http://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 46, No. 3, August 2021, pp. 912–945
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Statistical Query Algorithms for Mean Vector Estimation and Stochastic Convex Optimization

Vitaly Feldman,[a] Cristóbal Guzmán,[b] Santosh Vempala[c,d]

[a] Apple, Inc., Cupertino, California 95014; [b] Institute for Mathematical and Computational Engineering, Facultad de Matemáticas & Escuela de Ingeniería, Pontificia Universidad Católica de Chile, Region Metropolitana, 3580000 Santiago, Chile; [c] ANID – Millennium Science Initiative Program – Millennium Nucleus Center for the Discovery of Structures in Complex Data, 7820244 Santiago, Chile; [d] School of Computer Science, Georgia Institute of Technology, Atlanta, Georgia 30332

Contact: vitaly@post.harvard.edu (VF); crguzman@mat.uc.cl, https://orcid.org/0000-0002-1498-2055 (CG); vempala@gatech.edu (SV)

Received: June 23, 2018
Accepted: September 27, 2020
Published Online in Articles in Advance: March 8, 2021

MSC2000 Subject Classification: Primary: 90C15; secondary: 68Q32
OR/MS Subject Classification: Primary: Programming/stochastic

https://doi.org/10.1287/moor.2020.1111

Abstract. Stochastic convex optimization, by which 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 first-order iterative 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. The key ingredient of our work is SQ algorithms and lower bounds for estimating the mean vector of a distribution over vectors supported on a convex body in R^d. This natural problem has not been previously studied, and we show that our solutions can be used to get substantially improved SQ versions of Perceptron and other online algorithms for learning halfspaces.

Funding: The work of C. Guzmán was partially supported by Grant FONDECYT 11160939, and by ANID – Millennium Science Initiative Program – NCN17_059.

Keywords: stochastic • programming • convex • nonlinear • data • statistics

1. Introduction

Statistical query (SQ) algorithms, defined by Kearns [58] in the context of probably approximately correct (PAC) learning and by Feldman et al. [40] for general problems on inputs sampled independently and identically from distributions (i.i.d.), 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 (possibly dependent on ϕ). We 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*.

Statistical query access to data was introduced as a means to derive noise-tolerant algorithms in the PAC model of learning (Kearns [58]). Subsequently, it was realized that reducing data access to estimation of simple expectations has a wide variety of additional useful properties. It plays a key role in the development of the notion of differential privacy (Blum et al. [12], Dinur and Nissim [26], Dwork et al [31]) and is the subject of intense subsequent research in differential privacy[1] (see Dwork and Roth [30] for a literature review). It has important applications in a large number of other theoretical and practical contexts, such as distributed data access (Chu et al. [20], Roy et al. [78], Sujeeth et al. [90]), evolvability (Feldman [35, 36], Valiant [92]), and memory-/communication-limited machine learning (Balcan et al. [4], Steinhardt et al. [87]). Most recently, in a line of work initiated by Dwork et al. [32], SQs are used as a basis for understanding generalization in adaptive data analysis (Bassily et al [7], Dwork et al. [32, 33], Hardt and Ullman [49], Steinke and Ullman [88]).

Here, we consider the complexity of solving stochastic convex minimization problems by SQ algorithms. 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., Shalev-Shwartz and Ben-David [81], Srebro and Tewari [86]). 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 $F(x)$.

To the best of our knowledge, prior to this work, the complexity of this fundamental class of problems has not been studied in the SQ model. 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 as well as in a wide variety of other oracle models.

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. The first one is shown by Kearns [58] who proves that parity functions cannot be learned efficiently using SQs. Subsequent work develops several techniques for proving such lower bounds (e.g., Blum et al. [14], Feldman et al. [39, 40], Simon [85]), establishes relationships to other complexity measures (e.g., Kallweit and Simon [55], Sherstov [83]), and provides lower bounds for many important problems in learning theory (e.g., Blum et al. [14], Feldman et al. [38], Klivans and Sherstov [60]) and beyond (Bresler et al. [16], Feldman et al. [39, 40], Wang et al. [94]).

From this perspective, statistical algorithms for stochastic convex optimization have another important role. 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., Fiorini et al. [41], Lee et al. [61], Meka et al. [67], Rothvoß [77]). As shown in Feldman et al. [39], it is possible to convert SQ lower bounds into purely structural lower bounds on convex relaxations—in other words, lower bounds that hold without assumptions on the algorithm that is used to solve the problem (in particular, not just SQ algorithms). From this point of view, each SQ implementation of a convex optimization algorithm is a new lower bound against the corresponding convex relaxation of the problem.

1.1. Overview of Results

We focus on iterative first-order methods, namely, techniques that rely on updating the current point x^t using only the (sub)gradient of F at x^t. These are among the most widely used approaches for solving convex programs in theory and practice. 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 never been formally addressed.

We start with the case of linear optimization; namely, $\nabla F(x)$ is the same over the whole body $\mathcal{K}$. It turns out that, in this case, global approximation of the gradient (that is, one for which the linear approximation of F given by the estimated gradient is ε close to the true linear approximation of F) is sufficient. This means that the question becomes that of estimating the mean vector of a distribution over vectors in $\mathbb{R}^d$ in some norm that depends on the geometry of $\mathcal{K}$. This is a basic question (indeed, central to many high-dimensional problems), but it has not been carefully addressed even for the simplest norms, such as ℓ_2. We examine it in detail and provide an essentially complete picture for all ℓ_q norms with $q \in [1, \infty]$. We also briefly examine the case of general convex bodies (and corresponding norms) and provide some universal bounds.

The analysis of the preceding linear case gives us the basis for tackling first-order optimization methods for Lipschitz convex functions. That is, we can now obtain an estimate of the expected gradient at each iteration. However, we still need to determine whether the global approximation is needed or a local one would suffice and also 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 first-order oracles. Methods of this type have a long history (e.g., Poljak [72], Shor [84]); however, some of our methods of choice have only been studied recently. We give SQ algorithms for implementing the global and local oracles and then systematically study several traditional setups of convex optimization: nonsmooth, smooth, and strongly convex. Although that is not the most exciting task in itself, it serves to show the generality of our approach. Remarkably, in all of these common setups, we achieve the same estimation complexity as that known to be achievable with samples.

All of the previous results require that the optimized functions are Lipschitz; that is, the gradients are bounded in the appropriate norm (and the complexity depends polynomially on the bound). Addressing

non-Lipschitz optimization seems particularly challenging in the stochastic case and SQ model, in particular. Indeed, direct SQ implementation of some techniques would require queries of exponentially high accuracy. We give two approaches for dealing with this problem that require only that the convex functions in the support of distribution have a bounded range. The first one avoids gradients altogether by only using estimates of function values. It is based on the random walk techniques of Kalai and Vempala [54] and Lovász and Vempala [64]. The second one is based on a new analysis of the classic center-of-gravity method. There, we show that there exists a local norm, specifically that given by the inertial ellipsoid, that allows us to obtain a global approximation relatively cheaply. Interestingly, these very different methods have the same estimation complexity, which is also within a factor of d of our lower bound.

Finally, we highlight some theoretical applications of our results. First, we describe a high-level methodology of obtaining a lower bound for convex relaxations from our results and give an example for constraint-satisfaction problems. We then show that our mean estimation algorithms can greatly improve the estimation complexity of the SQ version of the classic Perceptron algorithm and several related algorithms. Finally, we give corollaries for two problems in differential privacy: (a) new algorithms for solving convex programs with the stringent local differential privacy and (b) strengthening and generalization of algorithms for answering sequences of convex minimization queries differentially privately given by Ullman [91].

1.2. Linear Optimization and 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 $\widetilde{w}$, such that, for every $x \in \mathcal{K}$, $|\langle x, \bar{w} \rangle - \langle x, \widetilde{w} \rangle| \leq \varepsilon$. Given such an estimate $\widetilde{w}$, we can solve the original problem with error of at most 2ε by solving $\min_{x \in \mathcal{K}} \langle x, \widetilde{w} \rangle$ (see Observation 1).

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, in which 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 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., Balcan and Feldman [3], Blum et al. [13], Bylander [17]).

The seemingly simple question we would like to answer is whether the SQ estimation complexity is different from the 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 $\mathcal{W}$ (cf. Pisier [71]). Also, some of the general techniques for proving upper bounds on sample complexity (see Appendix B) appeal directly to high-dimensional concentration and do not seem to extend to the intrinsically one-dimensional SQ model. We, therefore, focus our attention on the much more benign and well-studied ℓ_p/ℓ_q setting. That is, $\mathcal{K}$ is a unit ball in ℓ_p norm, $\mathcal{W}$ 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 $\|\widetilde{w} - \bar{w}\|_q \leq \varepsilon$ for a random variable $\mathbf{w}$ supported on the unit ℓ_q ball, and we refer to it as the ℓ_q mean estimation. Even in this standard setting, the picture is not so clear in the regime when $q \in [1, 2)$, where the sample complexity of ℓ_q mean estimation depends on both q and the relationship between d and ε.

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. The upper bounds are based on several different algorithms.

- For $q = \infty$, straightforward coordinate-wise estimation gives the desired guarantees.
- For $q = 2$, we demonstrate that Kashin's representation of vectors introduced by Lyubarskii and Vershynin [66] gives a set of $2d$ measurements that allow us to recover the mean with estimation complexity of $O(1/\varepsilon^2)$. We also give a randomized algorithm based on estimating the truncated coefficients of the mean on 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 nonzero coefficients have low dynamic range. For each ring, we combine ℓ_2 and ℓ_∞ estimation to ensure low error in ℓ_q and nearly optimal estimation complexity.

• For $q \in [1, 2)$, substantially more delicate analysis is necessary. For large ε, we first, 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 requires using statistical queries in which accuracy takes into account the variance of the random variable (modeled by the VSTAT oracle from Feldman et al. [40]). For small ε, a better upper bound can be obtained via a reduction to the ℓ_2 case.

The nearly tight lower bounds are proved using the technique recently introduced in Feldman et al. [39]. The lower bound also holds for the (potentially simpler) linear optimization problem. We remark that lower bounds on sample complexity do not imply lower bounds on estimation complexity because an SQ algorithm can use many queries.

We summarize the bounds in Table 1 and compare them with those achievable using samples (we provide the proof for these in Appendix B because we are not aware of a good reference for $q \in [1, 2)$).

We then briefly consider the case of general $\mathcal{K}$ with $\mathcal{W} = \mathrm{conv}(\mathcal{K}^*, -\mathcal{K}^*)$ (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 [52] ellipsoid 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 (see Conclusions for a list of additional open problems). This upper bound is nearly tight because, even for $\mathcal{W}$ being the ℓ_1 ball, we give a lower bound of $\tilde{\Omega}(d/\varepsilon^2)$.

1.3. The Gradient Descent Family

The linear case gives us the basis for the study of the traditional setups of convex optimization for Lipschitz functions: nonsmooth, smooth, and strongly convex. In this setting, we assume that, for each w in the support of the distribution D and $x \in \mathcal{K}$, $\|\partial f(x, w)\|_q \le L_0$, and the radius of $\mathcal{K}$ is bounded by R in the ℓ_p norm. The smooth and strongly convex settings correspond to second-order assumptions on F itself. For the first two classes of problems, our algorithms use global approximation of the gradient on $\mathcal{K}$, which, as we know, is already necessary in the linear case. 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.

For the nonsmooth case, we analyze and apply the classic mirror-descent method (Nemirovsky and Yudin [68]); for the smooth case, we rely on the analysis by d'Aspremont [23] of an inexact variant of Nesterov's [69] accelerated method; and for the strongly convex case, we use recent results by Devolder et al. [24] on the inexact dual gradient method. We summarize our results for the ℓ_2 norm in Table 2. Our results for the mirror-descent and Nesterov's [69] 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.

It is important to note that, unlike in the linear case, the SQ algorithms for optimization of general convex functions are adaptive. In other words, the SQs being asked at step t of the iterative algorithm depend on the answers to queries in previous steps. This means that the number of samples that would be necessary to implement such SQ algorithms is no longer easy to determine. In particular, as demonstrated by Dwork et al. [32], the number of samples needed for estimation of adaptive SQs using empirical means might scale linearly with

Table 1. Bounds on ℓ_q mean estimation and linear optimization over ℓ_p ball. Statistical query upper bounds use at most $3d \log d$ (nonadaptive) queries. Lower bounds apply to all algorithms using $\mathrm{poly}(d/\varepsilon)$ queries. Sample complexity is for algorithms with access to i.i.d. samples.

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

Table 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.

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

the query complexity. Although better bounds can be easily achieved in our case (logarithmic as opposed to linear in dimension), they are still worse than the sample complexity. We are not aware of a way to bridge this intriguing gap or prove that it is not possible to answer the SQ queries of these algorithms with the same sample complexity.

