Statistical Query Algorithms for Mean Vector Estimation and Stochastic Convex Optimization[*]

Vitaly Feldman[†] Cristóbal Guzmán[‡§] Santosh Vempala[¶]

Abstract

Stochastic convex optimization, where the objective is the expectation of a random convex function, is an important and widely used method with numerous applications in machine learning, statistics, operations research and other areas. We study the complexity of stochastic convex optimization given only *statistical query* (SQ) access to the objective function. We show that well-known and popular 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 $\mathbb{R}^d$. 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.

1 Introduction

Statistical query (SQ) algorithms, defined by Kearns [47] in the context of PAC learning and by Feldman et al. [34] for general problems on inputs sampled i.i.d. from distributions, are algorithms that can be implemented using estimates of the expectation of any given function on a sample drawn randomly from the input distribution D instead of direct access to random samples. Such access is abstracted using a *statistical query oracle* that given a query function $\phi : \mathcal{W} \to [-1, 1]$ returns an estimate of $\mathbf{E}_{\mathbf{w}}[\phi(\mathbf{w})]$ within some tolerance τ (possibly dependent on ϕ). We will refer to the number of samples sufficient to estimate the expectation of each query of a SQ algorithm with some fixed constant confidence as its *estimation complexity* (often $1/\tau^2$) and the number of queries as its *query complexity*.

Statistical query access to data was introduced as means to derive noise tolerant algorithm in the PAC model of learning [47]. Subsequently, it has been realized that reducing data access to estimation of simple expectations has a wide variety of additional useful properties. It played a key role in the development of the notion of differential privacy [22, 12, 26] and has been subject of intense subsequent research in differential privacy[1] (see [25] for a literature review). It has important applications in a large number of other theoretical and practical contexts such as distributed data access [17, 61, 70], evolvability [72, 30, 31] and memory/communication limited machine learning [8, 4, 68]. Most recently, in a line of work initiated by Dwork et al. [27], SQs have been used as a basis for understanding generalization in adaptive data analysis [27, 42, 28, 69, 6].

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 $w^1, w^2, \dots, w^n$ 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.* [67, 63]). It is usually assumed that $\mathcal{K}$ is "known" to the algorithm (or in some cases given via a suffi-

[*] Extended abstract. Full version available at `https://arxiv.org/abs/1512.09170`

[†] IBM Research - Almaden. E-mail: `vitaly@post.harvard.edu`.

[‡] Facultad de Matemáticas & Escuela de Ingeniería, Pontificia Universidad Católica de Chile. E-mail: `crguzmanp@uc.cl`.

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

[¶] School of Computer Science, Georgia Institute of Technology. E-mail: `vempala@gatech.edu`.

[1] In this context an "empirical" version of SQs 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 [27].

ciently strong oracle) and the key challenge is understanding how to cope with estimation errors arising from the stochastic nature of information about $F(x)$.

Surprisingly, 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 was shown by Kearns [47] who proved that parity functions cannot be learned efficiently using SQs. Subsequent work has developed several techniques for proving such lower bounds (*e.g.* [10, 66, 34, 35]), established relationships to other complexity measures (*e.g.* [64, 45]) and provided lower bounds for many important problems in learning theory (*e.g.* [10, 48, 33]) and beyond [34, 35, 14, 74].

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.* [36, 60, 53, 49]). As was shown in [35], 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 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 like 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 linear case above 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 need to ensure that estimation errors from different iterations do not accumulate. Luckily, for this we can build on the study of the performance of first-order methods with inexact first-order oracles. Methods of this type have a long history (*e.g.* [57, 65]), 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 consider several traditional setups of convex optimization: non-smooth, smooth and strongly convex. While 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 what is 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 bounded range. The first one avoids gradients altogether by only using estimates of function values. It is based on random walk techniques of

Kalai and Vempala [44] and Lovász and Vempala [51]. 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 to obtain a global approximation relatively cheaply. Interestingly, these very different methods have the same estimation complexity which is also within factor of d of our lower bound.

