Lower Bounds on the Oracle Complexity of Nonsmooth Convex Optimization via Information Theory

Gábor Braun, Cristóbal Guzmán, and Sebastian Pokutta

ISyE, Georgia Institute of Technology, Atlanta, GA 30332, USA.
Email: gabor.braun@isye.gatech.edu, cguzman@gatech.edu, sebastian.pokutta@isye.gatech.edu

Abstract

We present an information-theoretic approach to lower bound the oracle complexity of nonsmooth black box convex optimization, unifying previous lower bounding techniques by identifying a combinatorial problem, namely string guessing, as a single source of hardness. As a measure of complexity we use distributional oracle complexity, which subsumes randomized oracle complexity as well as worst-case oracle complexity. We obtain strong lower bounds on distributional oracle complexity for the box $[-1,1]^n$, as well as for the L^p-ball for $p \geq 1$ (for both low-scale and large-scale regimes), matching worst-case lower bounds, and hence we close the gap between distributional complexity, and in particular, randomized complexity, and worst-case complexity. Furthermore, the bounds remain essentially the same for high-probability and bounded-error oracle complexity, and even for combination of the two, i.e., bounded-error high-probability oracle complexity. This considerably extends the applicability of known bounds.

1 Introduction

For studying complexity of algorithms, oracle models are popular to abstract away from computational resources, i.e., to focus on *information* instead of *computation* as the limiting resource. Therefore oracle models typically measure complexity by the number of required queries to the oracle, whose bounds are often not subject to strong computational complexity assumptions, such as $P \neq NP$.

We study the complexity of nonsmooth convex optimization in the standard black box model. The task is to find the optimum of a function f, which is only accessible through a local oracle. The oracle can be queried with any point x of the domain, and provides information about f in a small neighborhood of x. This generic model captures the behavior of most first-order methods, successfully applied in engineering Ben-Tal et al. [2001], machine learning Lacoste-Julien et al. [2013], image and signal processing Beck and Teboulle [2009a], Zhu et al. [2010], and compressed sensing Beck and Teboulle [2009b], Nesterov and Nemirovski [2013]. All these applications require only medium accuracy solutions due to noisy data. Moreover, in the era of big data, other general-purpose methods, such as interior-point or Newton-type, are prohibitively expensive. This makes a strong case for using first-order methods, and the black box model has been extensively studied for various function classes and domains such as L^p-balls and the box. Most of the lower bounds were established in Nemirovski and Yudin [1983], Nemirovski [1994], Nesterov [2004]. These bound worst-case complexity using the technique of *resisting oracles*, continuously changing the function f to provide the less informative consistent answers. In particular, for Lipschitz continuous convex functions on the n-dimensional L^p-ball for $1 \leq p < \infty$, depending on the accuracy ε, two regimes of interest were established: the high accuracy or *low-scale* regime, where the dimension n appears as a multiplicative term in the complexity $\Theta(n \log 1/\varepsilon)$, and the low accuracy or *large-scale* regime, where the complexity $\Theta(1/\varepsilon^{\max\{p,2\}})$ is independent of the dimension. Interestingly, each of the two regimes has its own optimal method: the Center of Gravity method in the low-scale regime, and the Mirror-Descent method in the large-scale Nemirovski and Yudin [1983].

We provide a unification of lower bounds on the oracle complexity for nonsmooth convex optimization. We will also identify a core combinatorial problem, namely, a string guessing problem, from which we derive all our lower bounds for convex optimization. Thus, we obtain strong lower bounds on distributional oracle complexity in the nonsmooth case, matching all known bounds of the worst-case. In fact, we will even show that these bounds do not only hold in expectation but also with high probability, and even for Monte Carlo algorithms, which provide correct answer only with a bounded error probability.

The core problem will be handled by information theory, which is a natural approach due to the informational nature of oracle models. Information theory has been prominently used to obtain strong lower bounds in other complexity problems as well.

Related work

Our approach through information theory was motivated by the following works obtaining information-theoretic lower bounds on: communication Braverman and Rao [2011], Braverman et al. [2012], data structures Pătraşcu [2011], Dasgupta et al. [2012], extended formulations Braverman and Moitra [2012], Braun and Pokutta [2013], Braun et al. [2013], streaming computation Chakrabarti et al. [2013], and many more. Lower bounds were established for many other classical oracle settings, such as submodular function minimization with access to function value oracles Goel et al. [2009], Iwata and Nagano [2009], Svitkina and Fleischer [2011], however typically not explicitly relying on information theory but rather bounding the randomized complexity by means of Yao's minimax principle. For first-order oracles, algorithms have been proposed Chudak and Nagano [2007], however next to nothing is known about strong lower bounds.

For convex optimization, oracles based on linear optimization have been studied extensively and lower bounds on the number of queries are typically obtained by observing that each iteration adds only one vertex at a time Jaggi [2013], Lan [2013], Garber and Hazan [2013]. These oracles are typically weaker than general local oracles.

