not rely on strong convexity are better).

6.4 Lower Bounds

We now describe a generic approach to combining SQ algorithms for stochastic convex optimization with lower bounds against SQ algorithms to obtain lower bounds against certain type of convex programs. These lower bounds are for problems in which we are given a set of cost functions $(v_i)_{i=1}^n$ from some collection of functions V over a set of "solutions" Z and the goal is to (approximately) minimize or maximize $\frac{1}{n}\sum_{i\in[n]} v_i(z)$ for $z \in Z$. Here either Z is non-convex or functions in V are non-convex (or both). Naturally, this captures loss (or error) of a model in machine learning and also the number of (un)satisfied constraints in constraint satisfaction problems (CSPs). For example, in the MAX-CUT problem $z \in \{0,1\}^d$ represents a subset of vertices and V consists of $\binom{d}{2}$, "$z_i \neq z_j$" predicates.

A standard approach to such non-convex problems is to map Z to a convex body $\mathcal{K} \subseteq \mathbb{R}^N$ and map V to convex functions over $\mathcal{K}$ in such a way that the resulting convex optimization problem can be solved efficiently and the solution allows one to recover a "good" solution to the original problem. For example, by ensuring that the mappings, $M : Z \to \mathcal{K}$ and $T : V \to \mathcal{F}$ satisfy: for all z and v, $v(z) = (T(v))(M(z))$ and for all instances of the problem $(v_i)_{i=1}^n$,

$$\min_{z\in Z} \frac{1}{n}\sum_{i\in[n]} v_i(z) - \min_{x\in\mathcal{K}} \frac{1}{n}\sum_{i\in[n]} (T(v_i))(x) < \varepsilon. \tag{20}$$

(Approximation is also often stated in terms of the ratio between the original and relaxed values and referred to as the integrality gap. This distinction will not be essential for our discussion.) The goal of lower bounds against such approaches is to show that specific mappings (or classes of mappings) will not allow solving the original problem via this approach, *e.g.* have a large integrality gap.

The class of convex relaxations for which our approach gices lower bounds are those that are "easy" for SQ algorithms. Accordingly, we define the following measure of complexity of convex optimization problems.

Definition 6.9. *For an SQ oracle $\mathcal{O}$, $t > 0$ and a problem P over distributions we say that $P \in$ Stat$(\mathcal{O}, t)$ if P can be solved using at most t queries to $\mathcal{O}$ for the input distribution. For a convex set $\mathcal{K}$, a set $\mathcal{F}$ of convex functions over $\mathcal{K}$ and $\varepsilon > 0$ we denote by Opt$(\mathcal{K}, \mathcal{F}, \varepsilon)$ the problem of finding, for every distribution D over $\mathcal{F}$, x^* such that $F(x^*) \leq \min_{x\in\mathcal{K}} F(x) + \varepsilon$, where $F(x) \doteq \mathbf{E}_{f\sim D}[f(x)]$.*

For simplicity, let's focus on the decision problem[6] in which the input distribution D belongs to $\mathcal{D} = \mathcal{D}_+ \cup \mathcal{D}_-$. Let $P(\mathcal{D}_+, \mathcal{D}_-)$ denote the problem of deciding whether the input distribution is in $\mathcal{D}_+$ or $\mathcal{D}_-$. This is a distributional version of a *promise* problem in which an instance can be of two types (for example completely satisfiable and one in which at most half of the constraints can be simultaneously satisfied). Statistical query complexity upper bounds are preserved under pointwise mappings of the domain elements and therefore an upper bound on the SQ complexity of a stochastic optimization problem implies an upper bound on any problem that can be reduced pointwise to the stochastic optimization problem.

Theorem 6.10. *Let $\mathcal{D}_+$ and $\mathcal{D}_-$ be two sets of distributions over a collection of functions V on the domain Z. Assume that for some $\mathcal{K}$ and $\mathcal{F}$ there exists a mapping $T : V \to \mathcal{F}$ such that for all $D \in \mathcal{D}^+$,*

[6] Indeed, hardness results for optimization are commonly obtained via hardness results for appropriately chosen decision problems.