Finally, we highlight some theoretical applications of our results. We show that our algorithms imply lower bounds for a broad and practically useful class of convex relaxations for any problem for which a sufficiently strong SQ lower bound is known. As an example, we demonstrate consequences for classic constraint satisfaction problems. We then show that our mean estimation algorithms imply dimension-independent estimation complexity for 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; (b) strengthening and generalization of algorithms for answering sequences of convex minimization queries differentially privately given by Ullman [71].

1.2 The Model

The algorithms we consider here have access to a statistical query oracle for the input distribution.

Definition 1.1. ([47, 34]) *Let D be a distribution over a domain $\mathcal{W}$, $\tau > 0$ and n be an integer. A statistical query oracle $STAT_D(\tau)$ is an oracle that given as input any function $\phi : \mathcal{W} \to [-1, 1]$, returns some value v such that $|v - \mathbf{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w})]| \le \tau$. A statistical query oracle $VSTAT_D(n)$ is an oracle that given as input any function $\phi : \mathcal{W} \to [0, 1]$ returns some value v such that $|v - p| \le \max\left\{\frac{1}{n}, \sqrt{\frac{p(1-p)}{n}}\right\}$, where $p \doteq \mathbf{E}_{\mathbf{w}\sim D}[\phi(\mathbf{w})]$. We say that an algorithm is* statistical query *(or, for brevity, just SQ) if it does not have direct access to n samples from the input distribution D, but instead makes calls to a statistical query oracle for the input distribution.*

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 $n = 1/\tau^2$. Note that the estimation complexity corresponds to the number of i.i.d. samples sufficient to simulate the oracle for a single query with at least some constant probability of success. However it is not necessarily true that the whole algorithm can be simulated using $O(n)$ samples since answers to many queries need to be estimated. Answering m fixed (or non-adaptive) statistical queries can be done using $O(\log m \cdot n)$ samples but when queries depend on previous answers more samples might be necessary (see [27] for a detailed discussion). Whenever that does not make a difference for our upper bounds on estimation complexity, we state results for STAT to ensure consistency with prior work in the SQ model. All our lower bounds are stated for the stronger VSTAT oracle.

1.3 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 $\tilde{w}$, such that for every $x \in \mathcal{K}$, $|\langle x, \bar{w}\rangle - \langle x, \tilde{w}\rangle| \le \varepsilon$. Given such an estimate $\tilde{w}$, we can solve the original problem with error of at most 2ε by solving $\min_{x\in\mathcal{K}}\langle x, \tilde{w}\rangle$.

An obvious way to estimate the high-dimensional mean using SQs is to simply estimate each of the coordinates of the mean vector using a separate SQ: that is $\mathbf{E}[\mathbf{w}_i/B_i]$, where $[-B_i, B_i]$ is the range of $\mathbf{w}_i$. Unfortunately, even in the most standard setting, where both $\mathcal{K}$ and $\mathcal{W}$ are ℓ_2 unit balls, this method requires accuracy that scales with $1/\sqrt{d}$ (or estimation complexity that scales linearly with d). In contrast, bounds obtained using samples are dimension-independent making this SQ implementation unsuitable for high-dimensional applications. Estimation of high-dimensional means for various distributions is 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.* [15, 11, 3]).

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.* [56]). Also some of the general techniques for proving upper bounds on sample complexity 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 and $\mathcal{W}$ is the unit ball in ℓ_q norm for $p \in [1, \infty]$ and $1/p + 1/q = 1$