The study of oracle complexity started with the seminal work Nemirovski and Yudin [1983], where worst-case complexity is determined up to a constant factor for several function families. (see also Nemirovski [1994], Nesterov [2004] for alternative proofs and approaches). Interestingly, these bounds were extended to randomized oracle complexity at the price of a logarithmic multiplicative gap [Nemirovski and Yudin, 1983, 4.4.3 Proposition 2], and also to stochastic oracles. In this paper, we do not consider stochastic oracles; for recent lower bounds, see Raginsky and Rakhlin [2011], Agarwal et al. [2012]. However, we will employ function families similar to those used in Nemirovski and Yudin [1983] for lower bounds on worst-case oracle complexity.

An interesting result by Srebro and Sridharan [2012] provides a general lower complexity bound for Lipschitz convex functions in terms of fat shattering numbers for the worst case. As expected, our lower bounds coincide with these fat shattering numbers, but hold under more general assumptions, namely the distributional setting.

While our lower bounds are obtained in a fashion somewhat similar to those in statistical minimax theory, the key in our approach is actually in estimating what is learned from each obtained subgradient given what has been learned from previous subgradients—in statistical minimax theory, we typically take (random!) samples drawn i.i.d (see Raginsky and Rakhlin [2011] for a detailed discussion).

Contribution

We unify lower bounding techniques for convex nonsmooth optimization by identifying a common source of hardness and introducing an emulation mechanism that allows us to reduce different convex optimization settings to this setup. Our arguments are surprisingly simple, allowing for a unified treatment.

Information-theoretic framework We present an information-theoretic framework to lower bound the oracle complexity of any type of oracle problem. The key insight is that if the information content of the oracle answer to a query is low on average, then this fact alone is enough for establishing a strong lower bound on both the distributional and the high probability complexity.

Common source of hardness Our base problem is learning a hidden string via guessing, called the *String Guessing Problem (SGP)*. In Proposition 3.3 we establish a strong lower bound on the distributional and high probability oracle complexity of the string guessing problem, even for algorithms with bounded error. These bounds on the oracle complexity are established via a new information-theoretic framework for iterative query-based algorithms.

We then introduce a special reduction mechanism, an *emulation* in Definition 3.4, rewriting algorithms and oracles between different problems, see Lemma 3.5. This will be the common framework for our lower bounds.

First lower bounds for distributional and high-probability complexity for all local oracles First, we establish lower bounds on the complexity for a simple class of first-order local oracles for Lipschitz continuous convex functions both on the L^∞-box in Theorem 4.2 and on the L^p-ball in Theorem 5.1.

In Section 6 we extend all lower bounds in Theorem 6.2 to arbitrary local oracles by using *random perturbation* instead of adaptive perturbation as in Nemirovski and Yudin [1983]. These bounds match classical lower bounds on worst-case complexity (see Figure 1), but established for distributional oracle complexity, i.e., average case complexity, and high-probability oracle complexity.

Finally, our analysis extends to bounded-error algorithms: even if the algorithm is allowed to provide erroneous answer with a bounded probability (e.g. discard a bounded subset of instances, or be correct only with a certain probability on every instance), essentially the same lower bounds hold.

Closing the gap between randomized and worst-case oracle complexity In the case of the box as well as the L^p-ball for $1 \leq p \leq \infty$, our bounds show that all four complexity measures coincide, namely, high-probability, distributional, randomized, and worst-case complexity. This not only simplifies the proofs in Nemirovski and Yudin [1983] for randomized complexity, but also closes the gap between worst-case and randomized complexity ([Nemirovski and Yudin, 1983, 4.4.3 Proposition 2]).

2 Preliminaries

2.1 Convex functions and approximate solutions

In the following, let X be a convex body in $(\mathbb{R}^n, \|\cdot\|)$ (full dimensional compact convex set). We denote by $B_p(x, r)$ the ball in $\mathbb{R}^n$ centered at x with radius r in the L^p norm, where $1 \leq p \leq \infty$. Recall that $B_\infty(x, r) = \prod_{i=1}^n [x_i - r, x_i + r]$. Let e_i denote the i-th coordinate vector in $\mathbb{R}^n$.

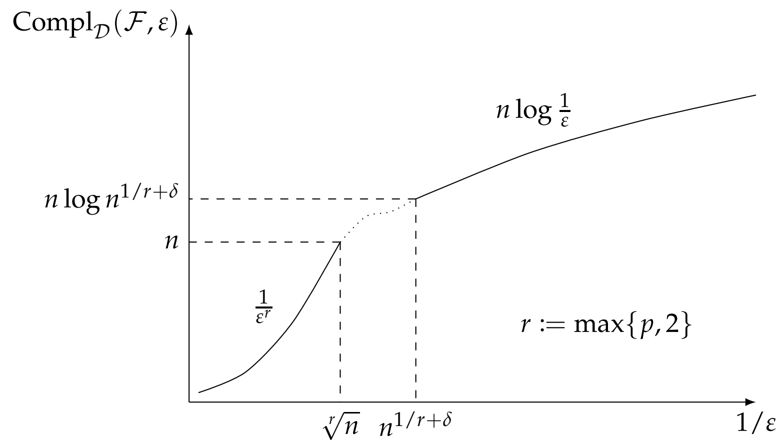

Figure 1: Distributional complexity as a function of $1/\varepsilon$ for the L^p-ball, $1 \leq p < \infty$.