$\min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)] > \alpha_+$ *and for all* $D \in \mathcal{D}^-$, $\min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)] \leq \alpha_-$. *Then if for an SQ oracle* $\mathcal{O}$ *and* t *we have a lower bound* $P(\mathcal{D}_+,\mathcal{D}_-) \notin$ *Stat*$(\mathcal{O},t)$ *then we obtain that Opt*$(\mathcal{K},\mathcal{F},\alpha_+ - \alpha_-) \notin$ *Stat*$(\mathcal{O},t)$.

The conclusion of this theorem, namely Opt$(\mathcal{K},\mathcal{F},\alpha_+ - \alpha_-) \notin$ Stat$(\mathcal{O},t)$, together with upper bounds from previous sections can be translated into a variety of concrete lower bounds on the dimension, radius, smoothness and other properties of convex relaxations to which one can map (pointwise) instances of $P(\mathcal{D}_+,\mathcal{D}_-)$. We also emphasize that the resulting lower bounds are structural and do not assume that the convex program is solved using an SQ oracle or efficiently.

Note that the assumptions on the mapping in Thm. 6.10 are stated for the expected value $\min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)]$ rather than for averages over given relaxed cost functions as in eq. (20). However these distributional settings are usually considered only when the number of available samples ensures that for every x the average over random samples $\frac{1}{n}\sum_{i\in[n]}(T(v_i))(x)$ is sufficiently close to the expectation $\mathbf{E}_{v\sim D}[(T(v))(x)]$ that the distinction does not matter.

Lower bounds for planted CSPs: We now describe an instantiation of this approach using lower bounds for constraint satisfaction problems established in [37]. Feldman et al. [37] describe implications of their lower bounds for convex relaxations using results from a preliminary version of this work (specifically Cor. 5.1) and discuss their relationship to those for lift-and-project hierarchies (Sherali-Adams, Lovász-Schrijver, Lasserre) of canonical LP/SDP formulations. To exemplify this approach, we give further implications based on our results for the first-order methods.

Let $Z = \{-1,1\}^d$ be the set of assignments to d Boolean variables. A distributional k-CSP problem is defined by a set $\mathcal{D}$ of distributions over Boolean k-ary predicates. One way to obtain a distribution over constraints is to first pick some assignment z and then generate random constraints that are consistent with z (or depend on z in some other predetermined way). In this way we can obtain a family of distributions $\mathcal{D}$ parameterized by a "planted" assignment z. Two standard examples of such instances are planted k-SAT (*e.g.* [21]) and the pseudorandom generator based on Goldreich's proposal for one-way functions [41].

Associated with every family created in this way is a complexity parameter r which, as shown in [37], characterizes the SQ complexity of finding the planted assignment z, or even distinguishing between a distribution in $\mathcal{D}$ and a uniform distribution over the same type of k-ary constraints. This is not crucial for discussion here but, roughly, the parameter r is the largest value r for which the generated distribution over variables in the constraint is $(r-1)$-wise independent. In particular, random and uniform k-XOR constraints (consistent with an assignment) have complexity k. The lower bound in [37] can be (somewhat informally) restated as follows.

Theorem 6.1 ([37]). *Let* $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^d}$ *be a set of "planted" distributions over* k*-ary constraints of complexity* r *and let* U_k *be the uniform distribution on (the same)* k*-ary constraints. Then any SQ algorithm that, given access to a distribution* $D \in \mathcal{D}\cup\{U_k\}$ *decides correctly whether* $D = D_z$ *or* $D = U_k$ *needs* $\Omega(t)$ *calls to VSTAT*$(\frac{d^r}{(\log t)^r})$ *for any* $t \geq 1$.