(general radii can be reduced to this setting by scaling). This is equivalent to requiring that $\|\tilde{w} - \bar{w}\|_q \leq \varepsilon$ for a random variable $\mathbf{w}$ supported on the unit ℓ_q ball (denoted by $\mathcal{B}_q^d(1)$) and we refer to it as ℓ_q mean estimation. Even in this standard setting the picture is not so clean in the regime when $q \in [1, 2)$, where the sample complexity of ℓ_q mean estimation depends both on 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. This can be achieved using d queries to $\mathrm{STAT}(\varepsilon)$;
- For $q = 2$ we demonstrate that Kashin's representation of vectors introduced by Lyubarskii and Vershynin [52] gives a set of $2d$ measurements which allow to recover the mean with estimation complexity of $O(1/\varepsilon^2)$. This algorithm can be implemented using $2d$ queries to $\mathrm{STAT}(\Omega(\varepsilon))$.
- For $q \in (2, \infty)$ we use decomposition of the samples into $\log d$ "rings" in which non-zero coefficients have low dynamic range. For each ring we combine ℓ_2 and ℓ_∞ estimation to ensure low error in ℓ_q and nearly optimal estimation complexity. The algorithm uses $3d \log d$ queries to $\mathrm{STAT}(\varepsilon/\log(d))$.
- 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. This analysis also relies on the stronger VSTAT oracle. For small ε a better upper bound can be obtained via a reduction to ℓ_2 case. Hence the algorithm uses $2d \log d$ queries to $\mathrm{VSTAT}((16 \log(d)/\varepsilon)^p)$ or $2d$ queries to $\mathrm{STAT}(\Omega(d^{1/2-1/q}\varepsilon))$.

Nearly-Optimal ℓ_2 Mean Estimation Algorithm: In addition to the algorithm based on Kashin's representation, we give a randomized algorithm for ℓ_2 mean estimation that has slightly worse $O(\log(1/\varepsilon)/\varepsilon^2)$ estimation complexity but its analysis is simpler, more intuitive, and self-contained.

The algorithm uses coordinate-wise estimation in a randomly and uniformly chosen basis. By concentration of measure on the sphere, the coefficients of any unit vector in such a basis will have magnitude of at most $\tilde{O}(1/\sqrt{d})$ with high probability. This implies that we can estimate the mean of each coefficient up to $\varepsilon/\sqrt{d}$ by truncating the values of the coefficient that are larger than $\tilde{O}(1/\sqrt{d})$, rescaling to $[-1, 1]$ and then feeding the result as a query to $\mathrm{STAT}(\tilde{\Omega}(\varepsilon))$. The concentration results allow us to control the error due to truncation and ensure success with any constant probability. To obtain logarithmic dependence on the failure probability δ, we use an additional confidence amplification step via a high-dimensional analogue of the median-of-means procedure.

Lower bounds: We prove lower bounds for stochastic linear optimization over the ℓ_p unit ball and consequently also for ℓ_q mean estimation using the technique from [35]. The technique is based on bounding the statistical dimension with discrimination norm. The *discrimination norm* of a set of distributions $\mathcal{D}'$ relative to a distribution D is defined as:

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

where the norm of h over D is $\|h\|_D = \sqrt{\mathbf{E}_D[h^2(x)]}$ and $D' \sim \mathcal{D}'$ refers to choosing D' randomly and uniformly from the set $\mathcal{D}'$. We give a simple construction of a hard family of distributions with low discrimination norm. It leads to the following lower bounds:

THEOREM 1.1. *For any $q \geq 1$, $\varepsilon > 0$ and $r > 0$, $2^{\Omega(r)}$ queries to VSTAT(n) are necessary for ε-accurate stochastic linear optimization over $\mathcal{B}_p^d(1)$ with success probability at least $2/3$, where $n = \min\{d^{2/q-1}/(r\varepsilon^2), 1/(r\varepsilon^p)\}$. The same lower bound holds for ℓ_q mean estimation with error ε.*

Note that the lower bound on query complexity grows exponentially when estimation complexity is below that of our upper bounds (by at least a logarithmic factor). We summarize the bounds in Table 1.3 and compare them with those achievable using samples.

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 ellipsoid [43] for $\mathcal{W}$ and therefore depends on $\mathcal{K}$. Designing a single algorithm that given a sufficiently strong oracle for $\mathcal{K}$ (such as a separation oracle) can achieve the same estimation complexity for all $\mathcal{K}$ is an interesting open problem. This upper bound is nearly tight since even for $\mathcal{W}$ being the ℓ_1 ball we give a lower bound of $\tilde{\Omega}(d/\varepsilon^2)$.