Let $\mathcal{F}$ be a family of real valued, Lipschitz continuous convex functions on X with Lipschitz constant L. Recall that a function $f\colon X \to \mathbb{R}$ is Lipschitz continuous in the norm $\|\cdot\|$ with Lipschitz constant L if for all $x, y \in X$, we have $|f(x) - f(y)| \leq L \|x - y\|$.

For each $f \in \mathcal{F}$, let $f^* := \min_{x \in X} f(x)$. Given an accuracy level $\varepsilon > 0$, an *ε-minimum of f* is a point $x \in X$ satisfying $f(x) < f^* + \varepsilon$. The set of ε-minima will be denoted by $\mathcal{S}_\varepsilon(f)$.

In general, an ε-minimum need not identify f uniquely. However, it simplifies the analysis when ε-minima identify the function instance, as this makes optimization equivalent to learning the instance. We call this the packing property:

Definition 2.1 (Packing property)**.** A function family $\mathcal{F}$ satisfies the *packing property* for an accuracy level ε, if no two different members $f, g \in \mathcal{F}$ have common ε-minima, i.e., $\mathcal{S}_\varepsilon(f) \cap \mathcal{S}_\varepsilon(g) = \emptyset$.

2.2 Oracles and Complexity

We analyze the distributional complexity of approximating solutions in convex optimization under the standard black box oracle model, where oracle-based algorithms A have access to the instance f only by querying an oracle $\mathcal{O}$. *Oracle complexity* is measured by the least number of queries $T_A(f, \varepsilon)$ required to find an ε-minimum.

The oracle answer to query x will be denoted by $\mathcal{O}_f(x)$. When the function instance f is clear from the context we shall omit the subscript f.

The considered oracles $\mathcal{O}$ are *local*, i.e., if f and g are two instances equal in some neighborhood of x, then $\mathcal{O}_f(x) = \mathcal{O}_g(x)$. An important case is a *first-order* oracle, which answers a query $x \in X$ by the function value $f(x)$ and a subgradient of f at x. Note however that not every first-order oracle is local: at singular points of f a non-local oracle can choose between various subgradients taking into account the whole f, thereby e.g., encoding an ε-minimum.

Let $\mathcal{A}(\mathcal{O})$ denote the set of deterministic algorithms based on oracle $\mathcal{O}$. The *worst-case* oracle complexity is defined as

$$\mathrm{Compl}_{WC}(\mathcal{O}, \mathcal{F}, \varepsilon) := \inf_{A \in \mathcal{A}(\mathcal{O})} \sup_{f \in \mathcal{F}} T_A(f, \varepsilon).$$

Following Nemirovski and Yudin [1983], *randomized complexity* is defined as

$$\mathrm{Compl}_{\mathcal{R}}(\mathcal{O}, \mathcal{F}, \varepsilon) := \inf_{A \in \Delta(\mathcal{A}(\mathcal{O}))} \sup_{f \in \mathcal{F}} \mathbb{E}_A \left[T_A(f, \varepsilon) \right],$$

where $\Delta(\mathcal{B})$ is the set of probability distributions on the set $\mathcal{B}$.

The measure we will bound in our work is the even weaker notion of *distributional complexity*

$$\mathrm{Compl}_{\mathcal{D}}(\mathcal{O}, \mathcal{F}, \varepsilon) \coloneqq \sup_{F \in \Delta(\mathcal{F})} \inf_{A \in \mathcal{A}(\mathcal{O})} \mathbb{E}_F \left[T_A(F, \varepsilon) \right],$$

leading to stronger lower bounds. We will also bound the *high-probability oracle complexity* defined as

$$\mathrm{Compl}_{\mathcal{HP}}(\mathcal{O}, \mathcal{F}, \varepsilon) \coloneqq \sup_{F \in \Delta(\mathcal{F})} \inf_{A \in \mathcal{A}(\mathcal{O})} \sup_{\tau : \mathbb{P}_F[T_A(F,\varepsilon) \geq \tau] = 1 - o(1)} \tau,$$

i.e., it is the number τ of required queries that any algorithm needs for the worst distribution with high probability. It is easily seen that