Combining this with Theorem 6.10 we get the following general statement:

Theorem 6.2. *Let* $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^d}$ *be a set of "planted" distributions over* k*-ary constraints of complexity* r *and let* U_k *be the uniform distribution on (the same)* k*-ary constraints. Assume that there exists a mapping* T *that maps each constraint* C *to a convex function* $f_C \in \mathcal{F}$ *over some convex* N*-dimensional set* $\mathcal{K}$ *such that for all* $z \in \{-1,1\}^d$, $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim D_z}[f_C(x)] \leq \alpha_-$ *and* $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim U_k}[f_C(x)] > \alpha_+$. *Then for every* $t \geq 1$, *Opt*$(\mathcal{K},\mathcal{F},\alpha_+ - \alpha_-) \notin$ *Stat*$($*VSTAT*$(\frac{d^r}{(\log t)^r}),\Omega(t))$.

Note that in the context of convex minimization that we consider here it is more natural to think of the relaxation as minimizing the number of unsatisfied constraints (although if the objective function is linear then the claim also applies to maximization over $\mathcal{K}$). We now instantiate this statement for solving the k-SAT problem via a convex program in the class $\mathcal{F}^0_{\|\cdot\|_p}(\mathcal{B}^N_p, 1)$ (see Sec. 4). Let $\mathcal{C}_k$ denote the set of all k-clauses (OR of k distinct variables or their negations). Let U_k be the uniform distribution over $\mathcal{C}_k$.

Corollary 6.11. *There exists a family of distributions $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^d}$ over $\mathcal{C}_k$ such that the support of D_z is satisfied by z with the following property: For every $p \in [1,2]$, if there exists a mapping $T : \mathcal{C}_k \to \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{B}^N_p, 1)$ such that for all z, $\min_{x\in\mathcal{B}^N_p} \mathbf{E}_{C\sim D_z}[(T(C))(x)] \leq 0$ and $\min_{x\in\mathcal{B}^N_p} \mathbf{E}_{C\sim U_k}[(T(C))(x)] > \alpha$ then $\alpha = \tilde{O}\left((d/\log(N))^{-k/2}\right)$.*

This lower bound excludes embeddings in exponentially high (*e.g.* $2^{d^{1/4}}$) dimension for which the value of the program for unsatisfiable instances differs from that for satisfiable instances by more than $d^{-k/4}$ (note that the range of functions in $\mathcal{F}^0_{\|\cdot\|_p}(\mathcal{B}^N_p, 1)$ can be as large as $[-1,1]$ so this is a normalized additive gap). For comparison, in the original problem the the values of these two types of instances are 1 and $\approx 1 - 2^{-k}$. In particular, this implies that the integrality gap is $1/(1-2^{-k}) - o(1)$ (which is optimal).

Acknowledgements

We thank Arkadi Nemirovski, Sasha Rakhlin, Ohad Shamir and Karthik Sridharan for discussions and valuable suggestions about this work.

References

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

[2] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *STOC*, pages 156–163, 1991.

[3] M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *ICML*, pages 73–80, 2006.

[4] M.-F. Balcan and V. Feldman. Statistical active learning algorithms. In *NIPS*, pages 1295–1303, 2013.

[5] M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In *COLT*, pages 26.1–26.22, 2012.

[6] K. Ball, E. Carlen, and E.H. Lieb. Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones mathematicae*, 115(1):463–482, 1994.

[7] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014.

[8] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. *CoRR*, abs/1511.02513, 2015. URL `http://arxiv.org/abs/1511.02513`.

[9] A. Belloni, T. Liang, H. Narayanan, and A. Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. *CoRR*, abs/1501.07242, 2015. URL `http://arxiv.org/abs/1501.07242`.

[10] S. Ben-David, A. Itai, and E. Kushilevitz. Learning by distances. In *COLT*, pages 232–245, 1990.

[11] A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization. `http://www2.isye.gatech.edu/~nemirovs/`, 2013.

[12] D. Bertsimas and S. Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, July 2004.

[13] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *STOC*, pages 253–262, 1994.

[14] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.

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

[16] G. Braun, C. Guzmán, and S. Pokutta. Lower Bounds on the Oracle Complexity of Convex Optimization Via Information Theory. arXiv:1407.5144, 2014.

[17] T. Bylander. Learning linear threshold functions in the presence of classification noise. In *COLT*, pages 340–347, 1994.

[18] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[19] K. Chaudhuri, C. Monteleoni, and A. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.