Nevertheless, estimation complexity is a key parameter even in the adaptive case. There are many other settings in which one might be interested in implementing answers to SQs, and in some of those, the complexity of the implementation depends on the estimation complexity and query complexity in other ways (for example, differential privacy). In a number of lower bounds for the SQ algorithm (including those in Section 3.2), there is a threshold phenomenon in which, as one goes below a certain estimation complexity, the query complexity lower bound grows from polynomial to exponential very quickly (e.g., Feldman et al. [39, 40]). For such lower bounds, only the estimation complexity matters as long as the query complexity of the algorithm is polynomial.

1.4. Non-Lipschitz Optimization

The estimation complexity bounds obtained for gradient descent–based methods depend polynomially on the Lipschitz constant L_0 and the radius R (unless F is strongly convex). In some cases, such bounds are too large, and we only have a bound on the range of $f(x, w)$ for all $w \in \mathcal{W}$ and $x \in \mathcal{K}$ (note that a bound of $L_0 R$ on the range is also implicit in the Lipschitz setting). This is a natural setting for stochastic optimization (and statistical algorithms in particular) because 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.

For simplicity, we assume $|f(x, w)| \le B = 1$ although our results can be extended to the setting in which only the variance of $f(x, \mathbf{w})$ is bounded by B^2 using the technique from Feldman [37]. Now, for every $x \in \mathcal{K}$, a single SQ for function $f(x, w)$ with tolerance τ gives a value $\widetilde{F}(x)$ such that $|F(x) - \widetilde{F}(x)| \le \tau$. This, as first observed by Valiant [93], gives a τ-approximate value (or zero-order) oracle for $F(x)$. It is proved by Nemirovsky and Yudin [68] and also by Grötschel et al. [45] (who refer to such oracle as a *weak evaluation oracle*) that a 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 $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$. The analysis in Nemirovsky and Yudin [68] and Grötschel et al. [45] is relatively involved and does not provide explicit bounds on τ.

Here, we substantially sharpen the understanding of optimization with the approximate value oracle. Specifically, we show that an (ε/d)-approximate value oracle for $F(x)$ suffices to ε-optimize in polynomial time.

Theorem 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* $\forall x \in \mathcal{K}, |F(x)| \le 1$ *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* $\mathrm{poly}(d, \frac{1}{\varepsilon}, \log(R_1/R_0))$ *using* $\mathrm{poly}(d, \frac{1}{\varepsilon})$ *queries to* $\Omega(\varepsilon/d)$*-approximate value oracle.*

We outline a proof of this theorem, which is based on an extension of the random walk approach of Kalai and Vempala [54] and Lovász and Vempala [64]. This result is also independently obtained in a recent work by Belloni et al. [8], who provide a detailed analysis of the running time and query complexity.

It turns out that the dependence on d in the tolerance parameter of this result cannot be removed altogether: Nemirovsky and Yudin [68] prove that even linear optimization over an ℓ_2 ball of radius one with a τ-approximate value oracle requires $\tau = \tilde{\Omega}(\varepsilon/\sqrt{d})$ for any polynomial-time algorithm. This result also highlights the difference between SQs and the approximate value oracle because the problem can be solved using the SQs of tolerance $\tau = O(\varepsilon)$. Optimization with the value oracle is also substantially more challenging algorithmically.

Luckily, SQs are not constrained to the value information, and we give a substantially simpler and more efficient algorithm for this setting. Our algorithm is based on the classic center-of-gravity method with a crucial new observation: in every iteration, the inertial ellipsoid, whose center is the center of gravity of the current body, can be used to define a (local) norm in which the gradients can be efficiently approximated globally. The exact center of gravity and inertial ellipsoid cannot be found efficiently, and the efficiently implementable ellipsoid method does not have the desired local norm. However, we show that the approximate center-of-gravity method introduced by Bertsimas and Vempala [11] and approximate computation of the inertial ellipsoid (Lovász and Vempala [65]) suffice for our purposes.

Theorem 2 (Informal)**.** *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex body given by a membership oracle $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$ and assume that, for all $w \in \mathcal{W}, x \in \mathcal{K}$, $|f(x,w)| \le 1$. Then, there is a randomized algorithm that, for every distribution D over $\mathcal{W}$, outputs an ε-optimal solution using $O(d^2 \log(1/\varepsilon))$ statistical queries with tolerance $\Omega(\varepsilon/d)$ and runs in $\mathrm{poly}(d, 1/\varepsilon, \log(R_1/R_0))$ time.*

Closing the gap between the tolerance of $\varepsilon/\sqrt{d}$ in the lower bound (already for the linear case) and the tolerance of ε/d in the upper bound is an interesting open problem. Remarkably, as Theorem 1 and the lower bound in Nemirovsky and Yudin [68] show, the same intriguing gap is also present for the approximate value oracle.

1.5. 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 (Valiant [92]), adaptive data analysis (Dwork et al. [32]), and distributed data analysis (Chu et al. [20]).

1.5.1. 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 a 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 that are harder to derive directly.

One inspiring example of this approach comes from using the statistical query algorithm for learning halfspaces (Blum et al. [13]). The structural property on which it relies is linear separability. Combined with the exponential lower bound for learning parities (Kearns [58]), it immediately implies that there is no mapping from $\{-1,1\}^d$ to $\mathbb{R}^N$, which makes parity functions linearly separable for any $N \le N_0 = 2^{\Omega(d)}$. Subsequently, and apparently unaware of this technique, Forster [42] 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 Sherstov [83], the connection between these two results is stated explicitly). His proof relies on a sophisticated and nonalgorithmic technique and is considered a major breakthrough in proving lower bounds on the sign-rank of explicit matrices.

Convex optimization algorithms rely on the existence of convex relaxations for problem instances that (approximately) preserve the value of the solution. Therefore, given an SQ lower bound for a problem, our algorithmic results can be directly translated into lower bounds for convex relaxations of the problem. We now focus on a concrete example that is easily implied by our algorithm and a lower bound for planted constraint satisfaction problems from Feldman et al. [39]. Consider the task of distinguishing a random satisfiable k-SAT formula over n variables of length m from a randomly and uniformly drawn k-SAT formula of length m. This is the refutation problem studied extensively over the past few decades (e.g., Feige [34]). Now, consider the following common approach to the problem: define a convex domain $\mathcal{K}$ and map every k-clause C (or of k distinct variables or their negations) to a convex function f_C over $\mathcal{K}$ scaled to the range $[-1,1]$. Then, given a formula ϕ consisting of clauses $C_1, \ldots, C_m$, find x that minimizes $F_\phi(x) = \frac{1}{m}\sum_i f_{C_i}(x)$, which roughly measures the fraction of unsatisfied clauses. (If f_C's are linear, then one can also maximize $F(x)$, in which case one can also think of the problem as satisfying the largest fraction of clauses.) The goal of such a relaxation is to ensure that, for every satisfiable ϕ, we have that $\min_{x\in\mathcal{K}} F_\phi(x) \le \alpha$ for some fixed α. At the same time, for a randomly chosen ϕ, we want to have, with high probability, $\min_{x\in\mathcal{K}} F_\phi(x) \ge \alpha + \varepsilon$. Ideally, one would hope to get $\varepsilon \approx 2^{-k}$

because, for sufficiently large m, every Boolean assignment leaves at least $\approx 2^{-k}$ fraction of the constraints unsatisfied. But the relaxation can reduce the difference to a smaller value.

We now plug in our algorithm for the ℓ_p/ℓ_q setting to get the following broad class of corollaries.

Corollary 1. *For $p \in \{1,2\}$, let $\mathcal{K} \subseteq \mathcal{B}_p^d$ be a convex body and $\mathcal{F}_p = \{f(\cdot) \mid \forall x \in \mathcal{K}, \|\nabla f(x)\|_q \le 1\}$. Assume that there exists a mapping that maps each k-clause C to a convex function $f_C \in \mathcal{F}_p$. Further assume that, for some $\varepsilon > 0$, if $\phi = C_1, \ldots, C_m$ is satisfiable, then*

$$\min_{x\in\mathcal{K}}\left\{\frac{1}{m}\sum_i f_{C_i}(x)\right\} \le 0.$$

Yet, for the uniform distribution U_k over all the k-clauses,

$$\min_{x\in\mathcal{K}}\left\{\mathop{\mathbf{E}}_{C\sim U_k}[f_C(x)]\right\} > \varepsilon.$$

Then, $d = 2^{\tilde{\Omega}(n\cdot\varepsilon^{2/k})}$.

Note that the second condition is equivalent to applying the relaxation to the formula that includes all the k-clauses. Also for every m, it is implied by the condition

$$\mathop{\mathbf{E}}_{C_1,\ldots,C_m\sim U_k}\left[\min_{x\in\mathcal{K}}\left\{\frac{1}{m}\sum_i f_{C_i}(x)\right\}\right] > \varepsilon.$$

As long as k is a constant and $\varepsilon = \Omega_k(1)$, we get a lower bound of $2^{\Omega(n)}$ on the dimension of any convex relaxation (for which the radius and the Lipschitz constant are at most one). We are not aware of any existing techniques that imply comparable lower bounds. More importantly, our results imply that Corollary 1 extends to a very broad class of general state-of-the-art approaches to stochastic convex optimization.

Current research focuses on the linear case and restricted $\mathcal{K}$s that are obtained through various hierarchies of linear/semidefinite programming (LP/SDP) relaxations or extended formulations (e.g., Schoenebeck [79]). The primary difference between the relaxations used in this line of work and our approach is that our approach only rules out relaxations for which the resulting stochastic convex program can be solved by a statistical algorithm. On the other hand, stochastic convex programs that arise from LP/SDP hierarchies and extended formulations cannot, in general, be solved given the available number of samples (each constraint is a sample). As a result, the use of such relaxations can lead to overfitting, and this is the reason why these relaxations fail. This difference makes our lower bounds incomparable and, in a way, complementary to existing work on lower bounds for specific hierarchies of convex relaxations. For a more detailed discussion of SQ lower bounds, we refer the reader to Feldman et al. [39].

1.5.2. 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. [13] to obtain algorithms tolerant to random classification noise for learning halfspaces and by Blum et al. [12] 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 because 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 (Balcan and Blum [2]). This allows it to be used in place of the support vector machine (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. Shalev-Shwartz and Ben-David [81]). Further details of this application are given in Section 5.2.

1.5.3. 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 are 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 [95] and have more recently been adopted in products by Google and Apple. Although there exists a large and growing literature on mean estimation and convex optimization with (global) differential privacy (e.g., Bassily et al. [6], Chaudhuri et al. [19], Dwork and Roth [30]), 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. [57], 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 (Duchi et al. [27, 28]). 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 on the problem of releasing answers to multiple queries over a single data set. A long line of research has considered this question for *linear* or *counting* queries, which, for a data set $S \subseteq \mathcal{W}^n$ and function $\phi : \mathcal{W} \to [0,1]$, output an estimate of $\frac{1}{n}\sum_{w\in S}\phi(w)$ (see Dwork and Roth [30] 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 (Hardt and Rothblum [48], Roth and Roughgarden [76]) (albeit the running time is linear in $|\mathcal{W}|$). Recently, Ullman [91] considered the question of answering *convex minimization* queries that ask for an approximate minimum of a convex program taking a data point as an input averaged over the data set. 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 [91] 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 the answering of convex minimization queries to the answering of counting queries. As a corollary, we strengthen and substantially generalize the results in Ullman [91].

Details of these applications appear in Sections 5.3 and 5.4.

1.6. Related Work

There is a long history of research on the complexity of convex optimization with access to some type of oracle (e.g., Braun et al. [15], Guzmán and Nemirovski [47], Nemirovsky and Yudin [68]) with a lot of renewed interest because of applications in machine learning (e.g., Agarwal et al. [1], Raginsky and Rakhlin [73]). 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 (d'Aspremont [23], Devolder et al. [25]). Our approach to getting statistical query algorithms for stochastic convex optimization is based on both establishing bridges to that literature and also on improving the state of the art for such oracles in the non-Lipschitz case.

A common way to model stochastic optimization is via a stochastic oracle for the objective function (Nemirovsky and Yudin [68]). Such an 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 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 the 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., Agarwal et al. [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).

SQ access is known to be equivalent (up to polynomial factors) to the setting in which the amount of information extracted from (or communicated about) each sample is limited (Ben-David and Dichterman [9], Feldman et al. [39, 40]). In a recent (and independent) work, Steinhardt et al. [87] establish a number of additional relationships between learning with SQs and learning with several types of restrictions on memory and communication. Among other results, they prove an unexpected upper bound on memory-bounded sparse least-squares regression by giving an SQ algorithm for the problem. Their analysis[2] is related to the one we give for inexact mirror-descent over the ℓ_1 ball. Note that, in optimization over the ℓ_1 ball, the straightforward coordinate-wise ℓ_∞ estimation of gradients suffices. Together with their framework, our results can be easily used to derive low-memory algorithms for other learning problems.