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

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

1.4 The Gradient Descent family

The linear case gives us the basis for the study of the traditional setups of convex optimization for Lipschitz functions: non-smooth, smooth and strongly convex. The most basic Lipschitz (also called non-smooth) setup is determined by a bound on the magnitude of gradients in dual norm,

$$\mathcal{F}^0_{\|\cdot\|}(\mathcal{K}, L_0) \doteq \{f : \mathcal{K} \to \mathbb{R} :\ f \text{ convex}, \\ \|\nabla f(x)\|_* \le L_0,\ \forall x \in \mathcal{K}, \nabla f(x) \in \partial f(x)\}.$$

In the stochastic ℓ_p/ℓ_q setting we assume that for each w in the support of the distribution D and $x \in \mathcal{K}$, $\|\nabla f(x,w)\|_q \le L_0$ and the radius of $\mathcal{K}$ is bounded by R in ℓ_p norm. The smooth and strongly convex settings correspond to second order assumptions on F itself. For the two first classes of problems, our algorithms use global approximation of the gradient on $\mathcal{K}$ which as we know is necessary already in the linear case. However, for the strongly convex case we can show that an oracle introduced by Devolder et al. [21] only requires *local* approximation of the gradient, leading to improved estimation complexity bounds.

Specifically, we use our SQ mean estimation algorithms to efficiently implement the following approximate oracles.

Definition 1.2. ([19, 21, 20]) *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 *if for all $x \in \mathcal{K}$ $|\langle \tilde{g}(x) - \nabla F(x), y - u\rangle| \le \eta \quad \forall y, u \in \mathcal{K}$. 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 *if for all $x, y \in \mathcal{K}$*

(1.1)
$$\frac{\mu}{2}\|y-x\|^2 \le F(y) - [\tilde{F}(x) - \langle \tilde{g}(x), y-x\rangle] \le \frac{M}{2}\|y-x\|^2 + \eta.$$

In the non-smooth case we use the mirror-descent algorithm with η-approximate gradient to derive the following corollary for ℓ_p norms for $p \in [1,\infty]$ (we state only the $p \in [1,2]$ case for brevity):

Corollary 1.1. *Let $p \in [1,2]$, $L_0, R > 0$, and $\mathcal{K} \subseteq \mathcal{B}^d_p(R)$ be a convex body. There exists an SQ algorithm that solves any problem of the form $\min_{x\in\mathcal{K}}\{\mathbf{E}_{\mathbf{w}}[f(x,\mathbf{w})]\}$, where for all $w \in \mathcal{W}$, $f(\cdot, w) \in \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{K}, L_0)$, with accuracy ε using $O\left(d \log d \cdot \frac{1}{(p-1)}\left(\frac{L_0 R}{\varepsilon}\right)^2\right)$ queries to $STAT\left(\Omega\left(\frac{\varepsilon}{[\log d] L_0 R}\right)\right)$. For $p \in \{1,2\}$, $STAT(\Omega(\varepsilon/(L_0R)))$ suffices.*

For the smooth case we rely on the analysis by d'Aspremont [19] of an inexact variant of Nesterov's accelerated method [55], and for the strongly convex case we use the recent results by Devolder et al. [20] on the inexact dual gradient method which we implement using (η, M, μ)-oracle. We summarize our results for the ℓ_2 norm in Table 1.4. In the full version we provide detailed statements of these results and also demonstrate 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. [27], the number of samples needed for estimation of adaptive SQs using empirical means might scale linearly with the query complexity. While better bounds can be easily achieved in our case (logarithmic – as opposed to linear– in dimension), they are still worse

Objective	Inexact gradient method	Query complexity	Estimation complexity
Non-smooth	Mirror-descent	$O\left(d\cdot\left(\frac{L_0R}{\varepsilon}\right)^2\right)$	$O\left(\left(\frac{L_0R}{\varepsilon}\right)^2\right)$
Smooth	Nesterov	$O\left(d\cdot\sqrt{\frac{L_1R^2}{\varepsilon}}\right)$	$O\left(\left(\frac{L_0R}{\varepsilon}\right)^2\right)$
Strongly convex non-smooth	Dual gradient	$O\left(d\cdot\frac{L_0^2}{\varepsilon\kappa}\log\left(\frac{L_0R}{\varepsilon}\right)\right)$	$O\left(\frac{L_0^2}{\varepsilon\kappa}\right)$
Strongly convex smooth	Dual gradient	$O\left(d\cdot\frac{L_1}{\kappa}\log\left(\frac{L_1R}{\varepsilon}\right)\right)$	$O\left(\frac{L_0^2}{\varepsilon\kappa}\right)$

Table 2: Upper bounds for inexact gradient methods in the stochastic ℓ_2/ℓ_2 setting. 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.