[20] C. Chu, S. Kim, Y. Lin, Y. Yu, G. Bradski, A. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, pages 281–288, 2006.

[21] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.

[22] S. Dasgupta, A.T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

[23] A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19 (3):1171–1183, 2008.

[24] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods with inexact oracle: the strongly convex case. CORE Discussion Papers 2013016, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2013. URL `http://EconPapers.repec.org/RePEc:cor:louvco:2013016`.

[25] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1-2):37–75, 2014.

[26] J. Duchi, M.I. Jordan, and M.J. Wainwright. Local privacy and statistical minimax rates. In *FOCS*, pages 429–438, 2013.

[27] J. Duchi, M.I. Jordan, and M.J. Wainwright. Privacy aware learning. *J. ACM*, 61(6):38, 2014.

[28] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Math. Program.*, 114(1):101–114, 2008.

[29] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy (preprint)*. Now Publishers Inc, 2014.

[30] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[31] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506.02629, 2015. URL `http://arxiv.org/abs/1506.02629`.

[32] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. *CoRR*, abs/1411.2664, 2014. Extended abstract in STOC 2015.

[33] V. Feldman. Evolvability from learning algorithms. In *Proceedings of STOC*, pages 619–628, 2008.

[34] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. In *Proceedings of FOCS*, pages 375–384, 2009.

[35] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

[36] V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. Statistical algorithms and a lower bound for planted clique. In *STOC*, pages 655–664. ACM, 2013.

[37] V. Feldman, W. Perkins, and S. Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. Extended abstract in STOC 2015.

[38] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. In *STOC*, pages 95–106, 2012.

[39] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.

[40] Y. Freund and R. Schapire. Large margin classification using the Perceptron algorithm. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 209–217, 1998.

[41] Oded Goldreich. Candidate one-way functions based on expander graphs. *IACR Cryptology ePrint Archive*, 2000:63, 2000.

[42] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[43] A. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 171–183, 1997.

[44] B. Grunbaum. Partitions of mass-distributions and convex bodies by hyperplanes. *Pacific J. Math.*, 10: 1257–1261, 1960.

[45] C. Guzmán and A. Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1 – 14, 2015. ISSN 0885-064X. doi: http://dx.doi.org/10.1016/j.jco.2014.08.003. URL `http://www.sciencedirect.com/science/article/pii/S0885064X14000831`.

[46] M. Hardt and G. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

[47] M. Hardt and J. Ullman. Preventing false discovery in interactive data analysis is hard. In *FOCS*, pages 454–463, 2014.

[48] D. Hsu and S. Sabato. Approximate loss minimization with heavy tails. *CoRR*, abs/1307.1827, 2013. URL `http://arxiv.org/abs/1307.1827`.

[49] F. John. Extremum problems with inequalities as subsidiary conditions. Studies Essays, pres. to R. Courant, 187-204 (1948)., 1948.

[50] S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800. Curran Associates, Inc., 2008.

[51] A. T. Kalai and S. Vempala. Simulated annealing for convex optimization. *Math. Oper. Res.*, 31(2): 253–266, 2006.

[52] R. Kannan, L. Lovász, and M. Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13(3-4):541–559, 1995.

[53] B. Kashin. The widths of certain finite dimensional sets and classes of smooth functions. *Izv. Akad. Nauk SSSR Ser. Mat. (in Russian)*, pages 334–351, 1977.

[54] S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, June 2011.

[55] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[56] L. G. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21(2):307–320, 1996.

[57] J.R. Lee, P. Raghavendra, and D. Steurer. Lower bounds on the size of semidefinite programming relaxations. In *STOC*, pages 567–576, 2015.

[58] A.Yu. Levin. On an algorithm for the minimization of convex functions. *Sov. Math., Dokl.*, 6:268–290, 1965. ISSN 0197-6788.

[59] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

[60] L. Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50. SIAM, 1987.

[61] L. Lovász and S. Vempala. Hit-and-run from a corner. *SIAM J. Computing*, 35:985–1005, 2006.

[62] L. Lovász and S. Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *FOCS*, pages 57–68, 2006.