2. Preliminaries

For integer $n \geq 1$, let $[n] \doteq \{1, \ldots, n\}$. Typically, d denotes the ambient space dimension, and n denotes number of samples. Random variables are denoted by bold letters, for example, $\mathbf{w}$, $\mathbf{U}$. 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 ith basis vector in $\mathbb{R}^d$. The $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ notation omits factors polylogarithmic in the argument.

2.1. Convex Bodies and Norms

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 $x \in \mathbb{R}^d$, $\|x\|_p \doteq (\sum_{i \in [d]} |x_i|^p)^{1/p}$. For $R \geq 0$, we denote $\mathcal{B}_p^d(R) = \mathcal{B}^d_{\|\cdot\|_p}(R)$ and similarly, for the unit ball, $\mathcal{B}_p^d = \mathcal{B}_p^d(1)$. 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, when clear from context, we omit the dependence on d.

2.2. Stochastic Optimization

We consider problems of the form

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

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 also assume that $\mathcal{K}$ contains the origin.

2.3. 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 [58] that gives an estimate of the mean with fixed tolerance suffices. We also rely on a stronger oracle from Feldman et al. [40] that takes into the account the variance of the query function and faithfully captures the estimation of the mean of a random variable from samples.

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

Clearly $\mathrm{VSTAT}_D(n)$ is at least as strong as $\mathrm{STAT}_D(1/\sqrt{n})$ (but no stronger than $\mathrm{STAT}_D(1/n)$). Query complexity of a statistical algorithm is the number of queries it uses. The estimation complexity of a statistical query algorithm using $\mathrm{VSTAT}_D(n)$ is the value n, and for an algorithm using $\mathrm{STAT}(\tau)$, it is $n = 1/\tau^2$.

Remark 1. To illustrate the notion of estimation complexity, consider the problem of implementing an SQ oracle using i.i.d. samples from the target distribution. Given samples $\mathbf{w}_1, \ldots, \mathbf{w}_n$, it is natural to use the empirical mean $v = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{w}_i)$ to estimate the true mean $\mu = \mathbb{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w})]$. Let us now see what is the number of samples sufficient for ensuring that the empirical mean satisfies the accuracy guarantees of the oracles defined.

- STAT: If $\phi : \mathcal{W} \to [-1, 1]$, then, by Hoeffding's inequality,

$$\mathop{\mathbf{Pr}}_{(\mathbf{w}_1,\ldots,\mathbf{w}_n)\sim D^n}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{w}_i) - \mu\right| > \tau\right] \le 2\exp\left\{-\frac{n\tau^2}{2}\right\}. \tag{2}$$

This way, choosing $n = 4/\tau^2$, we have that the empirical mean $v = \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{w}_i)$ satisfies $|v - \mu| \le \tau$ with probability 1/2.

- VSTAT: If $\phi : \mathcal{W} \to [0, 1]$, then, by Bernstein's inequality,

$$\mathop{\mathbf{Pr}}_{(\mathbf{w}_1,\ldots,\mathbf{w}_n)\sim D^n}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{w}_i) - \mu\right| > \tau\right] \le \exp\left\{-\frac{n\tau^2/2}{\mu(1-\mu) + \frac{1}{3}\tau}\right\}, \tag{3}$$

where we directly upper bound the variance with the Bhatia–Davis inequality $\mathrm{Var}[\mathbf{w}_1] \le (1-\mu)(\mu - 0)$. Suppose now that we choose the error $\tau = \max\{\frac{1}{n}, \sqrt{\frac{\mu(1-\mu)}{n}}\}$; then, we have two cases:

1. Case $\tau = 1/n$. Then, the probability of deviation is upper bounded by $e^{-3/8}$.
2. Case $\tau = \sqrt{\mu(1-\mu)/n}$. Then, the probability of deviation is upper bounded by

$$\exp\left\{-\frac{1}{2\left[1 + \sqrt{1/\left(3n\mu(1-\mu)\right)}\right]}\right\} \le e^{-1/4},$$

where we have used that $\sqrt{n\mu(1-\mu)} \ge 1$.

Hence, the "estimation complexity" corresponds to the number of i.i.d. samples sufficient to simulate the oracle for a single query with constant success probability. Note that constant probability of success can be amplified to $1 - \delta$ probability of success using n that is $\log(1/\delta)$ times larger than the estimation complexity.

Note that a statistical query algorithm may ask many queries, and it not necessarily true that valid answers to all the queries can be simulated using a number of samples that corresponds to the estimation complexity of a single query. If the m queries asked by the algorithm are nonadaptive (that is, do not depend on answers to previously asked queries), then $O(\log m \cdot n)$ samples suffice to answer all the queries correctly with some positive constant success probability (this follows from applying the union bound to the concentration inequalities discussed in Remark 1). However, if the algorithm queries depend on previous answers, the best known bounds require $O(\sqrt{m} \cdot n)$ samples (see Dwork et al. [32] for a detailed discussion). This also implies that a lower bound on the sample complexity of solving a problem does not directly imply lower bounds on the estimation complexity of an SQ algorithm for the problem.

Whenever that does not make a difference for our upper bounds on estimation complexity, we state the 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 1. *For any function $\phi : \mathcal{W} \to [0, 1]$, input distribution D and condition $A : \mathcal{W} \to \{0, 1\}$ such that $p_A \doteq \mathbf{Pr}_{\mathbf{w}\sim D}[A(\mathbf{w}) = 1] \ge \alpha$ and let $p \doteq \mathbf{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w}) \cdot A(\mathbf{w})]$. Then, query $\phi(w) \cdot A(w)$ to* VSTAT(n/α) *returns a value v such that $|v - p| \le \frac{p_A}{\sqrt{n}}$.*

The value v returned by VSTAT(n/α) on query $\phi(w) \cdot A(w)$ satisfies $|v - p| \le \max\{\frac{\alpha}{n}, \sqrt{\frac{p(1-p)\alpha}{n}}\}$. Note that $p = \mathbf{E}[\phi(\mathbf{w})A(\mathbf{w})] \le \mathbf{Pr}[A(\mathbf{w}) = 1] = p_A$. Hence, $|v - p| \le \frac{p_A}{\sqrt{n}}$.

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

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

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

For normalization purposes, we assume that the random variable $\mathbf{w}$ is supported on $\mathcal{W} = \{w \mid \forall x \in \mathcal{K}, |\langle x, w\rangle| \le 1\}$. Note that $\mathcal{W} = \mathrm{conv}(\mathcal{K}_*, -\mathcal{K}_*)$, 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 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 $\widetilde{w}$ be a vector such that $\|\widetilde{w} - \bar{w}\|_{\mathcal{W}} \le \varepsilon/2$. Let $\widetilde{x} \in \mathcal{K}$ be such that $\langle \widetilde{x}, \widetilde{w}\rangle \le \min_{x\in\mathcal{K}} \langle x, \widetilde{w}\rangle + \xi$. Then, for all $x \in \mathcal{K}$, $F(\widetilde{x}) \le F(x) + \varepsilon + \xi$.

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

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

□

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 $\widetilde{w}$ such that $\|\mathbf{E}_{\mathbf{w}\sim D}[\mathbf{w}] - \widetilde{w}\| \le \varepsilon$. We are concerned primarily with the case in which $\mathcal{W}$ 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 2. *Let $\|\cdot\|_A$ be a norm over $\mathbb{R}^{d_1}$ and $\|\cdot\|_B$ be a norm over $\mathbb{R}^{d_2}$ such that there exists a linear map $T : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and $a, b > 0$ that satisfy $\forall w \in \mathbb{R}^{d_1}$, $a\|Tw\|_B \le \|w\|_A \le b\|Tw\|_B$. 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 a statistical algorithm $\mathcal{A}$ that, for any input distribution supported on $\mathcal{B}_{\|\cdot\|_B}$, computes $\widetilde{z} \in \mathbb{R}^{d_2}$ satisfying $\|\widetilde{z} - \mathbf{E}_{\mathbf{z}}[\mathbf{z}]\|_B \le \frac{a}{2b}\varepsilon$.

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 $\phi'(w) = \phi(aTw)$. Notice that, by our assumption, $\|aTw\|_B \le \|w\|_A \le 1$. Let $\widetilde{y}$ be the output of $\mathcal{A}$ divided by a. By linearity, we have that $\|\widetilde{y} - T\bar{w}\|_B \le \frac{1}{2b}\varepsilon$. Let $\widetilde{w}$ be any vector such that $\|\widetilde{y} - T\widetilde{w}\|_B \le \frac{1}{2b}\varepsilon$. Then,

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

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

Remark 2. The reduction of Lemma 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 $\|z - Tw'\|_B \le \delta$, computing w such that $\|z - Tw\|_B \le \delta + \xi$ 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 (or, equivalently, two norms are within a factor r of each other), 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 because each coordinate has range $[-1, 1]$, and for an estimate $\widetilde{w}$ obtained in this way, we have $\|\widetilde{w} - \bar{w}\|_\infty = \max_i\{|\widetilde{w}_i - \mathbf{E}[\mathbf{w}_i]|\} \le \varepsilon$. This gives the following result.

Theorem 3. *The ℓ_∞ mean estimation problem with error ε can be efficiently solved using d queries to* STAT(ε).

A simple application of Theorem 3 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.[3] (We refer the reader to Hedayat and Wallis [50] for more information on Hadamard matrices.) Then, it is easy to verify that, for every $w \in \mathbb{R}^d$, $\|Hw\|_\infty \le \|w\|_1 \le \sqrt{d}\|Hw\|_\infty$. By Lemma 2, this directly implies the following algorithm:

Theorem 4. *The ℓ_1 mean estimation problem with error ε can be efficiently solved using $2d$ queries to* $\mathrm{STAT}(\varepsilon/\sqrt{2d})$.

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 requires an estimation complexity of $\Omega(d/\varepsilon^2)$. We describe two algorithms for this problem with (nearly) optimal estimation complexity. The first one is a simpler but slightly suboptimal algorithm based on truncating coordinate-wise estimation in a randomly rotated basis. The second algorithm is optimal and relies on the so-called Kashin's representations introduced by Lyubarskii and Vershynin [66].

3.1.1. ℓ_2 Mean Estimation Using a Random Basis. We present a simple to analyze a randomized algorithm that achieves dimension-independent estimation complexity for ℓ_2 mean estimation. The algorithm uses coordinate-wise estimation on a randomly and uniformly chosen basis. We show that, for such a basis, simply truncating coefficients that are too large, with high probability, has 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

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

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

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 one. Let $\mathbf{r} \doteq \mathbf{u} - m_a(\mathbf{u})$. For each i,

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

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

From this lemma, is easy to obtain the following algorithm.

Theorem 5. *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* $\mathrm{STAT}(\Omega(\varepsilon/\log(1/\varepsilon)))$.

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 $\mathrm{STAT}(\varepsilon/[2\sqrt{d}a])$ for query $\phi_{U,i} : \mathcal{W} \to [-1, 1]$ multiplied by a. Now, let $\widetilde{w}_{U,a} \doteq U^{-1}v$ and let $\bar{w}_{U,a} \doteq \mathbf{E}[m_{U,a}(\mathbf{w})]$. This way,

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

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,

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

Notice that, to bound the probability by δ, we may choose $a = \sqrt{2\ln(16/(\delta\varepsilon^2))/d}$. Therefore, the queries require querying $\mathrm{STAT}(\varepsilon/[2\sqrt{2\ln(16/\delta\varepsilon^2)}])$, 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 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 $\widetilde{w}^1, \ldots, \widetilde{w}^k$ as the outputs of the algorithm, we can compute the (high-dimensional) median $\widetilde{w}$, namely, the point $\widetilde{w}^j$ whose median ℓ_2 distance to all the other points is the smallest. It is easy to see that (e.g., Hsu and Sabato [51], Nemirovsky and Yudin [68])

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

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 $\mathrm{STAT}(\Omega(\varepsilon/\log(1/\varepsilon))$ to obtain success probability $1-\delta$. □

3.1.2. 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) (Lyubarskii and Vershynin [66]). 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. Studer et al. [89]). We show that the existence of such a representation leads directly to SQ algorithms for ℓ_2 mean estimation.

We start with some requisite definitions (following Lyubarskii and Vershynin [66]).

Definition 2. A sequence $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ is a *frame*[4] if, for all $w \in \mathbb{R}^d$,

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

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

An easy-to-prove property of a tight frame (see Lyubarskii and Vershynin [66, observation 2.1]) is that, for every frame representation $w = \sum_{j=1}^N a_j u_j$, it holds that $\sum_{j=1}^N a_j^2 \le \|w\|_2^2$.

Definition 3. Consider a sequence $(u_j)_{j=1}^N \subseteq \mathbb{R}^d$ and $w \in \mathbb{R}^d$. An expansion $w = \sum_{j=1}^N a_j u_j$ such that $\|a\|_\infty \le \frac{K}{\sqrt{N}}\|w\|_2$ is referred to as a Kashin's representation of w with level K.