$$\mathrm{Compl}_{\mathcal{HP}}(\mathcal{O}, \mathcal{F}, \varepsilon) \leq \mathrm{Compl}_{\mathcal{D}}(\mathcal{O}, \mathcal{F}, \varepsilon) \leq \mathrm{Compl}_{\mathcal{R}}(\mathcal{O}, \mathcal{F}, \varepsilon) \leq \mathrm{Compl}_{WC}(\mathcal{O}, \mathcal{F}, \varepsilon),$$

but it is open for which families $\mathcal{F}$ this inequality chain is tight, e.g., whether Yao's min-max principle applies (see e.g., Arora and Barak [2009]), as both $\mathcal{F}$ and $\mathcal{A}$ might be *a priori* infinite families. However it is known that worst-case and randomized complexity are equal up to a factor logarithmic in the dimension for several cases (see [Nemirovski and Yudin, 1983, 4.4.3 Proposition 2]).

2.3 Algorithm-oracle communication and string operations

For a given oracle-based (not necessarily minimization) algorithm, we record the communication between the algorithm and the oracle. Let Q_t be the t-th query of the algorithm and A_t be the t-th oracle answer. Thus $\Pi_t \coloneqq (Q_t, A_t)$ is the t-th query-answer pair. The full transcript of the communication is denoted by $\Pi = (\Pi_1, \Pi_2, \dots)$, and for given $t \geq 0$ partial transcripts are defined as $\Pi_{\leq t} \coloneqq (\Pi_1, \dots, \Pi_t)$ and $\Pi_{<t} \coloneqq (\Pi_1, \dots, \Pi_{t-1})$. By convention, $\Pi_{<1}$ and $\Pi_{\leq 0}$ are empty sequences.

As we will index functions by strings, let us introduce the necessary string operations. Let $s \in \{0,1\}^*$ be a binary string, then $s^{\oplus(i)}$ denotes the string obtained from s by flipping the i-th bit and deleting all bits following the i-th one. Let $s \sqsubseteq t$ denote that s is a prefix of t and $s \parallel t$ denote that neither is a prefix of the other. As a shorthand let $s|_l$ be the prefix of s consisting of the first l bits. We shall write $s0$ and $s1$ for the strings obtained by appending a 0 and 1 to s, respectively. Furthermore, the empty string is denoted by $\perp$. In the following we use the shorthand notation $[n] \coloneqq \{1, \dots, n\}$.

2.4 Information Theory

Notions from information theory are standard as defined in Cover and Thomas [2006]; we recall here those we need later. From now on, $\log(.)$ denotes the binary logarithm and capital letters will typically represent random variables or events. We can describe an event E as a random variable by the indicator function $I(E)$, which takes value 1 if E happens, and 0 otherwise.

The *entropy* of a discrete random variable A is

$$\mathbb{H}\left[A\right] \coloneqq - \sum_{a \in \mathrm{range}(A)} \mathbb{P}\left[A = a\right] \log \mathbb{P}\left[A = a\right].$$

This definition extends naturally to *conditional entropy* $\mathbb{H}\left[A \mid B\right]$ by using the corresponding conditional distribution and taking expectation, i.e., $\mathbb{H}\left[A \mid B\right] = \sum_b \mathbb{P}\left[B = b\right] \mathbb{H}\left[A \mid B = b\right]$.

Fact 2.2 (Properties of entropy)**.**

Bounds $0 \leq \mathbb{H}\left[A\right] \leq \log |\mathrm{range}(A)|$

$\mathbb{H}\left[A\right] = \log |\mathrm{range}(A)|$ if and only if A is uniformly distributed.

Monotonicity $\mathbb{H}[A] \geq \mathbb{H}[A \mid B]$;

The notion of *mutual information* defined as $\mathbb{I}[A;B] := \mathbb{H}[A] - \mathbb{H}[A \mid B]$ of two random variables A and B captures how much information about a 'hidden' A is leaked by observing B. Sometimes A and B are a collection of variables, then a comma is used to separate the components of A or B, and a semicolon to separate A and B themselves: e.g., $\mathbb{I}[A_1, A_2; B] = \mathbb{I}[(A_1, A_2); B]$. Mutual information is a symmetric quantity and naturally extends to *conditional mutual information* $\mathbb{I}[A; B \mid C]$ as in the case of entropy. Clearly, $\mathbb{H}[A] = \mathbb{I}[A; A]$.

Fact 2.3 (Properties of mutual information)**.**

Bounds If A is a discrete variable, then $0 \leq \mathbb{I}[A; B] \leq \mathbb{H}[A]$

Chain rule $\mathbb{I}[A_1, A_2; B] = \mathbb{I}[A_1; B] + \mathbb{I}[A_2; B \mid A_1]$.

Symmetry $\mathbb{I}[A; B] = \mathbb{I}[B; A]$.

Independent variables The variables A and B are independent if and only if $\mathbb{I}[A; B] = 0$.