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 SQ algorithms there is a threshold phenomenon in which as one goes below certain estimation complexity, the query complexity lower bound grows from polynomial to exponential very quickly (*e.g.* [34, 35]). For such lower bounds only the estimation complexity matters as long as the query complexity of the algorithm is polynomial.

1.5 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_0R on range is also implicit in the Lipschitz setting). This is a natural setting for stochastic optimization (and statistical algorithms, in particular) since even estimating the value of a given solution x with high probability and any desired accuracy from samples requires some assumptions about the range of most functions.

For simplicity we will assume $|f(x,w)|\le B=1$, although our results can be extended to the setting where only the variance of $f(x,\mathbf{w})$ is bounded by B^2 using the technique from [32]. Now, for every $x\in\mathcal{K}$, a single SQ for function $f(x,w)$ with tolerance τ gives a value $\tilde{F}(x)$ such that $|F(x)-\tilde{F}(x)|\le\tau$. This, as first observed by Valiant [73], gives a τ-approximate value (or zero-order) oracle for $F(x)$. It was proved by Nemirovsky and Yudin [54] and also by Grötschel et al. [39] (who refer to such oracle as *weak evaluation oracle*) that τ-approximate value oracle suffices to ε-minimize $F(x)$ over $\mathcal{K}$ with running time and $1/\tau$ being polynomial in $d,1/\varepsilon,\log(R_1/R_0)$, where $\mathcal{B}_2^d(R_0)\subseteq\mathcal{K}\subseteq\mathcal{B}_2^d(R_1)$. The analysis in [54, 39] is relatively involved and does not provide explicit bounds on τ.

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

Theorem 1.1. *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 poly*$(d,\frac{1}{\varepsilon},\log(R_1/R_0))$ *using 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 [44] and Lovász and Vempala [51]. This result was also independently obtained in a recent work of Belloni et al. [7] 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 [54] prove that even linear optimization over ℓ_2 ball of radius 1 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 approximate value oracle since the problem can be solved using SQs of tolerance $\tau=O(\varepsilon)$. Optimization with 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 [9] and approximate computation of the inertial ellipsoid [50] suffice for our purposes.

THEOREM 1.2. (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 $w \in \mathcal{W}, x \in \mathcal{K}$, $|f(x,w)| \leq 1$. 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 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 Thm. 1.1 and the lower bound in [54] show, the same intriguing gap is also present for approximate value oracle.

1.6 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 [72], adaptive data analysis [27] and distributed data analysis [17]).

1.6.1 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 SQ algorithms that are much stronger than known computational lower bounds for the problem. A classical example of such problem is learning of parity functions with noise (or, equivalently, finding an assignment that maximizes the fraction of satisfied XOR constraints). This implies that any algorithm that can be implemented using statistical queries with complexity below the lower bound cannot solve the problem. If the algorithm relies solely on some structural property of the problem, such as approximation of functions by polynomials or computation by a certain type of circuit, then we can immediately conclude a lower bound for that structural property. This indirect argument exploits the power of the algorithm and hence can lead to results which are harder to derive directly.

One inspiring example of this approach comes from using the statistical query algorithm for learning halfspaces [11]. The structural property it relies on is linear separability. Combined with the exponential lower bound for learning parities [47], it immediately implies that there is no mapping from $\{-1,1\}^d$ to $\mathbb{R}^N$ which makes parity functions linearly separable for any $N \leq N_0 = 2^{\Omega(d)}$. Subsequently, and apparently unaware of this technique, Forster [37] 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 [64] the connection between these two results is stated explicitly). His proof relies on a sophisticated and non-algorithmic technique and is considered a major breakthrough in proving lower bounds on the sign-rank of explicit matrices.