Theorem 6 (Lyubarskii and Vershynin [66])**.** *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 [56] theorem. Lyubarskii and Vershynin [66] 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 aforementioned 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 the u_j's. Other algorithms for the task are discussed in Studer et al. [89].

Theorem 7. *For every d, there is an efficient algorithm that solves the ℓ_2 mean estimation problem (over $\mathcal{B}_2^d$) with error ε using $2d$ queries to* $\mathrm{STAT}(\Omega(\varepsilon))$.

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 6). 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 Lyubarskii and Vershynin [66]). 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 $\widetilde{a}_j$ denote the answer of $\mathrm{STAT}(\varepsilon/K)$ 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 $\mathrm{STAT}(\varepsilon/K)$, we have that $|\bar{a}_j - \widetilde{a}_j| \le \frac{\varepsilon}{\sqrt{N}}$ for every $j \in [N]$. Let $\widetilde{w} \doteq \sum_{j=1}^N \widetilde{a}_j u_j$.

Then, by the property of tight frames,

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

Remark 3 (ℓ_2 Mean Estimation for Unbounded Distributions)**.** In the mean estimation problem, we assume a uniform bound on the norm of vectors in the support of the input distribution D. However, in high-dimensional mean estimation with i.i.d. samples, this assumption is unnecessary, and it suffices to have a bound on the second moment of the norm of the vectors (in fact, the proof used in Appendix B can be easily extended to this situation). For SQs, the same setting and (almost) the same estimation complexity can be obtained using recent results from Feldman [37]. The results show that VSTAT allows estimation of expectation of any unbounded function ϕ of $\mathbf{w}$ within $\varepsilon\sigma$ using $1/\varepsilon^2$ queries of estimation complexity $\tilde{O}(1/\varepsilon^2)$, where σ is the standard deviation of $\phi(\mathbf{w})$. As shown in Feldman [37], this implies that ℓ_2 mean estimation can done with the uniform upper bound replaced by an upper bound on the second moment of the norm.

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 behind 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 nonzero 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 rely on the following simple lemma.

Lemma 4. *For any $x\in\mathbb{R}^d$ and any two $0<p<r$,*

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

Proof. 1.

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

2.

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

Theorem 8. *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 $j=0,\dots,k$, we define

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

and $R_\infty(w)\doteq\sum_{i=1}^{d}e_iw_i\mathbf{1}_{\{|w_i|\le 2^{-(k+1)}\}}$. 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 $\|R_j(w)\|_\infty\le 2^{-j}$, and by Lemma 4, $\|R_j(w)\|_2\le 2^{-(j+1)(1-q/2)}$. 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 $j=0,\dots,k$, we perform the following queries:

• By using $2d$ queries to STAT$(\Omega(\varepsilon'))$, we obtain a vector $\widetilde{w}^{2,j}$ such that $\|\widetilde{w}^{2,j}-\bar{w}^j\|_2\le 2^{(\frac{q}{2}-1)(j+1)}\varepsilon'$. 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 7.

• By using d queries to STAT(ε'), we obtain a vector $\widetilde{w}^{\infty,j}$ such that $\|\widetilde{w}^{\infty,j}-\bar{w}^j\|_\infty\le 2^{-j}\varepsilon'$. For this, notice that $R_j(\mathbf{w})/[2^{-j}]$ is supported on $\mathcal{B}_\infty^d$ and appeals to Theorem 3.

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

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

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

By Lemma 4,

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

Next, we estimate $\bar{w}^\infty$. Because $2^{-(k+1)} = 2^{-\lfloor \ln d/q \rfloor + 1} \le 4d^{-1/q}$, 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 $\widetilde{w}^\infty$ satisfying $\|\widetilde{w}^\infty - \bar{w}^\infty\|_q \le d^{1/q}\|\widetilde{w}^\infty - \bar{w}^\infty\|_\infty \le \varepsilon/2$. Let now $\widetilde{w} = [\sum_{j=0}^k \widetilde{w}^j] + \widetilde{w}^\infty$. We have

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

□

3.1.4. ℓ_q Mean Estimation for $q \in (1,2)$. Finally, we consider the case in which $q \in (1,2)$. Here, we get the nearly optimal estimation complexity via two bounds.

The first bound follows from the simple fact that, for all $w \in \mathbb{R}^d$, $\|w\|_2 \le \|w\|_q \le d^{1/q-1/2}\|w\|_2$. 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 2 with the identity embedding). Using Theorem 7, we then get the following theorem.

Theorem 9. *For $q \in (1,2)$ and every d, there is an efficient algorithm that solves the 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 the VSTAT oracle. (The estimation complexity for STAT is quadratically worse. That still gives an improvement over Theorem 9 for some range of values of ε.) In in the case of $q > 2$, our algorithm decomposes each point into a sum of at most $\log d$ points, each of which has a small dynamic range of nonzero 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 in which the mean of the component itself is close to zero.

Theorem 10. *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 $w^+ \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{w_i \ge 0\}}$, 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 two 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 $k \doteq \lfloor \log(d)/q \rfloor - 2$, $R_j(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{2^{-(j+1)} < |w_i| \le 2^{-j}\}}$, and $R_\infty(w) \doteq \sum_{i=1}^d e_i w_i \mathbf{1}_{\{|w_i| \le 2^{-k-1}\}}$.

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 $n = (\varepsilon'/8)^{-p} \le (16\log(d)/\varepsilon)^p$. Note that $2^j \cdot (R_j(w^+))_i \in [0,1]$ for all w and j. Further, let $v_i = v_i' \cdot \mathbf{1}_{\{|v_i'| \ge 2/n\}}$. We start by proving the following decomposition of the error of v.

Lemma 5. *Let $u \doteq 2^j \cdot \overline{w^{+,j}}$ and $z \doteq u - v$. Then, $\|z\|_q^q \le \|u^<\|_q^q + n^{-q/2} \cdot \|u^>\|_{q/2}^{q/2}$, where $u_i^< = u_i \cdot \mathbf{1}_{\{u_i < 4/n\}}$ and $u_i^> = u_i \cdot \mathbf{1}_{\{u_i \ge 1/n\}}$ 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 because, otherwise (when $u_i \ge 4/n$), by the guarantees of VSTAT(n), we would have $|v_i' - u_i| \le \sqrt{\frac{u_i}{n}}$ and $v_i' \ge u_i - \sqrt{\frac{u_i}{n}} \ge 2/n$. Therefore, in this case, $u_i = u_i^<$ and $z_i = u_i - v_i = u_i^<$.

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

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

We next observe that, by Lemma 4, for every $w \in \mathcal{B}_q^d$,

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

This implies that

$$\|u\|_1 = 2^j \cdot \|\overline{w^{+,j}}\|_1 = 2^j \cdot \left\|\mathbf{E}[R_j(\mathbf{w}^+)]\right\|_1 \le 2^j \cdot \left(2^{-j-1}\right)^{1-q} = 2^{(j+1)q-1}. \tag{4}$$

Now, by Lemma 4 and Equation (4), we have

$$\|u^<\|_q^q \le \left(\frac{4}{n}\right)^{q-1} \cdot \|u^<\|_1 = n^{1-q} \cdot 2^{(j+3)q-3}. \tag{5}$$

Also by Lemma 4 and Equation (4), we have

$$\|u^>\|_{q/2}^{q/2} \le \left(\frac{1}{n}\right)^{q/2-1} \cdot \|u^>\|_1 \le n^{1-q/2} \cdot 2^{(j+1)q-1}. \tag{6}$$

Substituting Equations (5) and (6) into Lemma 5, we get

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

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

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

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 $2^{-k-1} \le 2^{1-\lfloor \log(d)/q \rfloor} \le 4d^{-1/q}$. 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 $\widetilde{w}^\infty$ obtained in this way satisfies $\|\bar{w}^\infty - \widetilde{w}^\infty\|_q \le \varepsilon'$.

Now, let $\widetilde{w} = \sum_{j=0}^k (\widetilde{w}^{+,j} - \widetilde{w}^{-,j}) + \widetilde{w}^\infty$. Each of the estimates has an 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 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 2.

Corollary 2. *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 2 and Theorem 7, we have the following.

Proposition 1. *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 1, for an arbitrary convex body $\mathcal{K}$, the stochastic linear optimization problem over $\mathcal{K}$ reduces to a 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 is proved in Corollary 4.

A drawback of this approach is that it depends on knowledge of the John ellipsoid for $\mathcal{W}$, which, in general, cannot be computed efficiently (e.g., Ben-Tal and Nemirovski [10]). 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 (Khachiyan [59]). From this, we conclude the following.

Corollary 3. *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 polynomial time 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 Feldman et al. [39] that is based on bounding the statistical dimension with discrimination norm. The *discrimination norm* of a finite 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 $\|h\|_D = \sqrt{\mathbf{E}_D[h^2]}$ and $D' \sim \mathcal{D}'$ refers to choosing D' randomly and uniformly from the set $\mathcal{D}'$.

Let X be the domain in which all distributions of interest are supported. 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 one if $D' \in \mathcal{D}$ and zero if $D' = D$.

Definition 4 (Feldman et al. [40]). 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 is 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 11 (Feldman et al. [40]). *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 $\mathrm{VSTAT}(1/(3 \cdot \kappa^2))$.*

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 and 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 to distinguish every distribution in $\mathcal{D}_\alpha$ from D—in other words, to solve the decision problem $\mathcal{B}(\mathcal{D}_\alpha, D)$.

Lemma 6. *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 $2/3$.*

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

$$\mathop{\mathbf{E}}_{D_\alpha v}[h] - \mathop{\mathbf{E}}_{D}[h] = \frac{\alpha}{2d} \sum_{i\in[d]} v_i \cdot (h(e_i) - h(-e_i)). \tag{7}$$

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

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

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

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

From here, a simple manipulation (see Shalev-Shwartz and Ben-David [81, Lemma A.4]) implies that

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

Note that

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

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 Equation (7), we have that

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

By Definition 4, this implies that, for every $t > 0$, $\mathrm{SDN}(\mathcal{B}(\mathcal{D}_\alpha, \mathrm{D}), \alpha\sqrt{\log \mathrm{t}/\mathrm{d}}) \ge \mathrm{t}$. By Theorem 11, for any $r > 0$, $2^{\Omega(r)}$ queries to $\mathrm{VSTAT}(d/(r\alpha^2))$ 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 4. *For any $\varepsilon \le d^{1/q-1}/2$ 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 $\varepsilon \ge d^{1/q-1}/2 = d^{-1/p}/2$. 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 $\varepsilon \ge d^{-1/p}/2$, let $d' = (2\varepsilon)^{-p}$ (ignoring for simplicity the minor issues with rounding). Now, Corollary 4 applied to d' implies that $2^{\Omega(r)}$ queries to $\mathrm{VSTAT}((d')^{2/q-1}/(r\varepsilon^2))$ 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 5. *For any $q > 1$, $\varepsilon \ge d^{1/q-1}/2$, 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 $\mathrm{VSTAT}(n)$ 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 (Feldman et al. [40].

4. The Gradient Descent Family

We now describe approaches for solving convex programs by SQ algorithms that are based on the broad literature of inexact gradient methods. We 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 $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E}_{\mathbf{w}}[f(x,\mathbf{w})]\}$, 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;[5] 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 $x \in \mathcal{K}$, $|f(x)| \le B$.

• $\mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$ of L_0-Lipschitz continuous functions with respect to (w.r.t.) $\|\cdot\|$ if, for all $x, y \in \mathcal{K}$, $|f(x) - f(y)| \le L_0\|x - y\|$; this implies

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + L_0\|y - x\|. \tag{8}$$

• $\mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$ of functions with L_1-Lipschitz continuous gradient w.r.t. $\|\cdot\|$ if, for all $x, y \in \mathcal{K}$, $\|\nabla f(x) - \nabla f(y)\|_* \le L_1\|x - y\|$; this implies

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L_1}{2}\|y - x\|^2. \tag{9}$$

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

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\kappa}{2}\|y - x\|^2. \tag{10}$$

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 5 (Global Approximate Gradient, d'Aspremont [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 $u, x, y \in \mathcal{K}$,

$$|\langle \tilde{g}(x) - \nabla F(x), y - u\rangle| \le \eta. \tag{11}$$

Observation 2. Let $\|\cdot\|$ be a norm such that $\mathcal{K} \subseteq \mathcal{B}_{\|\cdot\|}(0, R)$ for some $R > 0$. If $F(x) = \mathbf{E}_{\mathbf{w}}[f(x, \mathbf{w})]$ satisfies for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0)$, then implementing an η-approximate gradient oracle reduces to a mean estimation problem in $\|\cdot\|_*$ with error $\eta/[2RL_0]$. Indeed, we consider the estimation problem of a random linear function $\nabla f(x, \mathbf{w})/L_0$, which is a random variable supported in $\mathcal{B}_{\|\cdot\|_*}$, and if we solve this problem with error $\eta/[2RL_0]$,

$$|\langle \nabla F(x) - \tilde{g}(x), y - u\rangle| \le \|\nabla F(x) - \tilde{g}(x)\|_*\|y - u\| \le \|\nabla F(x) - \tilde{g}(x)\|_*(2R) \le \eta.$$

Definition 6 (Inexact Oracle, Devolder et al. [24, 25])**.** 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 $x, y \in \mathcal{K}$,

$$\frac{\mu}{2}\|y - x\|^2 \le F(y) - \left[\tilde{F}(x) - \langle \tilde{g}(x), y - x\rangle\right] \le \frac{M}{2}\|y - x\|^2 + \eta. \tag{12}$$

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

Lemma 7. *Let $\eta > 0$, $0 < \kappa \le L_1$ 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 $M = 2L_0^2/\eta$.