3 Source of hardness and oracle emulation

We provide a general method to lower bound the number of queries of an algorithm that identifies a hidden random variable. This method is based on information theory and will allow us to lower bound the distributional and high probability oracle complexity, even for bounded-error algorithms. We apply this technique to the problem of identifying a random binary string, which we call the String Guessing Problem. Finally, we introduce an oracle emulation technique, that will allow us to compare the complexity of different oracles solving the same problem.

3.1 Information-theoretic lower bounds

We consider an unknown instance F that is randomly chosen from a finite family $\mathcal{F}$ of instances. For a given algorithm querying an oracle $\mathcal{O}$, let T be the number of queries the algorithm asks to determine the instance. Of course, the number T may depend on the instance, as algorithms can adapt their queries according to the oracle answers. However, we assume that $T < \infty$ almost surely, i.e., we require algorithms to almost always terminate (this is a mild assumption as $\mathcal{F}$ is finite).

Algorithms are allowed to have an error probability bounded by P_e, i.e., the algorithm is only required to return the correct answer with probability $1 - P_e$ *across all instances*. The latter statement is important as both, being perfectly correct on a $1 - P_e$ fraction of the input and outputting garbage in P_e cases, as well as providing the correct answer for *each instance* with probability $1 - P_e$, are admissible here.

For bounded-error algorithms, the high-probability complexity is the required number of queries to produce a correct answer with probability $1 - P_e - o(1)$. This adjustment is justified, as a wrong answer is allowed with probability P_e.

Lemma 3.1. *Let F be a random variable with finite range $\mathcal{F}$. For a given algorithm determining F via querying an oracle, with error probability bounded by P_e, suppose that the useful information of each oracle answer is bounded, i.e., for some constant C*

$$\mathbb{I}[F; A_t \mid \Pi_{<t}, Q_t, T \geq t] \leq C, \qquad t \geq 0.$$

Then, the distributional oracle complexity of the algorithm is lower bounded by

$$\mathbb{E}[T] \geq \frac{\mathbb{H}[F] - \mathbb{H}[P_e] - P_e \log|\mathcal{F}|}{C}.$$

Moreover, for all t we have

$$\mathbb{P}[T < t] \leq \frac{\mathbb{H}[P_e] + P_e \log|\mathcal{F}| + Ct}{\mathbb{H}[F]}.$$

In particular, if F is uniformly distributed, then $\mathbb{P}[T = \Omega(\log|\mathcal{F}|)] = 1 - P_e - o(1)$.

Proof. By induction on t we will first prove the following claim

$$\mathbb{I}[F;\Pi] = \sum_{i=1}^{t} \mathbb{I}[F;\Pi_i \mid \Pi_{<i}, T \geq i]\, \mathbb{P}[T \geq i] + \mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t]\, \mathbb{P}[T \geq t]. \tag{1}$$

The case $t = 0$ is obvious. For $t > 0$, note that the event $T = t$ is independent of F given $\Pi_{\leq t}$, as at step t the algorithm has to decide whether to continue based solely on the previous oracle answers and private random sources. If the algorithm stops, then $\Pi = \Pi_{\leq t}$. Therefore,

$$\begin{aligned}
\mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t] &= \mathbb{I}[F;\Pi, I(T=t) \mid \Pi_{\leq t}, T \geq t] \\
&= \underbrace{\mathbb{I}[F; I(T=t) \mid \Pi_{\leq t}, T \geq t]}_{=0} + \mathbb{I}[F;\Pi \mid \Pi_{\leq t}, I(T=t), T \geq t] \\
&= \underbrace{\mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T = t]}_{=0,\text{ as } \Pi_{\leq t} = \Pi} \mathbb{P}[T = t \mid T \geq t] + \mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t+1]\, \mathbb{P}[T \geq t+1 \mid T \geq t] \\
&= \left(\mathbb{I}[F;\Pi_{t+1} \mid \Pi_{<t+1}, T \geq t+1] + \mathbb{I}[F;\Pi \mid \Pi_{\leq t+1}, T \geq t+1]\right) \mathbb{P}[T \geq t+1 \mid T \geq t],
\end{aligned}$$

obtaining the identity

$$\mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t]\, \mathbb{P}[T \geq t] = \left(\mathbb{I}[F;\Pi_{t+1} \mid \Pi_{<t+1}, T \geq t+1] + \mathbb{I}[F;\Pi \mid \Pi_{\leq t+1}, T \geq t+1]\right) \mathbb{P}[T \geq t+1],$$

from which the induction follows.