[63] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. Syst. Sci.*, 72(2):392–417, 2006.

[64] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007. doi: 10.1002/rsa.20135. URL `http://dx.doi.org/10.1002/rsa.20135`.

[65] Y. Lyubarskii and R. Vershynin. Uncertainty principles and vector quantization. *Information Theory, IEEE Transactions on*, 56(7):3491–3501, 2010.

[66] R. Meka, A. Potechin, and A. Wigderson. Sum-of-squares lower bounds for planted clique. In *STOC*, pages 87–96, 2015.

[67] A. Nemirovski. Efficient Methods in Convex Programming. `http://www2.isye.gatech.edu/~nemirovs/`, 1994.

[68] A.S. Nemirovsky and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley @ Sons, New York, 1983.

[69] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[70] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[71] G. Pisier. Martingales in Banach spaces (in connections with Type and Cotype). Course IHP, 2011.

[72] B.T. Poljak. *Introduction to Optimization*. Optimization Software, 1987.

[73] M. Raginsky and A. Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 57(10):7036–7056, 2011.

[74] R.T. Rockafellar. *Conjugate Duality and Optimization*. Regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1974.

[75] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[76] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *STOC*, pages 765–774, 2010.

[77] T. Rothvoß. The matching polytope has exponential extension complexity. In *STOC*, pages 263–272, 2014.

[78] R. Servedio. On pac learning using winnow, perceptron, and a perceptron-like algorithm. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 296–307, 1999.

[79] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 1107057132, 9781107057135.

[80] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[81] A.A. Sherstov. Halfspace matrices. *Computational Complexity*, 17(2):149–178, 2008.

[82] N.Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*. Nonconvex Optimization and Its Applications. Springer US, 2011.

[83] H. Simon. A characterization of strong learnability in the statistical query model. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 393–404, 2007.

[84] N. Srebro and A. Tewari. Stochastic optimization: ICML 2010 tutorial. `http://www.ttic.edu/icml2010stochopt/`, 2010.

[85] J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:126, 2015. URL `http://eccc.hpi-web.de/report/2015/126`.

[86] T. Steinke and J. Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *COLT*, pages 1588–1628, 2015.

[87] C. Studer, T. Goldstein, W. Yin, and R. Baraniuk. Democratic representations. *CoRR*, abs/1401.3420, 2014. URL `http://arxiv.org/abs/1401.3420`.

[88] B. Szörényi. Characterizing statistical query learning: Simplified notions and proofs. In *Proceedings of ALT*, pages 186–200, 2009.

[89] J. Ullman. Private multiplicative weights beyond linear queries. In *PODS*, pages 303–312, 2015.

[90] L. G. Valiant. Evolvability. *Journal of the ACM*, 56(1):3.1–3.21, 2009. Earlier version in ECCC, 2006.

[91] P. Valiant. Evolvability of real functions. *TOCT*, 6(3):12.1–12.19, 2014.

[92] S.L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[93] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010.

A Uniform convexity, uniform smoothness and consequences

A space $(E, \|\cdot\|)$ is r-uniformly convex if there exists constant $0 < \delta \leq 1$ such that for all $x, y \in E$

$$\|x\|^r + \delta \|y\|^r \leq \frac{\|x+y\|^r + \|x-y\|^r}{2}. \tag{21}$$

From classical inequalities (see, e.g., [6]) it is known that ℓ_p^d for $1 < p < \infty$ is r-uniformly convex for $r = \max\{2, p\}$. Furthermore,

- When $p = 1$, the function $\Psi(x) = \frac{1}{2(p(d)-1)}\|x\|_{p(d)}^2$ (with $p(d) = 1 + 1/\ln d$) is 2-uniformly convex w.r.t. $\|\cdot\|_1$;
- When $1 < p \leq 2$, the function $\Psi(x) = \frac{1}{2(p-1)}\|x\|_p^2$ is 2-uniformly convex w.r.t. $\|\cdot\|_p$;
- When $2 < p < \infty$, the function $\Psi(x) = \frac{2^{p-2}}{p}\|x\|_p^p$ is p-uniformly convex w.r.t. $\|\cdot\|_p$.