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 $|\hat{F}(x) - F(x)| \le \eta/4$ and then use as approximation

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

This way, $|F(x) - \tilde{F}(x)| \le |F(x) - \hat{F}(x)| + |\hat{F}(x) - \tilde{F}(x)| \le 3\eta/4$, and also $F(x) - \tilde{F}(x) = F(x) - \hat{F}(x) + \eta/2 \ge \eta/4$. 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

$$\|\nabla F(x) - \tilde{g}(x)\|_* \le \sqrt{\eta\kappa}/2 \le \sqrt{\eta L_1}/2, \tag{13}$$

suffices for a (η, μ, M)-oracle, where $\mu = \kappa/2$, $M = 2L_1$. For convenience, we refer to the first inequality in (12) as the *lower bound* and the second as the *upper bound*.

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

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

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

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

In order to prove this inequality, notice that, among all y's such that $\|y - x\| = t$, the minimum of the expression 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

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

whose minimum is attained at $t = \dfrac{\|\nabla F(x) - \tilde{g}(x)\|_*}{\mu}$ 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.

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

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

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

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

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

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

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

which is precisely the upper bound.

As a conclusion, we prove that, in order to obtain $\tilde{g}$ for a (η, M, μ)-oracle, it suffices to obtain an approximate gradient satisfying (13). Obtaining $\tilde{g}$ reduces to a mean estimation problem in $\|\cdot\|_*$ with error $\sqrt{\eta\kappa}/[2L_0]$; more precisely, we consider the mean estimation problem for the random linear function $\nabla f(x, \mathbf{w})/L_0$ (whose distribution is supported in $\mathcal{B}_{\|\cdot\|_*}$) with accuracy $\sqrt{\eta\kappa}/[2L_0]$. 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 (8), we can prove that, for all $x, y \in \mathcal{K}$,

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

and thus, we can use $M = 2L_0^2/\eta$. This is sufficient for carrying out the proof, 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 see that, in terms of estimation complexity, there is no significant gain from the nonsmooth 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 7). 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 almost surely—for example, in regularized linear regression—as long as the expectation is strongly convex.

4.2.1. Nonsmooth 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 (in which $2 \le r < \infty$) $\Psi : \mathcal{K} \to \mathbb{R}_+$ w.r.t. the norm $\|\cdot\|$, that is, that satisfies,[6] for all $x, y \in \mathcal{K}$,

$$\Psi(y) \ge \Psi(x) + \langle \nabla\Psi(x), y - x\rangle + \frac{1}{r}\|y - x\|^r. \tag{14}$$

We assume without loss of generality that $\inf_{x\in\mathcal{K}} \Psi(x) = 0$.

The existence of r-uniformly convex functions holds in rather general situations (Pisier [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 $x \in \operatorname{int}(\mathcal{K})$ 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 (14), we have $V_x(y) \ge \frac{1}{r}\|y - x\|^r$.

For the first-order methods in this section, we 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 problem can be efficiently solved in more general situations (Nemirovsky and Yudin [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 (Nemirovsky and Yudin [68]). This is a classic method for minimization of nonsmooth 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 an 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 computes its next query point as

$$x^{t+1} = \arg\min_{y\in\mathcal{K}} \{\alpha\langle \tilde{g}^t, y - x^t\rangle + V_{x^t}(y)\}, \tag{15}$$

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

Proposition 2. *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*

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

Proof. We first state without proof the following identity for prox-functions (for example, see Ben-Tal and Nemirovski [10, 5.3.20]): for all x, $x\prime$, 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 (15) are

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

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

$$\begin{aligned}
\alpha[F(x^t) - F(u)] &\le \alpha\left\langle \nabla F(x^t), x^t - u \right\rangle \\
&\le \alpha\left\langle \tilde{g}^t, x^t - u \right\rangle + \alpha\eta \\
&= \alpha\left\langle \tilde{g}^t, x^t - x^{t+1} \right\rangle + \alpha\left\langle \tilde{g}^t, x^{t+1} - u \right\rangle + \alpha\eta \\
&\le \alpha\left\langle \tilde{g}^t, x^t - x^{t+1} \right\rangle - \left\langle \nabla V_{x^t}\left(x^{t+1}\right), x^{t+1} - u \right\rangle + \alpha\eta \\
&= \alpha\left\langle \tilde{g}^t, x^t - x^{t+1} \right\rangle + V_{x^t}(u) - V_{x^{t+1}}(u) - V_{x^t}\left(x^{t+1}\right) + \alpha\eta \\
&\le \left[\alpha\left\langle \tilde{g}^t, x^t - x^{t+1} \right\rangle - \frac{1}{r}\left|x^t - x^{t+1}\right|^r\right] + V_{x^t}(u) - V_{x^{t+1}}(u) + \alpha\eta \\
&\le \frac{1}{s}\left|\alpha\tilde{g}^t\right|_*^s + V_{x^t}(u) - V_{x^{t+1}}(u) + \alpha\eta,
\end{aligned}$$

where we have used all the observations, 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,

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

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

We can readily apply the result to stochastic convex programs in nonsmooth ℓ_p settings.

Definition 7 (ℓ_p-Setup)**.** Let $1 \le p \le \infty$, $L_0, R > 0$, and $\mathcal{K} \subseteq \mathcal{B}_p^d(R)$ be a convex body. We define as the (nonsmooth) 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 (15) is solvable for simple enough $\mathcal{K}$). This result, in combination with Lemma 7 and Theorem 2 gives precise estimation complexity results that are summarized in the following corollary and proved in Appendix C.

Corollary 6. *The stochastic optimization problem in the nonsmooth ℓ_p-setup can be solved with accuracy ε by*

- *If $p = 1$, using $O(d \log d \cdot (\frac{L_0 R}{\varepsilon})^2)$ queries to $STAT(\frac{\varepsilon}{4L_0R})$.*
- *If $1 < p < 2$, using $O(d \log d \cdot \frac{1}{(p-1)}(\frac{L_0 R}{\varepsilon})^2)$ queries to $STAT(\Omega(\frac{\varepsilon}{[\log d]L_0 R}))$.*
- *If $p = 2$, using $O(d \cdot (\frac{L_0 R}{\varepsilon})^2)$ queries to $STAT(\Omega(\frac{\varepsilon}{L_0 R}))$.*
- *If $2 < p < \infty$, using $O(d \log d \cdot 4^p(\frac{L_0 R}{\varepsilon})^p)$ queries to $VSTAT((\frac{64 L_0 R \log d}{\varepsilon})^p)$.*

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 restrict the analysis to the case in which the prox-function is obtained from a strongly convex function, that is r-uniform convexity with $r = 2$. We utilize a known inexact variant of Nesterov's [69] accelerated method.

Proposition 3 (d'Aspremont [23])**.** *Let $F \in \mathcal{F}^1_{\|\cdot\|}(\mathcal{K}, L_1)$ and let $\Psi : \mathcal{K} \to \mathbb{R}_+$ be a one-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 in which the algorithm has access to an η-approximate gradient oracle for F over $\mathcal{K}$. Then,*

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

The consequences for the smooth ℓ_p-setup, which are straightforward from the theorem and Observation 2, are summarized as follows and proved in Appendix D.

Corollary 7. *Any stochastic convex optimization problem in the smooth ℓ_p-setup can be solved with accuracy ε by*

- *If $p = 1$, using $O(d\sqrt{\log d} \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}})$ queries to $STAT(\frac{\varepsilon}{12L_0R})$.*
- *If $1 < p < 2$, using $O(d \log d \cdot \frac{1}{\sqrt{p-1}}\sqrt{\frac{L_1 R^2}{\varepsilon}})$ queries to $STAT(\Omega(\frac{\varepsilon}{[\log d]L_0 R}))$.*
- *If $p = 2$, using $O(d \cdot \sqrt{\frac{L_1 R^2}{\varepsilon}})$ queries to $STAT(\Omega(\frac{\varepsilon}{L_0 R}))$.*

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, that is, $\|\cdot\| = \|\cdot\|_2$. There are two main advantages of having a strongly convex objective: on the one hand, gradient methods in this case achieve a linear convergence rate; on the other hand, we 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 (Devolder et al. [24]), achieving a linear convergence rate. The result is as follows:

Theorem 12 (Devolder et al. [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,*

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

The results in Devolder et al. [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 7 leads to the following result.

Corollary 8. *The stochastic convex optimization problem $\min_{x\in\mathcal{K}}\{F(x) \doteq \mathbf{E}_{\mathbf{w}}[f(x,w)]\}$, 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 $O(d\cdot\frac{L_1}{\kappa}\log(\frac{L_1R}{\varepsilon}))$ 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 $O(d\cdot\frac{L_0^2}{\varepsilon\kappa}\log(\frac{L_0R}{\varepsilon}))$ queries to $STAT(\Omega(\sqrt{\varepsilon\kappa}/L_0))$.

4.3. Applications to Generalized Linear Regression

We provide 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, for example, 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, in which $\ell(z',z) = (z'-z)^2$.

4.3.1. Nonsmooth 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, that is, $\|\cdot\| = \|\cdot\|_p$. Hence, the Lipschitz constant of our stochastic objective $f(\cdot,(w,z)) = \ell(\langle w,\cdot\rangle,z)$ can be upper bounded as $L_0 \le L_{\ell,0}\cdot W$. For this setting, Kakade et al. [53] show that the sample complexity of achieving excess error $\varepsilon > 0$ with constant success probability is $n = O((\frac{L_{\ell,0}WR}{\varepsilon})^2 \ln d)$ when $p = 1$, and $n = O((\frac{L_{\ell,0}WR}{\varepsilon})^2(q-1))$ for $1 < p \le 2$. Using Corollary 6, 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. [53] do not provide sample complexity bounds for $p > 2$; however, because 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.

4.3.2. 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 one-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 nonsmooth linear regression in ℓ_2-setup (as described), Shalev-Shwartz et al. [82] provide a sample complexity bound of $O(\frac{(L_{\ell,0}W)^2}{\lambda\varepsilon})$ (with constant success probability). Note that the expected objective is 2λ-strongly convex, and therefore, applying Corollary 8, we get the same (up to constant factors) bounds on estimation complexity of solving this problem by SQ algorithms.

5. Applications

In this section, we describe several applications of our results. We start by showing that our algorithms, together with lower bounds for SQ algorithms, give lower bounds against convex programs. We then give several easy examples of using upper bounds in other contexts: (1) new SQ implementation of algorithms for learning halfspaces that eliminate the linear dependence on the dimension in previous work, (2) algorithms for high-dimensional mean estimation with local differential privacy that rederive and generalize existing bounds (we also give the first algorithm for solving general stochastic convex programs with local differential privacy), and (3) strengthening and generalization of algorithms for answering sequences of convex minimization queries differentially privately given in Ullman [91].

Additional applications in settings in which 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 Bassily et al. [7] and Dwork et al. [32, 33]) can be used to solve a sequence of adaptively chosen stochastic convex minimization problems. This question has been recently studied by Bassily et al. [7], and our bounds can be easily seen to strengthen and generalize some of their results (see Section 5.4 for an analogous comparison).

5.1. Lower Bounds

We 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 $(v_i)_{i=1}^m$ from some collection of functions V over a set of "solutions" Z, and the goal is to (approximately) minimize or maximize $\frac{1}{m}\sum_{i\in[m]} v_i(z)$ for $z\in Z$. Here, either Z is nonconvex or functions in V are nonconvex (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, $z\in\{0,1\}^n$ represents a subset of vertices and V consists of $\binom{n}{2}$ "$z_i\neq z_j$" predicates.

A standard approach to such nonconvex problems is to map Z to a convex body $\mathcal{K}\subseteq\mathbb{R}^d$ 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 $(v_i)_{i=1}^m$,

$$\min_{z\in Z}\frac{1}{m}\sum_{i\in[m]} v_i(z) - \min_{x\in\mathcal{K}}\frac{1}{m}\sum_{i\in[m]}(T(v_i))(x) < \varepsilon. \tag{16}$$

(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 is not essential for our discussion.) The goal of lower bounds against such approaches is to show that specific mappings (or classes of mappings) do not allow solving the original problem via this approach, for example, have a large integrality gap.