Now, in (1) by letting t go to infinity, $\mathbb{P}[T \geq t]$ will converge to 0, while $\mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t]$ is bounded by $\mathbb{H}[F]$, proving that

$$\mathbb{I}[F;\Pi] = \sum_{i=1}^{\infty} \mathbb{I}[F;\Pi_i \mid \Pi_{<i}, T \geq i]\, \mathbb{P}[T \geq i]. \tag{2}$$

Note that Q_i is chosen solely based on $\Pi_{<i}$, and is conditionally independent of F. Therefore, by the chain rule, $\mathbb{I}[F;\Pi_i \mid \Pi_{<i}, T \geq i] = \mathbb{I}[F; A_i \mid \Pi_{<i}, Q_i, T \geq i]$. Plugging this equation into (2), we obtain

$$\mathbb{I}[F;\Pi] = \sum_{i=1}^{\infty} \mathbb{I}[F; A_i \mid \Pi_{<i}, Q_i, T \geq i]\, \mathbb{P}[T \geq i] \leq C \sum_{i=0}^{\infty} \mathbb{P}[T \geq i] = C \cdot \mathbb{E}[T].$$

Finally, as the algorithm determines F with error probability at most P_e, Fano's inequality [Cover and Thomas, 2006, Theorem 2.10.1] applies

$$\mathbb{H}[F \mid \Pi] \leq \mathbb{H}[P_e] + P_e \log|\mathcal{F}|. \tag{3}$$

We therefore obtain

$$\mathbb{H}[F] = \mathbb{H}[F \mid \Pi] + \mathbb{I}[F;\Pi] \leq \mathbb{H}[P_e] + P_e \log|\mathcal{F}| + C \cdot \mathbb{E}[T],$$

and therefore

$$\mathbb{E}[T] \geq \frac{\mathbb{H}[F] - \mathbb{H}[P_e] - P_e \log|\mathcal{F}|}{C},$$

as claimed.

We will now establish concentration for the number of required queries. For this we reuse (1), the split-up of information up to query t:

$$\begin{aligned}
\mathbb{I}[F;\Pi] &= \sum_{i=1}^{t} \mathbb{I}[F;\Pi_i \mid \Pi_{<i}, T \geq i]\,\mathbb{P}[T \geq i] + \mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t]\,\mathbb{P}[T \geq t] \\
&= \sum_{i=1}^{t} \underbrace{\mathbb{I}[F;A_i \mid \Pi_{<i}, Q_i, T \geq i]}_{\leq C}\,\mathbb{P}[T \geq i] + \underbrace{\mathbb{I}[F;\Pi \mid \Pi_{\leq t}, T \geq t]}_{\leq \mathbb{H}[F]}\,\mathbb{P}[T \geq t] \\
&\leq Ct + \mathbb{H}[F]\,\mathbb{P}[T \geq t],
\end{aligned}$$

which we combine with (3):

$$\mathbb{H}[F] = \mathbb{H}[F \mid \Pi] + \mathbb{I}[F;\Pi] \leq \mathbb{H}[P_e] + P_e \log|\mathcal{F}| + Ct + \mathbb{H}[F]\,\mathbb{P}[T \geq t],$$

and therefore

$$\mathbb{P}[T < t] \leq \frac{\mathbb{H}[P_e] + P_e \log|\mathcal{F}| + Ct}{\mathbb{H}[F]}.$$

Specializing to uniform distributions provides the last claim of the Lemma. □

3.2 Identifying binary strings

For a fixed length M we consider the problem of identifying a hidden string $S \in \{0,1\}^M$ picked uniformly at random. The oracle $\mathcal{O}_S$ accepts queries for any part of the string. Formally, a query is a pair (s, σ), where s is a string of length at most M, and $\sigma\colon [|s|] \to [M]$ is an embedding indicating an order of preference. The intent is to ask whether $S_{\sigma(k)} = s_k$ for all k. The oracle will reveal the smallest k so that $S_{\sigma(k)} \neq s_k$ if such a k exists or will assert correctness of the guessed part of the string. More formally we have:

Oracle 3.2 (String Guessing Oracle $\mathcal{O}_S$)**.**
Query: A string $s \in \{0,1\}^{\leq M}$ and an injective function $\sigma\colon [|s|] \to [M]$.
Answer: Smallest $k \in \mathbb{N}$ so that $S_{\sigma(k)} \neq s_k$ if it exists, otherwise `EQUAL`.

From Lemma 3.1 we can establish an expectation and high probability lower bound on the number of queries, even for bounded error algorithms. The key is that the oracle does not reveal any information about the bits after the first wrongly guessed bit, not even involuntarily.

Proposition 3.3 (String Guessing Problem)**.** *Let M be a positive integer, and S be a uniformly random binary string of length M. Let $\mathcal{O}_S$ be the String Guessing Oracle (Oracle 3.2). Then for any bounded error algorithm having access to S only through $\mathcal{O}_S$, the expected number of queries required to identify S with error probability at most P_e is at least $[(1-P_e)M - 1]/2$. Moreover, $\mathbb{P}[T = \Omega(M)] = 1 - P_e - o(1)$, where T is the number of queries.*

Proof. We will prove the following claim by induction: At any step t, given the partial transcript $\Pi_{<t}$, some bits of S are totally determined, and the remaining ones are still uniformly distributed. The claim is obvious for $t = 0$. Now suppose that the claim holds for some $t - 1 \geq 0$. The next query $Q_t := (s;\sigma)$ is independent of S given $\Pi_{<t}$. Let us fix $\Pi_{<t}$ and $(s;\sigma)$, and implicitly condition on them until stated otherwise. We differentiate two cases.