By duality, a Banach space $(E, \|\cdot\|)$ being r-uniformly convex is equivalent to the dual space $(E^*, \|\cdot\|_*)$ being s-uniformly smooth, where $1/r + 1/s = 1$. This means there exists a constant $C \geq 1$ such that for all $w, z \in E^*$

$$\frac{\|w+z\|_*^s + \|w-z\|_*^s}{2} \leq \|w\|_*^s + C\|z\|_*^s. \tag{22}$$

In the case of ℓ_p^d space we obtain that its dual ℓ_q^d is s-uniformly smooth for $s = \min\{2, q\}$. Furthermore, when $1 < q \leq 2$ the norm $\|\cdot\|_q$ satisfies (22) with $s = q$ and $C = 1$; when $2 \leq q < \infty$, the norm $\|\cdot\|_q$ satisfies (22) with $s = 2$ and $C = q-1$. Finally, observe that for ℓ_∞^d we can use the equivalent norm $\|\cdot\|_{q(d)}$, with $q(d) = \ln d + 1$:

$$\|x\|_\infty \leq \|x\|_{q(d)} \leq e\,\|x\|_\infty,$$

and this equivalent norm satisfies (22) with $s = 2$ and $C = q(d) - 1 = \ln d$, that grows only moderately with dimension.

B Sample complexity of mean estimation

The following is a standard analysis based on Rademacher complexity and uniform convexity (see, e.g., [71]). Let $(E, \|\cdot\|)$ be an r-uniformly convex space. We are interested in the convergence of the empirical mean to the true mean in the dual norm (to the one we optimize in). By Observation 3.1 this is sufficient to bound the error of optimization using the empirical estimate of the gradient on $\mathcal{K} \doteq \mathcal{B}_{\|\cdot\|}$.

Let $(\mathbf{w}^j)_{j=1}^n$ be i.i.d. samples of a random variable $\mathbf{w}$ with mean $\bar{w}$, and let $\bar{\mathbf{w}}^n \doteq \frac{1}{n}\sum_{j=1}^n \mathbf{w}^j$ be the empirical mean estimator. Notice that

$$\|\bar{\mathbf{w}}^n - \bar{w}\|_* = \sup_{x\in\mathcal{K}} |\langle \bar{\mathbf{w}}^n - \bar{w}, x\rangle| .$$

Let $(\sigma_j)_{j=1}^n$ be i.i.d. Rademacher random variables (independent of $(\mathbf{w}^j)_j$). By a standard symmetrization argument, we have

$$\mathop{\mathbf{E}}_{\mathbf{w}^1,\ldots,\mathbf{w}^n} \sup_{x\in\mathcal{K}} \left| \left\langle \frac{1}{n}\sum_{j=1}^n \mathbf{w}^j, x \right\rangle - \langle \bar{w}, x\rangle \right| \quad\leq\quad 2 \mathop{\mathbf{E}}_{\sigma_1,\ldots,\sigma_n} \mathop{\mathbf{E}}_{\mathbf{w}^1,\ldots,\mathbf{w}^n} \sup_{x\in\mathcal{K}} \left| \sum_{j=1}^n \sigma_j \langle \mathbf{w}^j, x\rangle \right| .$$

For simplicity, we will denote $\|\mathcal{K}\| \doteq \sup_{x\in\mathcal{K}} \|x\|$ the $\|\cdot\|$ radius of $\mathcal{K}$. Now by the Fenchel inequality