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

Definition 8. For an SQ oracle $\mathcal{O}$, $t>0$, and a problem P over distributions, we say that $P\in\mathrm{Stat}(\mathcal{O},t)$ 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 $\mathrm{Opt}(\mathcal{K},\mathcal{F},\varepsilon)$ the problem of finding, for every distribution D over $\mathcal{F}$, x^* such that $F(x^*)\le\min_{x\in\mathcal{K}}F(x)+\varepsilon$, where $F(x)\doteq\mathbf{E}_{f\sim D}[f(x)]$.

For simplicity, let's focus on the decision problem[7] 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 13. *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 $D \in \mathcal{D}^+$, $\min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)] > \alpha$, and for all $D \in \mathcal{D}^-$, $\min_{x\in\mathcal{K}} \mathbf{E}_{v\sim D}[(T(v))(x)] \le 0$. 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 $Opt(\mathcal{K}, \mathcal{F}, \alpha/2) \notin Stat(\mathcal{O}, t)$.*

The conclusion of this theorem, namely $\mathrm{Opt}(\mathcal{K}, \mathcal{F}, \alpha/2) \notin \mathrm{Stat}(\mathcal{O}, t)$, 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 in some specific way, for example, using an SQ oracle efficiently.

Note that the assumptions on the mapping in Theorem 13 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 Equation (16). However, for a sufficiently large number of samples m, for every x, the average over random samples $\frac{1}{m}\sum_{i\in[m]}(T(v_i))(x)$ is close to the expectation $\mathbf{E}_{v\sim D}[(T(v))(x)]$. Therefore, the condition can be equivalently reformulated in terms of the average over a sufficiently large number of samples drawn i.i.d. from D.

5.1.1. Lower Bounds for Planted CSPs. We now describe an instantiation of this approach using lower bounds for constraint satisfaction problems established in Feldman et al. [39]. Feldman et al. [39] describe implications of their lower bounds for convex relaxations using results for more general (non-Lipschitz) stochastic convex optimization and discuss their relationship to those for lift-and-project hierarchies (Sherali-Adams, Lovász-Schrijver, Lasserre) of canonical LP/SDP formulations. Here, we give examples of implications of our results for the Lipschitz case.

Let $Z = \{-1, 1\}^n$ be the set of assignments to n 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., Coja-Oghlan et al. [21]) and the pseudorandom generator based on Goldreich's [44] proposal for one-way functions. Associated with every family created in this way is a complexity parameter r which, as shown in Feldman et al. [39], 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 Feldman et al. [39] can be (somewhat informally) restated as follows.

Theorem 14 (Feldman et al. [39])**.** *Let $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^n}$ 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 $VSTAT(\frac{n^r}{(\log t)^r})$ for any $t \ge 1$.*

Combining this with Theorem 13, we get the following general statement:

Theorem 15. *Let $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^n}$ 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 d-dimensional set $\mathcal{K}$ such that, for all $z \in \{-1, 1\}^n$, $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim D_z}[f_C(x)] \le 0$ and $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim U_k}[f_C(x)] > \alpha$. Then, for every $t \ge 1$, $Opt(\mathcal{K}, \mathcal{F}, \alpha/2) \notin Stat(VSTAT(\frac{n^r}{(\log t)^r}),\ (t))$.*

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 $\mathcal{F}^0_{\|\cdot\|_p}(\mathcal{B}^d_p, 1)$ (see Section 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 9. *There exists a family of distributions $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^n}$ 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 $T : \mathcal{C}_k \to \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{B}^d_p, 1)$ such that, for all z, $\min_{x\in\mathcal{B}^d_p} \mathbf{E}_{C\sim D_z}[(T(C))(x)] \le 0$ and $\min_{x\in\mathcal{B}^d_p} \mathbf{E}_{C\sim U_k}[(T(C))(x)] > \varepsilon$, then $d = 2^{\tilde{\Omega}(n^{k/(k+2)}\cdot\varepsilon^{2/(k+2)})}$. For $p \in \{1,2\}$, we get a slightly stronger lower bound of $d = 2^{\tilde{\Omega}(nI\cdot\varepsilon^{2/k})}$.*

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

We note that the problem described in Corollary 9 is easier than the distributional k-SAT refutation problem, in which $\mathcal{D}$ contains all distributions with satisfiable support. Therefore, the assumptions of Corollary 1 that we stated in the introduction imply the assumptions of Corollary 9.

Similarly, we can use the results of Section 5 to obtain the following lower bound on the dimension of any convex relaxation:

Corollary 10. *There exists a family of distributions $\mathcal{D} = \{D_z\}_{z\in\{-1,1\}^n}$ over $\mathcal{C}_k$ such that the support of D_z is satisfied by z with the following property: for every convex body $\mathcal{K} \subseteq \mathbb{R}^d$, if there exists a mapping $T : \mathcal{C}_k \to \mathcal{F}(\mathcal{K}, 1)$ such that, for all z, $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim D_z}[(T(C))(x)] \le 0$ and $\min_{x\in\mathcal{K}} \mathbf{E}_{C\sim U_k}[(T(C))(x)] > \varepsilon$, then $d = \tilde{\Omega}(n^{k/2}\cdot\varepsilon)$.*

5.2. 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 $f_{w,\theta}(x) \doteq \mathtt{sign}(\langle w, x\rangle - \theta)$.

5.2.1. Margin Perceptron. We start with the classic Perceptron algorithm (Novikoff [70], Rosenblatt [75]). 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 (Balcan and Blum [2]). The margin Perceptron with parameter η works as follows. Initialize the weights $w^0 = 0^d$. At around $t \ge 1$, given a vector x^t and correct prediction $y^t \in \{-1,1\}$, if $y^t \cdot \langle w^{t-1}, x^t\rangle \ge \eta$, 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 16 (Balcan and Blum [2])**.** *Let $(x^1, y^1), \ldots, (x^t, y^t)$ be any sequence of examples in $\mathcal{B}^d_2(R) \times \{-1,1\}$ and assume that there exists a vector $w^* \in \mathcal{B}^d_2(W)$ such that, for all t, $y^t\langle w^*, x^t\rangle \ge \gamma > 0$. Let M be the number of rounds in which the margin Perceptron with parameter η updates the weights on this sequence of examples. Then, $M \le R^2W^2/(\gamma - \eta)^2$.*

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 Perceptron, which does not guarantee any margin). For example, by choosing $\eta = \gamma/2$, one can approximately maximize the margin while only paying a factor four in the upper bound on the number of updates. This means that the halfspace produced by the 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 $\mathcal{B}^d_2(R) \times \{-1,1\}$. The distribution is assumed to be supported only on examples (x, y) that, for some vector w^* satisfy $y\langle w^*, x\rangle \ge \gamma$. 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 (Blum et al. [13], Bylander [17]). Namely, update using the example $(\bar{x}^t, 1)$, where $\bar{x}^t = \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\cdot\mathbf{x} \,|\, \mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta]$. Naturally, by linearity of the expectation, we have that $\langle w^{t-1}, \bar{x}^t\rangle < \eta$ and $\langle w^*, \bar{x}^t\rangle \ge \gamma$ and also, by convexity, that $\bar{x}^t \in \mathcal{B}^d_2(R)$. 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 $\tilde{x}$ that satisfies these conditions (at least approximately). The main difficulty here is

preserving the condition $\langle w^*, \tilde{x}\rangle \geq \gamma$ because 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 Theorem 16 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,

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

In PAC learning with error ε, we can assume that $\alpha \doteq \mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbf{y}\langle w^{t-1}, \mathbf{x}\rangle < \eta] \geq \varepsilon$ because, 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 7, we can get the desired estimate using $2d$ queries to $\mathrm{STAT}(\Omega(\varepsilon\gamma/(RW)))$. 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 1, 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 17. *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 $\mathrm{VSTAT}(O((WR/\gamma)^2/\varepsilon))$.*

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., Cesa-Bianchi et al. [18]). SQ implementations of Perceptron were used to establish learnability of large-margin halfspaces with random classification noise (Bylander [17]) and to give a private version of Perceptron (Blum et al. [12]). Perceptron is also the basis of SQ algorithms for learning halfspaces that do not require a margin assumption (Blum et al. [13], Dunagan and Vempala [29]). All previous analyses that we are aware of used coordinate-wise estimation of $\bar{x}$ and resulted in an 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.

5.2.2. Online p-Norm Algorithms. The Perceptron algorithm can be seen as a member in the family of online p-norm algorithms (Grove et al. [46]) with $p = 2$. The other famous member of this family is the Winnow algorithm (Littlestone [63]), which corresponds to $p = \infty$. For $p \in [2, \infty]$, a p-norm algorithm is based on a 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 17.

Theorem 18. *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 $\mathrm{VSTAT}(O(\log d(WR/\gamma)^2/\varepsilon))$, and for $p = \infty$, p-norm-SQ uses $O(d \log d(WR/\gamma)^2)$ queries to $\mathrm{VSTAT}(O((WR/\gamma)^2/\varepsilon))$.*

It is not hard to prove that a 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., Dasgupta et al. [22], Freund and Schapire [43], Servedio [80]). The analysis

inevitably relies on a margin assumption (and its relaxations) and, hence, we believe, can be implemented using SQs in a similar manner.

5.3. Local Differential Privacy

We now exploit the simulation of SQ algorithms by locally differentially private (LDP) algorithms (Kasiviswanathan et al. [57]) 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 9. An α-local randomizer $R : \mathcal{W} \to \mathcal{Z}$ is a randomized algorithm that satisfies $\forall w \in \mathcal{W}$ and $z_1, z_2 \in \mathcal{Z}$, $\mathbf{Pr}[R(w) = z_1] \le e^{\alpha}\mathbf{Pr}[R(w) = z_2]$. An $\mathrm{LR_D}$ 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 the $\mathrm{LR_D}$ oracle. Further, if the algorithm uses n samples such that sample i is obtained from α_i-randomizer R_i, then $\sum_{i\in[n]} \alpha_i \le \alpha$.

The composition properties of differential privacy imply that an α-local algorithm is α-differentially private (Dwork et al. [31]).

Kasiviswanathan et al. [57] show that one can simulate a $\mathrm{STAT}_D(\tau)$ oracle with success probability $1 - \delta$ by an α-local algorithm using $n = O(\log(1/\delta)/(\alpha\tau)^2)$ samples from the $\mathbf{LR}_D$ oracle. This has the following implication for simulating SQ algorithms.

Theorem 19 (Kasiviswanathan et al. [57])**.** *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 the $\mathbf{LR}_D$ 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. [57] also prove a converse of this theorem that uses n queries to $\mathrm{STAT}(\Theta(e^{2\alpha}\delta/n))$ 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.

5.3.1. Mean Estimation.

Duchi et al. [27] give α-local algorithms for ℓ_2 mean estimation using $O(d/(\varepsilon\alpha)^2)$ sample 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 Theorem 19 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 11. *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 $\mathbf{LR}_D$, where*

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

5.3.2. Convex Optimization.

Duchi et al. [28] give locally private versions of the mirror-descent algorithm for the ℓ_1-setup and gradient descent for the ℓ_2-setup. Their algorithms achieve the guarantees of the (nonprivate) 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 Section 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 the ℓ_1-setup, we obtain an $O(d\log d/\alpha^2 \cdot (RW/\varepsilon)^4)$ bound. At the same time, our results in Sections 4 and 5 are substantially more general. In particular, our center-of-gravity-based algorithm (Theorem 15) gives the first α-local algorithm for a non-Lipschitz setting.

Corollary 12. *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 $\mathrm{poly}(d, \frac{B}{\alpha\varepsilon}, \log(R_1/R_0))$ and using $n = \tilde{O}(d^4B^2/(\varepsilon^2\alpha^2))$ samples from $\mathbf{LR}_D$.*

We note that a closely related application is also discussed in Belloni et al. [8]. It relies on the random walk-based approximate value oracle optimization algorithm similar to the one we outlined in Section 5.1. Known optimization algorithms that use only the approximate value oracle require a substantially larger number of

queries than our algorithm in Theorem 15 and, hence, need a substantially larger number of samples to implement. (Specifically, for the setting in Corollary 12, $n = \tilde{O}(d^{6.5}B^2/(\varepsilon^2\alpha^2))$ is implied by the algorithm given in Belloni et al. [8].)