Case 1: *The oracle answer is* EQUAL. This is the case if and only if $s_\ell = S_{\sigma(\ell)}$ for all $\ell \in [|s|]$. Thus the oracle answer reveals the bits $\{S_{\sigma(\ell)} \mid \ell \in [|s|]\}$, actually determining them.

Case 2: *The oracle answer is k.* This is the case if and only if $s_j = S_{\sigma(j)}$ for all $j < k$ and $s_k \neq S_{\sigma(k)}$. Thus the oracle answer reveals $\{S_{\sigma(\ell)} \mid \ell \in [k]\}$ (the k-th bit by flipping), determining them.

In both cases, the answer is independent of the other bits, therefore the ones among them, which are not determined by previous oracle answers, remain uniformly distributed and mutually independent. This establishes the claim for Π_t, finishing the induction.

We extend the analysis to estimate the mutual information of S and the oracle answer A_t. We keep $\Pi_{<t}$ and Q_t fixed, and implicitly assume $T \geq t$, as otherwise Q_t and A_t don't exist. For readability, we drop the conditions in the computations below; all quantities are to be considered conditioned on $\Pi_{<t}, Q_t$ provided $T \geq t$.

Let $m := \mathbb{H}[S]$ be the number of undetermined bits just before query t. Let K be the number of additionally determined bits due to query Q_t and oracle answer A_t, hence obviously

$$\mathbb{H}[S \mid A_t] = \mathbb{E}[m - K].$$

The analysis above shows that for all $k \geq 1$, a necessary condition for $K \geq k$ is that $s_j = S_{\sigma(j)}$ for the $k-1$ smallest j with $S_{\sigma(j)}$ not determined before query t and that these $k-1$ smallest j really exist. The probability of this condition is $1/2^{k-1}$ (or 0 if there are not sufficiently many j) and so in any case we have

$$\mathbb{P}[K \geq k] \leq \frac{1}{2^{k-1}}, \qquad k \geq 1.$$

Combining these statements we see that,

$$\mathbb{I}[S; A_t] = \mathbb{H}[S] - \mathbb{H}[S \mid A_t] = m - \mathbb{E}[m-K] = \mathbb{E}[K] = \sum_{i \in [m]} \mathbb{P}[K \geq i] \leq \sum_{i \in [\infty]} \frac{1}{2^{i-1}} = 2,$$

with $\Pi_{<t}, Q_t$ still fixed.

Now we re-add the conditionals, vary $\Pi_{<t}, Q_t$, and take expectation still assuming $T \geq t$, obtaining

$$\mathbb{I}[S; A_t \mid \Pi_{<t}, Q_t, T \geq t] \leq 2$$

where T is the number of queries. By Lemma 3.1 applies we obtain $\mathbb{E}[T] \geq [(1-P_e)M - \mathbb{H}[P_e]]/2 \geq [(1-P_e)M - 1]/2$ (the binary entropy is upper bounded by 1) and $\mathbb{P}[T = \Omega(M)] = 1 - P_e - o(1)$, as claimed. □

3.3 Oracle emulation

In this section we introduce *oracle emulation*, which is a special type of reduction from one oracle to another, both for the same family of instances. This reduction allows to transform algorithms based on one oracle to the other preserving their oracle complexity, i.e, the number of queries asked. The crucial result is Lemma 3.5, which we will apply to emulations of various convex optimization oracles by the String Guessing Oracle $\mathcal{O}_S$.

Definition 3.4 (Oracle emulation). Let $\mathcal{O}_1 \colon Q_1 \to R_1$ and $\mathcal{O}_2 \colon Q_2 \to R_2$ be two oracles for the same problem. An *emulation* of $\mathcal{O}_1$ by $\mathcal{O}_2$ consists of

(i) a query emulation function $q \colon Q_1 \to Q_2$ (translating queries of $\mathcal{O}_1$ for $\mathcal{O}_2$),

(ii) an answer emulation function $a \colon Q_1 \times R_2 \to R_1$ (translating answers back) such that $\mathcal{O}_1(x) = a(x, \mathcal{O}_2(q(x)))$ for all $x \in Q_1$.

An emulation leads to a reduction, since emulating oracles are at least as complex as the oracles they emulate.