Convex optimization algorithms rely on existence of convex relaxations for problem instances that (approximately) preserve the value of the solution. Therefore given a SQ lower bound for a problem, our algorithmic results can be directly translated into lower bounds for convex relaxations of the problem. We now focus on a concrete example that is easily implied by our algorithm and a lower bound for planted constraint satisfaction problems from [35]. 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 exactly the refutation problem studied extensively over the past several decades (*e.g.* [29]). Now, consider the following common approach to the problem: define a convex domain $\mathcal{K}$ and map every k-clause C 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 relaxation is to ensure that for every satisfiable ϕ we have that $\min_{x\in\mathcal{K}} F_\phi(x) \leq \alpha$ for some fixed α. At the same time for a randomly chosen ϕ, we want to have with high probability $\min_{x\in\mathcal{K}} F_\phi(x) \geq \alpha + \varepsilon$. Ideally one would hope to get $\varepsilon \approx 2^{-k}$ since for sufficiently large m, every Boolean assignment leaves at least $\approx 2^{-k}$ fraction of the constraints unsatisfied. But (so called) integrality gap of this relaxation can reduce the difference to a smaller value.

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

Corollary 1.2. *For $p \in \{1, 2\}$, let $\mathcal{K} \subseteq \mathcal{B}_p^d(1)$ be a convex body and $\mathcal{F}_p \doteq \mathcal{F}^0_{\|\cdot\|_p}(\mathcal{K}, 1)$. Assume that there exists a mapping $\mathcal{M}$ that maps each k-clause $C \in X_k$ to a convex function $f_C \in \mathcal{F}_p$. Further assume that for some $\alpha \in \mathbb{R}$, $\varepsilon > 0$ and $m = \Omega(n/\varepsilon^2)$: If $\phi = C_1, \ldots, C_m$ is satisfiable then*

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

If ϕ is drawn from the uniform distribution U_k over k-SAT formulas of length m then

$$\Pr_{C_1,\ldots,C_m \sim U_k} \left[\min_{x \in \mathcal{K}} \left\{ \frac{1}{m} \sum_i f_{C_i}(x) \right\} > \alpha + \varepsilon \right] \geq 2/3.$$

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

For example, 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 (where the radius and the Lipschitz constant are at most 1). We are not aware of any existing techniques that imply comparable lower bounds. More importantly, our results imply that Corollary 1.2 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 which are obtained through various hierarchies of LP/SDP relaxations. For those cases additional structure of $\mathcal{K}$ was used to prove lower bounds that are not implied by our work. A more formal treatment of this technique can be found in the full version.

1.6.2 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 [15] and Blum et al. [11] 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 since Perceptron is usually used with a kernel (or in other high-dimensional space) and Winnow's main property is the logarithmic dependence of its sample complexity on the dimension. Formally, we get an algorithm with the following guarantees:

Theorem 1.3. *For every $p \in [2, \infty]$, there exists an efficient algorithm p-norm-SQ that for every $\varepsilon > 0$ and distribution D over $\mathcal{B}_p^d(R) \times \{-1, 1\}$ that is supported on examples (x, y) that for some vector $w^* \in \mathcal{B}_q^d(W)$ satisfy $y\langle w^*, x\rangle \geq \gamma$, outputs a halfspace w such that $\mathbf{Pr}_{(x,y)\sim D}[y\langle w, x\rangle < 0] \leq \varepsilon$. For $p \in [2, \infty)$ p-norm-SQ uses $O(d \log d (WR/\gamma)^2)$ queries to $VSTAT(O(\log d (WR/\gamma)^2/\varepsilon))$ and for $p = \infty$ p-norm-SQ uses $O(d \log d (WR/\gamma)^2)$ queries to $VSTAT(O((WR/\gamma)^2/\varepsilon))$.*

This implementation immediately implies the strongest known sample complexity bounds for learning halfspaces with random classification noise or differential privacy.