5.4. Differentially Private Answering of Convex Minimization Queries

An additional implication in the context of differentially private data analysis is for the problem of releasing answers to convex minimization queries over a single data set that was recently studied by Ullman [91]. For a data set $S = (w^i)_{i=1}^n \in \mathcal{W}^n$, 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 $q_f(S) \doteq \operatorname{argmin}_{x\in\mathcal{K}} \frac{1}{n}\sum_{i\in[n]} f(x, w^i)$. Ullman [91] considers the question of how to answer sequences of such queries ε-approximately (that is, by a point $\tilde{x}$ such that $\frac{1}{n}\sum_{i\in[n]} f(\tilde{x}, w^i) \le q_f(S) + \varepsilon$). 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 $|v - \frac{1}{n}\sum_{i\in[n]} \phi(w^i)| \le \tau$. A long line of research in differential privacy has considered the question of answering counting queries (see Dwork and Roth [30] for an overview). In particular, Hardt and Rothblum [48] prove that, given a data set of size $n \ge n_0 = O(\sqrt{\log(|\mathcal{W}|)\log(1/\beta)}\cdot \log t/(\alpha\tau^2))$, it is possible to (α, β)-differentially privately answer any sequence of t counting queries with accuracy τ (and success probability $\ge 2/3$). 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 Hardt and Rothblum [48] depends only logarithmically on t. As a result, the additional price for such implementation is relatively small because such algorithms are usually considered in the setting in which t is large and $\log|\mathcal{W}| = \Theta(d)$. Hence, the counting query algorithm in Hardt and Rothblum [48] together with the results in Corollary 6 immediately imply an algorithm for answering such queries that strengthens quantitatively and generalizes the results in Ullman [91].

Corollary 13. *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 data sets of size n for*

$$n \ge n_0 = \tilde{O}\left(\frac{(L_0R)^2\sqrt{\log(|\mathcal{W}|)}\cdot\log t}{\varepsilon^2\alpha}\cdot \operatorname{polylog}\left(\frac{d}{\beta\delta}\right)\right).$$

For comparison, the results in Ullman [91] only consider the $p = 2$ case, and the stated upper bound is

$$n \ge n_0 = \tilde{O}\left(\frac{(L_0R)^2\sqrt{\log(|\mathcal{W}|)}\cdot\max\{\log t, \sqrt{d}\}}{\varepsilon^2\alpha}\cdot \operatorname{polylog}\left(\frac{1}{\beta\delta}\right)\right).$$

Our bound is a significant generalization and an improvement by a factor of at least $\tilde{O}(\sqrt{d}/\log t)$. Ullman [91] also shows that, for generalized linear regression, one can replace the $\sqrt{d}$ in the maximum by L_0R/ε. The bound in Corollary 13 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 8 into the algorithm in Hardt and Rothblum [48], we obtain that it suffices to use a data set of size

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

The bound obtained by Ullman [91] for the same function class is

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

Here, our improvement over Ullman [91] is two-fold: we eliminate the $\sqrt{d}$ factor, and we essentially eliminate the dependence on R (as in the nonprivate setting). We remark that our bound might appear incomparable to that in Ullman [91], but it is, in fact, stronger because it can be assumed that $\kappa \geq \varepsilon/R^2$ (otherwise, bounds that do not rely on strong convexity are better).

6. Conclusions

In this work, we give the first treatment of two basic problems in the SQ query model: high-dimensional mean estimation and stochastic convex optimization. In the process, we demonstrate new connections of our questions to concepts and tools from convex geometry, optimization with approximate oracles, and compressed sensing. Our results provide detailed (but by no means exhaustive) answers to some of the most basic questions about these problems. At a high level, our findings can be summarized as "estimation complexity of polynomial-time SQ algorithms behaves like sample complexity" for many natural settings of those problems. This correspondence should not, however, be taken for granted. In many cases, the SQ version requires a completely different algorithm, and for some problems, we have not been able to provide upper bounds that match the sample complexity (see the following).

Given the fundamental role that the SQ model plays in a variety of settings, our primary motivation and focus is understanding the SQ complexity of these basic tasks for its own sake. At the same time our results lead to numerous applications, among which are new strong lower bounds for convex relaxations and results that subsume and improve on recent work that requires substantial technical effort.

As usual when exploring uncharted territory, some of the most useful results can be proven relatively easily given the wealth of existing literature on related topics. Still, for many questions, new insights and analyses are necessary (such as the characterization of the complexity of mean estimation for all $q \in [1, \infty)$), and we believe that those will prove useful in further research on the SQ model and its applications. There are also many interesting questions that we encountered but were not able to answer. We list some of those here:

1. How many samples are necessary and sufficient for answering the queries of our adaptive algorithms, such as those based on the inexact mirror descent? The answer to this question should shed new light on the power of adaptivity in statistical data analysis (Dwork et al. [32]).
2. Is there an SQ equivalent of upper bounds on sample complexity of mean estimation for uniformly smooth norms (see Appendix B for details)? Such a result would give a purely geometric characterization of estimation complexity of mean estimation.
3. In the absence of a general technique, such as the preceding one, there are still many important norms we have not addressed. Most notably, we do not know what is the estimation complexity of mean estimation in the spectral norm of a matrix (or other Schatten norms).
4. Is there an efficient algorithm for mean estimation (or at least linear optimization) with estimation complexity of $O(d/\varepsilon^2)$ for which a membership oracle for $\mathcal{K}$ suffices? (Our current algorithm is efficient only for a fixed $\mathcal{K}$ as it assumes knowledge of John's ellipsoid for $\mathcal{K}$.)

Subsequent work by Li et al. [62] partially resolved question 2 and completely resolved question 3 raised here. Namely, they extend our optimal mean estimation algorithms to arbitrary *symmetric* norms (i.e., invariant under permutations of coordinates and sign flips) of type 2 and show polynomial in d lower bounds for mean estimation in Schatten norms when $p \neq 2$. This shows, in particular, that uniform smoothness alone does not lead to (nearly) dimension-independent SQ mean estimation.

Acknowledgments

The authors thank Arkadi Nemirovski, Sasha Rakhlin, Ohad Shamir, and Karthik Sridharan for discussions and valuable suggestions about this work. Work was done while V. Feldman was at IBM Research–Almaden. Part of this work was done during C. Guzmán's internship at IBM Research–Almaden, at a postdoctoral position at Universidad de Chile, and at a postdoctoral position of Centrum Wiskunde & Informatica.

Appendix 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 $x, y \in E$,

$$\|x\|^r + \delta\|y\|^r \leq \frac{\|x+y\|^r + \|x-y\|^r}{2}. \tag{A.1}$$

From classical inequalities (see, e.g., Ball et al. [5]) 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 two-uniformly convex w.r.t. $\|\cdot\|_1$.
- When $1 < p \le 2$, the function $\Psi(x) = \frac{1}{2(p-1)}\|x\|_p^2$ is two-uniformly convex w.r.t. $\|\cdot\|_p$.
- When $2 < p < \infty$, the function $\Psi(x) = \frac{1}{p}\|x\|_p^p$ 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 $C \ge 1$ such that, for all $w, z \in E^*$,

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

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 $1 < q \le 2$, the norm $\|\cdot\|_q$ satisfies (A.2) with $s = q$ and $C = 1$; when $2 \le q < \infty$, the norm $\|\cdot\|_q$ satisfies (A.2) with $s = 2$ and $C = q - 1$. Finally, observe that, for ℓ_∞^d, we can use the equivalent norm $\|\cdot\|_{q(d)}$ with $q(d) = \ln d + 1$:

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

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

Appendix B. Sample Complexity of Mean Estimation

The following is a standard analysis based on Rademacher complexity and uniform convexity (see, e.g., Pisier [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 in which we optimize). By Observation 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

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

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

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

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

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

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

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

By simply upper bounding the quantity by $\varepsilon > 0$, we get a sample complexity bound for achieving ε accuracy in expectation, $n = \lceil C^{r/s}/\varepsilon^r \rceil$, where $C \ge 1$ is any constant satisfying (A.2). For the standard ℓ_p^d-setup, that is, 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 $n = O(\frac{\ln d}{\varepsilon^2})$ samples suffice.

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

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

Appendix C. Proof of Corollary 6

Note that, by Proposition 2, in order to obtain an ε-optimal solution to a nonsmooth convex optimization problem, it suffices to choose $\eta = \varepsilon/2$ and $T = \lceil r 2^r L_0^r D_\Psi(\mathcal{K})/\varepsilon^r \rceil$. Because $\mathcal{K} \subseteq \mathcal{B}_p(R)$, to satisfy (11), it is sufficient to have, for all $y \in \mathcal{B}_p(R)$,

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

Maximizing the left-hand side on y, we get a sufficient condition: $\|\nabla F(x) - \tilde{g}(x)\|_q R \le \eta/2$. We can satisfy this condition by solving the mean estimation problem in the 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 nonsmooth 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 consequence, solving the convex program amounts to using $O(d \cdot (\frac{L_0R}{\varepsilon})^2 \ln d)$ queries to STAT($\frac{\varepsilon}{4L_0R}$).
- $1 < p < 2$: 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 $O(d \log d \cdot \frac{1}{(p-1)} (\frac{L_0R}{\varepsilon})^2)$ queries to STAT($\Omega(\frac{\varepsilon}{[\log d] L_0R}$)).
- $p = 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = R^2$. As a consequence, solving the convex program amounts to using $O(d \cdot (\frac{L_0R}{\varepsilon})^2)$ queries to STAT($\Omega(\frac{\varepsilon}{L_0R})$).
- $2 < p < \infty$: 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 $O(d \log d \cdot 2^{2p-2} (\frac{L_0R}{\varepsilon})^p)$ queries to VSTAT($(\frac{64 L_0 R \log d}{\varepsilon})^p$).

□

Appendix D. Proof of Corollary 7

Similarly to in Appendix C, given $x \in \mathcal{K}$, we can obtain $\tilde{g}(x)$ by the mean estimation problem in ℓ_q-norm with error $\varepsilon/[12L_0R]$. (Notice that we have chosen $\eta = \varepsilon/6$.)

Now, by Proposition 3, in order to obtain an ε-optimal solution, it suffices to run the accelerated method for $T = \lceil \sqrt{2L_1 D_\Psi(\mathcal{K})/\varepsilon} \rceil$ iterations, each of them requiring $\tilde{g}$ as defined earlier. By using the two-uniformly convex functions for ℓ_p with $1 \le p \le 2$ from Appendix A, together with the bound on the number of queries and error for the mean estimation problems in the 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 consequence, solving the convex program amounts to using $O(d \cdot \sqrt{\ln d \cdot \frac{L_1R^2}{\varepsilon}})$ queries to STAT($\frac{\varepsilon}{12L_0R}$).
- $1 < p < 2$: 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 $O(d \log d \cdot \sqrt{\frac{1}{(p-1)} \cdot \frac{L_1R^2}{\varepsilon}})$ queries to STAT($\Omega(\frac{\varepsilon}{[\log d] L_0R}$)).
- $p = 2$: We have $r = 2$ and $D_\Psi(\mathcal{K}) = R^2$. As a consequence, solving the convex program amounts to using $O(d \cdot \sqrt{\frac{L_1R^2}{\varepsilon}})$ queries to STAT($\Omega(\frac{\varepsilon}{L_0R})$).

□

Endnotes

[1] In this context, an "empirical" version of SQ is used, which is referred to as counting or linear queries. It is now known that empirical values are close to expectations when differential privacy is preserved (Dwork et al. [32]).

[2] The analysis and bounds they give are inaccurate, but a similar conclusion follows from the bounds we give in Corollary 6.

[3] If d is not a power of two, we can first pad the input distribution to $\mathbb{R}^{d'}$, where $d' = 2^{\lceil \log d \rceil} \le 2d$.

[4] In Lyubarskii and Vershynin [66], complex vector spaces are considered, but the results also hold in the real case.

[5] We omit some necessary technical conditions, for example, measurability, for the gradient selection in the stochastic setting. We refer the reader to Rockafellar [74] for a detailed discussion.

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

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

References

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

[2] Balcan M, Blum A (2006) On a theory of learning with similarity functions. *ICML*, 73–80.