$$\begin{aligned}
\mathop{\mathbf{E}}_{\sigma_1,\ldots,\sigma_n} \sup_{x\in\mathcal{K}} \left| \sum_{j=1}^n \sigma_j \langle \mathbf{w}^j, x\rangle \right| \quad&\leq\quad \inf_{\lambda>0} \mathop{\mathbf{E}}_{\sigma_1,\ldots,\sigma_n} \left\{ \frac{1}{r\lambda} \sup_{x\in\mathcal{K}} \|x\|^r + \frac{1}{s\lambda} \left\| \frac{\lambda}{n} \sum_{j=1}^n \sigma_j \mathbf{w}^j \right\|_*^s \right\} \\
&\leq\quad \inf_{\lambda>0} \mathop{\mathbf{E}}_{\sigma_1,\ldots,\sigma_{n-1}} \left\{ \frac{1}{r\lambda}\|\mathcal{K}\|^r \right. \\
&\qquad \left. + \frac{\lambda^{s-1}}{sn^s}\frac{1}{2} \left[\left\| \sum_{j=1}^{n-1} \sigma_j \mathbf{w}^j + \sigma_n \mathbf{w}^n \right\|_*^s + \left\| \sum_{j=1}^{n-1} \sigma_j \mathbf{w}^j - \sigma_n \mathbf{w}^n \right\|_*^s \right] \right\} \\
&\leq\quad \inf_{\lambda>0} \mathop{\mathbf{E}}_{\sigma_1,\ldots,\sigma_{n-1}} \left\{ \frac{1}{r\lambda}\|\mathcal{K}\|^r + \frac{\lambda^{s-1}}{sn^s} \left[\left\| \sum_{j=1}^{n-1} \sigma_j \mathbf{w}^j \right\|_*^s + C\|\mathbf{w}^n\|_*^s \right] \right\},
\end{aligned}$$

where the last inequality holds from the s-uniform smoothness of $(E^*, \|\cdot\|_*)$. Proceeding inductively we obtain

$$\mathop{\mathbf{E}}_{\sigma_1,\dots,\sigma_n} \sup_{x\in\mathcal{K}} \left|\sum_{j=1}^n \sigma_j \langle \mathbf{w}^j, x\rangle\right| \quad\leq\quad \inf_{\lambda>0} \left\{ \frac{1}{r\lambda}\|\mathcal{K}\|^r + \frac{C\lambda^{s-1}}{sn^s}\sum_{j=1}^n \|\mathbf{w}^j\|_*^s \right\}.$$

It is a straightforward computation to obtain the optimal $\bar{\lambda} = \frac{\|\mathcal{K}\|^{r-1}n}{C^{1/s}\left(\sum_j \|\mathbf{w}^j\|_*^s\right)^{1/s}}$, which gives an upper bound

$$\mathop{\mathbf{E}}_{\sigma_1,\dots,\sigma_n} \sup_{x\in\mathcal{K}} \left|\sum_{j=1}^n \sigma_j \langle \mathbf{w}^j, x\rangle\right| \leq \frac{1}{n^{1/r}} C^{1/s} \sup_{x\in\mathcal{K}} \|x\| \left(\frac{1}{n}\sum_{j=1}^n \|\mathbf{w}^j\|_*^s\right)^{1/s}.$$

By simply upper bounding the quantity above by $\varepsilon > 0$, we get a sample complexity bound for achieving ε accuracy in expectation, $n = \lceil C^{r/s}/\varepsilon^r \rceil$, where $C \geq 1$ is any constant satisfying (22). For the standard ℓ_p^d-setup, i.e., where $(E, \|\cdot\|) = (\mathbb{R}^d, \|\cdot\|_p)$, by the parameters of uniform convexity and uniform smoothness provided in Appendix A, we obtain the following bounds on sample complexity:

(i) For $p = 1$, we have $r = s = 2$ and $C = \ln d$, by using the equivalent norm $\|\cdot\|_{p(d)}$. This implies that $n = O\left(\frac{\ln d}{\varepsilon^2}\right)$ samples suffice.

(ii) For $1 < p \leq 2$, we have $r = s = 2$ and $C = q - 1$. This implies that $n = \left\lceil \frac{q-1}{\varepsilon^2} \right\rceil$ samples suffice.

(iii) For $2 < p < \infty$, we have $r = p$, $s = q$ and $C = 1$. This implies that $n = \left\lceil \frac{1}{\varepsilon^p} \right\rceil$ samples suffice.