We note that a variant of the Perceptron algorithm referred to as Margin Perceptron outputs a halfspace that approximately maximizes the margin [2]. This allows it to be used in place of the SVM algorithm. Our SQ implementation of this algorithm gives an SVM-like algorithm with estimation complexity of $O(1/\gamma^2)$, where γ is the (normalized) margin. This is the same as the sample complexity of SVM (*cf.* [63]). Many other variants of the Perceptron and Winnow algorithms have been studied in the literature and applied in a variety of settings (*e.g.* [38, 62, 18]). The analysis inevitably relies on a margin assumption (and its relaxations) and hence, we believe, can be implemented using SQs in a similar manner.

1.6.3 Differential Privacy In local or *randomized-response* differential privacy the users provide the analyst with differentially private versions of their data points. Any analysis performed on such data is differentially private so, in effect, the data analyst need not be trusted. Such algorithms have been studied and applied for privacy preservation since at least the work of Warner [75] and have more recently been adopted in products by Google and Apple. While there exists a large and growing literature on mean estimation and convex optimization with (global) differential privacy (*e.g.* [16, 25, 5]), 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. [46] 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 [24, 23]. Our corollaries for mean estimation achieve the same bounds up to logarithmic in d factors. We also obtain corollaries for more general mean estimation problems and results for optimization that, to the best of our knowledge, were not previously known.

An additional implication in the context of differentially private data analysis is to the problem of releasing answers to multiple queries over a single dataset. A long line of research has considered this question for *linear* or *counting* queries which for a dataset $S \subseteq \mathcal{W}^n$ and function $\phi : \mathcal{W} \to [0,1]$ output an estimate of $\frac{1}{n}\sum_{w\in S}\phi(w)$ (see [25] 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 [59, 41] (albeit the running time is linear in $|\mathcal{W}|$). Recently, Ullman [71] has considered the question of answering *convex minimization* queries which ask for an approximate minimum of a convex program taking a data point as an input averaged over the dataset. For several convex minimization problems he gives algorithms that can answer an exponential number of convex minimization queries. It is easy to see that the problem considered by Ullman [71] is a special case of our problem by taking the input distribution to be uniform over the points in S. A statistical query for this distribution is equivalent to a counting query and hence our algorithms effectively reduce answering of convex minimization queries to answering of counting queries. Therefore an immediate corollary of our bounds is a strengthening and substantial generalization of the results in [71]. The detailed statement and comparison appear in the full version.

1.7 Related work

There is long history of research on the complexity of convex optimization with access to some type of oracle (*e.g.* [54, 13, 40]) with a lot of renewed interest due to applications in machine learning (*e.g.* [58, 1]). In particular, a number of works study robustness of optimization methods to errors by considering oracles that provide approximate information about F and its (sub-)gradients [19, 21]. Our approach to getting statistical query algorithms for stochastic convex optimization is based on establishing bridges to that literature.

A τ-approximate value (or zero-order) oracle for $F(x)$ is an oracle that for any $x \in \mathcal{K}$ returns a value $\tilde{F}(x)$ such that $|F(x) - \tilde{F}(x)| \le \tau$. It was proved by Nemirovsky and Yudin [54] and also by Grötschel et al. [39] that τ-approximate value oracle suffices to ε-minimize $F(x)$ over $\mathcal{K}$ with running time and $1/\tau$ being polynomial in $d, 1/\varepsilon, \log(R_1/R_0)$, where $\mathcal{B}_2^d(R_0) \subseteq \mathcal{K} \subseteq \mathcal{B}_2^d(R_1)$. If we assume that $|f(x,w)| \le B = 1$ for all w, then a single SQ for function $f(x,w)$ with tolerance τ gives a τ-approximate value of $F(x)$. Hence, as first observed in [73], it is possible to optimize general stochastic convex programs using SQs in polynomial time. Unfortunately and unavoidably, in many cases of interest this general approach leads to query and estimation complexity that are much worse than bounds that we get here since our bounds are based on gradient information and also exploit Lipschitz properties, either in the (standard) global norm or for suitable local norms, such as for the center-of-gravity method.