[3] Balcan MF, Feldman V (2013) Statistical active learning algorithms. *NIPS*, 1295–1303.
[4] Balcan M, Blum A, Fine S, Mansour Y (2012) Distributed learning, communication complexity and privacy. *COLT*, 26.1–26.22.
[5] Ball K, Carlen E, Lieb E (1994) Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones Mathematicae* 115(1):463–482.
[6] Bassily R, Smith A, Thakurta A (2014) Private empirical risk minimization: Efficient algorithms and tight error bounds. *FOCS*, 464–473.
[7] Bassily R, Nissim K, Smith AD, Steinke T, Stemmer U, Ullman J (2016) Algorithmic stability for adaptive data analysis. *Proc. 48th Annual ACM Sympos. Theory Comput.* (ACM, New York), 1046–1059.
[8] Belloni A, Liang T, Narayanan H, Rakhlin A (2015) Escaping the local minima via simulated annealing: Optimization of approximately convex functions. Grunwald P, Hazan E, Kale S, eds. *Proc. Machine Learn. Res. (Paris),* 240–265.
[9] Ben-David S, Dichterman E (1998) Learning with restricted focus of attention. *J. Comput. System Sci.* 56(3):277–298.
[10] Ben-Tal A, Nemirovski A (2013) Lectures on modern convex optimization. Accessed July 2016, http://www2.isye.gatech.edu/~nemirovs/.
[11] Bertsimas D, Vempala S (2004) Solving convex programs by random walks. *J. ACM* 51(4):540–556.
[12] Blum A, Dwork C, McSherry F, Nissim K (2005) Practical privacy: The SuLQ framework. Li C, ed. *Proc. PODS* (ACM, New York), 128–138.
[13] Blum A, Frieze A, Kannan R, Vempala S (1997) A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica* 22(1/2):35–52.
[14] Blum A, Furst M, Jackson J, Kearns M, Mansour Y, Rudich S (1994) Weakly learning DNF and characterizing statistical query learning using Fourier analysis. *Proc. STOC*, 253–262.
[15] Braun G, Guzmán C, Pokutta S (2017) Lower bounds on the oracle complexity of nonsmooth convex optimization via information theory. *IEEE Trans. Inform. Theory* 63(7):4709–4724.
[16] Bresler G, Gamarnik D, Shah D (2014) Structure learning of antiferromagnetic ising models. *NIPS*, 2852–2860.
[17] Bylander T (1994) Learning linear threshold functions in the presence of classification noise. *Proc. COLT*, 340–347.
[18] Cesa-Bianchi N, Conconi A, Gentile C (2004) On the generalization ability of on-line learning algorithms. *IEEE Trans. Inform. Theory* 50(9):2050–2057.
[19] Chaudhuri K, Monteleoni C, Sarwate AD (2011) Differentially private empirical risk minimization. *J. Machine Learn. Res.* 12:1069–1109.
[20] Chu C, Kim S, Lin Y, Yu Y, Bradski G, Ng A, Olukotun K (2006) Map-reduce for machine learning on multicore. *Proc. NIPS*, 281–288.
[21] Coja-Oghlan A, Cooper C, Frieze A (2010) An efficient sparse regularity concept. *SIAM J. Discrete Math.* 23(4):2000–2034.
[22] Dasgupta S, Kalai AT, Monteleoni C (2009) Analysis of perceptron-based active learning. *J. Machine Learn. Res.* 10:281–299.
[23] d'Aspremont A (2008) Smooth optimization with approximate gradient. *SIAM J. Optim.* 19(3):1171–1183.
[24] Devolder O, Glineur F, Nesterov Y (2013) First-order methods with inexact oracle: The strongly convex case. CORE Discussion Papers 2013016, Université Catholique de Louvain. Accessed July 1, 2016, http://EconPapers.repec.org/RePEc:cor:louvco:2013016.
[25] Devolder O, Glineur F, Nesterov Y (2014) First-order methods of smooth convex optimization with inexact oracle. *Math. Programming* 146(1–2):37–75.
[26] Dinur I, Nissim K (2003) Revealing information while preserving privacy. *PODS*, 202–210.
[27] Duchi JC, Jordan MI, Wainwright MJ (2013) Local privacy and statistical minimax rates. *FOCS*, 429–438.
[28] Duchi J, Jordan M, Wainwright M (2014) Privacy aware learning.*J. ACM* 61(6):1–57.
[29] Dunagan J, Vempala S (2008) A simple polynomial-time rescaling algorithm for solving linear programs. *Math. Programming* 114(1):101–114.
[30] Dwork C, Roth A (2014) The Algorithmic Foundations of Differential Privacy. *Foundations Trends Theories Comput. Sci.* 9(3–4):211–407.
[31] Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. *TCC*, 265–284.
[32] Dwork C, Feldman V, Hardt M, Pitassi T, Reingold O, Roth A (2014) Preserving statistical validity in adaptive data analysis. Preprint, submitted November 10, http://arxiv.org/abs/1411.2664.
[33] Dwork C, Feldman V, Hardt M, Pitassi T, Reingold O, Roth A (2015) Generalization in adaptive data analysis and holdout reuse. Preprint, submitted June 8, http://arxiv.org/abs/1506.02629.
[34] Feige U (2002) Relations between average case complexity and approximation complexity. STOC. (ACM), 534–543.
[35] Feldman V (2008) Evolvability from learning algorithms. *Proc. STOC*, 619–628.
[36] Feldman V (2009) A complete characterization of statistical query learning with applications to evolvability. *Proc. FOCS*, 375–384.
[37] Feldman V (2016) Dealing with range anxiety in mean estimation via statistical queries. Preprint, submitted November 20, http://arxiv.org/abs/1611.06475.
[38] Feldman V, Lee H, Servedio R (2011) Lower bounds and hardness amplification for learning shallow monotone formulas. COLT. vol. 19, 273–292.
[39] Feldman V, Perkins W, Vempala S (2013) On the complexity of random satisfiability problems with planted solutions. Preprint, submitted November 19, http://arxiv.org/abs/1311.4821.
[40] Feldman V, Grigorescu E, Reyzin L, Vempala S, Xiao Y (2012) Statistical algorithms and a lower bound for detecting planted cliques. Preprint, submitted January 5, http://arxiv.org/abs/1201.1214.
[41] Fiorini S, Massar S, Pokutta S, Tiwary H, de Wolf R (2012) Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *STOC*, 95–106.
[42] Forster J (2002) A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. System Sci.* 65(4):612–625.
[43] Freund Y, Schapire R (1998) Large margin classification using the Perceptron algorithm. *COLT*, 209–217.
[44] Goldreich O (2011) Candidate one-way functions based on expander graphs. Goldreich O, ed. *Studies in Complexity and Cryptography: Miscellanea on the Interplay Between Randomness and Computation*, Lecture Notes in ComputerScience, vol 6650 (Springer, Berlin, Heidelberg), 76–87.
[45] Grötschel M, Lovász L, Schrijver A (1988) *Geometric Algorithms and Combinatorial Optimization* (Springer, Berlin).
[46] Grove A, Littlestone N, Schuurmans D (1997) General convergence results for linear discriminant updates. *Proc. 10th Annual Conf. Comput. Learn. Theory*, 171–183.
[47] Guzmán C, Nemirovski A (2015) On lower complexity bounds for large-scale smooth convex optimization. *J. Complexity* 31(1):1–14.
[48] Hardt M, Rothblum G (2010) A multiplicative weights mechanism for privacy-preserving data analysis. *FOCS*, 61–70.
[49] Hardt M, Ullman J (2014) Preventing false discovery in interactive data analysis is hard. *FOCS*, 454–463.

[50] Hedayat A, Wallis WD (1978) Hadamard matrices and their applications. *Annals Statist.* 6(6):1184–1238.
[51] Hsu D, Sabato S (2013) Approximate loss minimization with heavy tails. Preprint, submitted July 7, http://arxiv.org/abs/1307.1827.
[52] John F (1948) Extremum problems with inequalities as subsidiary conditions. Studies and Essays Presented to R. Courant (Interscience Publishers, New York), 187–204.
[53] Kakade S, Sridharan K, Tewari A (2008) On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. NIPS (Curran Associates, Inc.), 793–800.
[54] Kalai AT, Vempala S (2006) Simulated annealing for convex optimization. *Math. Oper. Res.* 31(2):253–266.
[55] Kallweit M, Simon H (2011) A close look to margin complexity and related parameters. *COLT*, 437–456.
[56] Kashin B (1977) The widths of certain finite dimensional sets and classes of smooth functions. Izv. Akad. Nauk SSSR Ser. Mat, 334–351.
[57] Kasiviswanathan SP, Lee HK, Nissim K, Raskhodnikova S, Smith A (2011) What can we learn privately? *SIAM J. Comput.* 40(3):793–826.
[58] Kearns M (1998) Efficient noise-tolerant learning from statistical queries. *J. ACM* 45(6):983–1006.
[59] Khachiyan LG (1996) Rounding of polytopes in the real number model of computation. *Math. Oper. Res.* 21(2):307–320.
[60] Klivans A, Sherstov A (2007) Unconditional lower bounds for learning intersections of halfspaces. *Machine Learn*. 69(2–3):97–114.
[61] Lee J, Raghavendra P, Steurer D (2015) Lower bounds on the size of semidefinite programming relaxations. *STOC*, 567–576.
[62] Li J, Nikolov A, Razenshteyn I, Waingarten E (2019) On mean estimation for general norms with statistical queries. *Proc. Machine Learn. Res.*, vol. 99 (PMLR, Phoenix, AZ), 2158–2172.
[63] Littlestone N (1987) Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learn*. 2:285–318.
[64] Lovász L, Vempala S (2006) Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. *FOCS*, 57–68.
[65] Lovász L, Vempala S (2006) Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. System Sci.* 72(2):392–417.
[66] Lyubarskii Y, Vershynin R (2010) Uncertainty principles and vector quantization. *IEEE Trans.Inform. Theory* 56(7):3491–3501.
[67] Meka R, Potechin A, Wigderson A (2015) Sum-of-squares lower bounds for planted clique. *STOC*, 87–96.
[68] Nemirovsky A, Yudin D (1983) *Problem Complexity and Method Efficiency in Optimization* (J. Wiley Sons, New York).
[69] Nesterov Y (1983) A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Math. Doklady* 27(2):372–376.
[70] Novikoff A (1962) On convergence proofs on perceptrons. *Proc. Sympos. Math. Theory Automata*, vol. XII, 615–622.
[71] Pisier G (2016) *Martingales in Banach Spaces. Cambridge Studies in Advanced Mathematics* (Cambridge University Press, Cambridge, MA).
[72] Poljak B (1987) *Introduction to Optimization* (Optimization Software, New York).
[73] Raginsky M, Rakhlin A (2011) Information-based complexity, feedback and dynamics in convex programming. *IEEE Trans. Inform. Theory* 57(10):7036–7056.
[74] Rockafellar R (1974) *Conjugate Duality and Optimization*. Regional conference series in applied mathematics (Society for Industrial and Applied Mathematics, Philadelphia).
[75] Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.* 65(6):386–407.
[76] Roth A, Roughgarden T (2010) Interactive privacy via the median mechanism. *STOC*, 765–774.
[77] Rothvoß T (2014) The matching polytope has exponential extension complexity. *STOC*, 263–272.
[78] Roy I, Setty STV, Kilzer A, Shmatikov V, Witchel E (2010) Airavat: Security and privacy for MapReduce. *NSDI*, 297–312.
[79] Schoenebeck G (2008) Linear level lasserre lower bounds for certain k-csps. *FOCS*, 593–602.
[80] Servedio R (1999) On PAC learning using Winnow, Perceptron, and a Perceptron-like algorithm. *Proc. 12th Annual Conf. Computat. Learn. Theory*, 296–307.
[81] Shalev-Shwartz S, Ben-David S (2014) *Understanding Machine Learning: From Theory to Algorithms* (Cambridge University Press, Cambridge, MA).
[82] Shalev-Shwartz S, Shamir O, Srebro N, Sridharan K (2009) Stochastic convex optimization. *COLT*.
[83] Sherstov AA (2008) Halfspace matrices. *Comput. Complexity* 17(2):149–178.
[84] Shor N (2011) *Nondifferentiable Optimization and Polynomial Problems* (Springer, Berlin).
[85] Simon H (2007) A characterization of strong learnability in the statistical query model. *Proc. Sympos. Theoretical Aspects Comput. Sci.*, 393–404.
[86] Srebro N, Tewari A (2010) Stochastic optimization: ICML 2010 tutorial. Accessed July 1, 2016, http://www.ttic.edu/icml2010stochopt/.
[87] Steinhardt J, Valiant G, Wager S (2016) Memory, communication, and statistical queries. *COLT*, 1490–1516.
[88] Steinke T, Ullman J (2015) Interactive fingerprinting codes and the hardness of preventing false discovery. *COLT*, 1588–1628.
[89] Studer C, Goldstein T, Yin W, Baraniuk R (2014) Democratic representations. Preprint, submitted January 15, http://arxiv.org/abs/1401.3420.
[90] Sujeeth AK, Lee H, Brown KJ, Chafi H, Wu M, Atreya AR, Olukotun K, Rompf T, Odersky M (2011) OptiML: an implicitly parallel domain specific language for machine learning. *Proc. 28th Internat. Conf. Machine Learn.* (ACM, New York), 609–616.
[91] Ullman J (2015) Private multiplicative weights beyond linear queries. *PODS*, 303–312.
[92] Valiant LG (2009) Evolvability. *J. ACM* 56(1):3.1–3.216.
[93] Valiant P (2014) Evolvability of real functions. *ACM Trans. Comput. Theory* 6(3):12.1–12.19.
[94] Wang Z, Gu Q, Liu H (2014) Statistical-Computational phase transitions in planted models: The high-dimensional setting. *Proc. 31st Internat. Conf. Machine Learn*. 32(2):244–252.
[95] Warner SL (1965) Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60(309):63–69.