C Proof of Corollary 4.7

Note that by Proposition 4.5 in order to obtain an ε-optimal solution to a non-smooth convex optimization problem it suffices to choose $\eta = \varepsilon/2$, and $T = \lceil r2^r L_0^r D_\Psi(\mathcal{K})/\varepsilon^r \rceil$. Since $\mathcal{K} \subseteq \mathcal{B}_p(R)$, to satisfy (9) it is sufficient to have for all $y \in \mathcal{B}_p(R)$,

$$\langle \nabla F(x) - \tilde{g}(x), y\rangle \leq \eta/2.$$

Maximizing the left hand side on y, we get a sufficient condition: $\|\nabla F(x) - \tilde{g}(x)\|_q R \leq \eta/2$. We can satisfy this condition by solving the mean estimation problem in ℓ_q-norm with error $\eta/[2L_0R] = \varepsilon/[4L_0R]$ (recall that $f(\cdot, w)$ is L_0 Lipschitz w.r.t. $\|\cdot\|_p$). Next, using the uniformly convex functions for ℓ_p from Appendix A, together with the bound on the number of queries and error for the mean estimation problems in ℓ_q-norm from Section 3.1, we obtain that the total number of queries and the type of queries we need for stochastic optimization in the non-smooth ℓ_p-setup are:

- $p = 1$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = \frac{e^2 \ln d}{2} R^2$. As a consequnce, solving the convex program amounts to using $O\left(d \cdot \left(\frac{L_0 R}{\varepsilon}\right)^2 \ln d\right)$ queries to STAT $\left(\frac{\varepsilon}{4L_0R}\right)$.
- $1 < p < 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = \frac{1}{2(p-1)} R^2$. As a consequence, solving the convex program amounts to using $O\left(d \log d \cdot \frac{1}{(p-1)} \left(\frac{L_0R}{\varepsilon}\right)^2\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{[\log d] L_0 R}\right)\right)$.

- $p = 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = R^2$. As a consequence, solving the convex program amounts to using $O\left(d \cdot \left(\frac{L_0 R}{\varepsilon}\right)^2\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{L_0 R}\right)\right)$.
- $2 < p < \infty$**:** We may choose $r = p$, $D_\Psi(\mathcal{K}) = \frac{2^{p-2}}{p} R^p$. As a consequence, solving the convex program amounts to using $O\left(d \log d \cdot 2^{2p-2} \left(\frac{L_0 R}{\varepsilon}\right)^p\right)$ queries to VSTAT $\left(\left(\frac{64 L_0 R \log d}{\varepsilon}\right)^p\right)$.

□

D Proof of Corollary 4.9

Similarly as in Appendix C, given $x \in \mathcal{K}$, we can obtain $\tilde{g}(x)$ by mean estimation problem in ℓ_q-norm with error $\varepsilon/[12 L_0 R]$ (notice we have chosen $\eta = \varepsilon/6$).

Now, by Proposition 4.8, in order to obtain an ε-optimal solution it suffices to run the accelerated method for $T = \left\lceil \sqrt{2 L_1 D_\Psi(\mathcal{K})/\varepsilon} \right\rceil$ iterations, each of them requiring $\tilde{g}$ as defined above. By using the 2-uniformly convex functions for ℓ_p, with $1 \leq p \leq 2$, from Appendix A, together with the bound on the number of queries and error for the mean estimation problems in ℓ_q-norm from Section 3.1, we obtain that the total number of queries and the type of queries we need for stochastic optimization in the smooth ℓ_p-setup is:

- $p = 1$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = \frac{e^2 \ln d}{2} R^2$. As a consequnce, solving the convex program amounts to using $O\left(d \cdot \sqrt{\ln d \cdot \frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\frac{\varepsilon}{12 L_0 R}\right)$.
- $1 < p < 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = \frac{1}{2(p-1)} R^2$. As a consequence, solving the convex program amounts to using $O\left(d \log d \cdot \sqrt{\frac{1}{(p-1)} \cdot \frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{[\log d] L_0 R}\right)\right)$;
- $p = 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = R^2$. As a consequence, solving the convex program amounts to using $O\left(d \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}}\right)$ queries to STAT $\left(\Omega\left(\frac{\varepsilon}{L_0 R}\right)\right)$.

□