A common way to model stochastic optimization is via a stochastic oracle for the objective function [54]. Such oracle is assumed to return a random variable whose expectation is equal to the exact value of the function and/or its gradient (most commonly the random variable is Gaussian or has bounded variance). Analyses of such algorithms (most notably the Stochastic Gradient Descent (SGD)) are rather different from ours although in both cases linearity and robustness properties of first-order methods are exploited. In most settings we consider, estimation complexity of our SQ agorithms is comparable to sample complexity of solving the same problem using an appropriate version of SGD (which is, in turn, often known to be optimal). On the other hand lower bounds for stochastic oracles (*e.g.* [1]) have a very different nature and it is impossible to obtain superpolynomial lower bounds on the number of oracle calls (such as those we prove here).

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 [8, 34, 35]. In a recent (and independent) work Steinhardt et al. [68] have established a number of additional relationships between learning with SQs and learning with several types of restrictions on memory and communication. Among other results, they proved an unexpected upper bound on memory-bounded sparse least-squares regression by giving 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 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 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.

Given the fundamental role that SQ model plays in a variety of settings, our primary motivation and focus is understanding of 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 required substantial technical effort.

Acknowledgements

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

References

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

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

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

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

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

[6] R. Bassily, K. Nissim, A. D. Smith, T. Steinke, U. Stemmer, and J. Ullman. Algorithmic stability for adaptive data analysis. *CoRR*, abs/1511.02513, 2015. URL `http://arxiv.org/abs/1511.02513`.

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

[8] S. Ben-David and E. Dichterman. Learning with restricted focus of attention. *J. Comput. Syst. Sci.*, 56(3):277–298, 1998.

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

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

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

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

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

[14] G. Bresler, D. Gamarnik, and D. Shah. Structure learning of antiferromagnetic ising models. In *NIPS*, pages 2852–2860, 2014.

[2] The analysis and bounds they give are inaccurate but a similar conclusion follows from the bounds we give in Cor.1.1.

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

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

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

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

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

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

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

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

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

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

[25] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy (preprint)*. 2014.

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

[27] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Preserving statistical validity in adaptive data analysis. *CoRR*, abs/1411.2664, 2014. Extended abstract in STOC 2015.

[28] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506, 2015.

[29] U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543. ACM, 2002.

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

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

[32] V. Feldman. Dealing with range anxiety in the statistical query model. Preprint, 2016.

[33] V. Feldman, H. Lee, and R. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *COLT*, volume 19, pages 273–292, 2011.

[34] V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *arXiv, CoRR*, abs/1201.1214, 2012. Extended abstract in STOC 2013.

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

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

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

[38] Y. Freund and R. Schapire. Large margin classification using the Perceptron algorithm. In *COLT*, pages 209–217, 1998.

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

[40] C. Guzmán and A. Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1 – 14, 2015.

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

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

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

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

[45] M. Kallweit and H. Simon. A close look to margin complexity and related parameters. In *COLT*, pages 437–456, 2011.

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

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

[48] A. Klivans and A. Sherstov. Unconditional lower bounds for learning intersections of halfspaces. *Machine Learning*, 69(2-3):97–114, 2007.

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

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

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

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

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

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

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

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

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

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

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

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

[61] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for MapReduce. In *NSDI*, pages 297–312, 2010.

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

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

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

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

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

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

[68] J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. In *COLT*, pages 1490–1516, 2016.

[69] T. Steinke and J. Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *COLT*, pages 1588–1628, 2015. URL `http://jmlr.org/proceedings/papers/v40/Steinke15.html`.

[70] A. K. Sujeeth, H. Lee, K. J. Brown, H. Chafi, M. Wu, A. R. Atreya, K. Olukotun, T. Rompf, and M. Odersky. OptiML: an implicitly parallel domainspecific language for machine learning. In *ICML*, 2011.

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

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

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

[74] Z. Wang, Q. Gu, and H. Liu. Sharp computational-statistical phase transitions via oracle computational model. *arXiv*, abs/1512.08861, 2015. URL `http://arxiv.org/abs/1512.08861`.

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