Lemma 3.5. *If there is an emulation of $\mathcal{O}_1$ by $\mathcal{O}_2$, then the oracle complexity of $\mathcal{O}_1$ is at least that of $\mathcal{O}_2$. Here oracle complexity can be worst-case, randomized, distributional, and high probability; all even for bounded-error algorithms.*

Proof. Let A_1 be an algorithm using $\mathcal{O}_1$, and let $\mathcal{O}_2$ emulate $\mathcal{O}_1$. Let q and a be the query emulation function and the answer emulation function, respectively. We define an algorithm A_2 for $\mathcal{O}_2$ simulating A_1 as follows: Whenever A_1 asks a query x to oracle $\mathcal{O}_1$, oracle $\mathcal{O}_2$ is queried with $q(x)$, and the simulated A_1 receives as answer $a(x, \mathcal{O}_2(q(x)))$ (which is $\mathcal{O}_1(x)$ by definition of the emulation). Finally, the return value of the simulated A_1 is returned.

Obviously, A_2 makes the same number of queries as A_1 for every input, and therefore the two algorithms have the same oracle complexity. This proves that the oracle complexity of $\mathcal{O}_1$ is at least that of $\mathcal{O}_2$. □

4 Single-coordinate oracle complexity for the box

In the following we will analyze a simple class of oracles, called 'single-coordinate', closely mimicking the string guessing oracle. Later, all results will be carried over to general local oracles via perturbation in Section 6.

Definition 4.1 (Single-coordinate oracle)**.** A first-order oracle $\tilde{\mathcal{O}}$ is *single-coordinate* if for all $x \in X$ the subgradient $\nabla f(x)$ in its answer is the one supported on the least coordinate axis; i.e., $\nabla f(x) = \lambda e_i$ for the smallest $1 \leq i \leq n$ with some $\lambda \in \mathbb{R}$.

Choosing the smallest possible i corresponds to choosing the first wrong bit by the String Guessing Oracle. Not all function families possess a single-coordinate oracle, but maximum of coordinate functions do, and single-coordinate oracles are a natural choice for them. From now on, we will denote single-coordinate oracles exclusively by $\tilde{\mathcal{O}}$.

We establish a lower bound on the *distributional* and *high probability oracle complexity* for nonsmooth convex optimization over $[-R, +R]^n$, for single-coordinate oracles.

Theorem 4.2. *Let $L, R > 0$. There exists a finite family $\mathcal{F}$ of Lipschitz continuous convex functions on the L^∞-ball $B_\infty(0, R)$ with Lipschitz constant L in the L^∞ norm, and a single-coordinate local oracle $\tilde{\mathcal{O}}$, such that both the distributional and the high-probability oracle complexity for finding an ε-minimum of a uniformly random instance is $\Omega\left(n \log \frac{LR}{\varepsilon}\right)$.*

For bounded-error algorithms with error bound P_e, the distributional complexity is $\Omega\left((1 - P_e) n \log \frac{LR}{\varepsilon}\right)$, and the high-probability complexity is $\Omega\left(n \log \frac{LR}{\varepsilon}\right)$.

In the following we will restrict ourselves to the case $L = R = 1$, as the theorem reduces to it via an easy scaling argument. We start with the one dimensional case in Section 4.1 for a simpler presentation of the main ideas. We generalize to multiple dimensions in Section 4.2 by considering maxima of coordinate functions, thereby using the different coordinates to represent different portions of a string.

4.1 One dimensional case

Let $X := [-1, 1]$, we define recursively a function family $\mathcal{F}$ on X, which is inspired by the one in [Nemirovski, 1994, Lemma 1.1.1]. For an interval $I = [a, b]$, let $I(t) := a + (1+t)(b-a)/2$ denote the t-point on I for $-1 \le t \le 1$, e.g., $I(-1)$ is the left end point of I, and $I(+1)$ is the right end point, and $I(0)$ is the midpoint. Let $I[t_1, t_2]$ denote the subinterval $[I(t_1), I(t_2)]$. The family $\mathcal{F} = \{f_s\}_s$ will be indexed by binary strings s of length M, where $M \in \mathbb{N}$ depends on the accuracy and will be chosen later. It is convenient to define f_s also for shorter strings, as we proceed by recursion on the length of s. We also define intervals I_s and breakpoints b_l of the range of the functions satisfying the following properties:

(F-1) The interval I_s has length $2 \cdot (1/4)^{|s|}$.
Motivation: allow a strictly nesting family.

(F-2) If $s \parallel t$, then $\operatorname{int}(I_s) \cap \operatorname{int}(I_t) = \emptyset$. If $t \sqsubseteq s$, we have $I_s \subseteq I_t$ (the I_s are nested intervals).
Motivation: instances can be distinguished by its associated intervals. Captures packing property.

(F-3) $f_s \ge f_{s|_l}$ with $f_s(x) = f_{s|_l}(x)$ if $x \in [-1, 1] \setminus \operatorname{int}\left(I_{s|_l}\right)$.
Motivation: long prefix determines much of the function.

(F-4) The function f_s restricted to the interval I_s is of the form

$$f_s(x) = b_{|s|} - 2^{-3|s|} + 2^{-|s|}|x - I_s(0)| \qquad x \in I_s,$$

where $b_{|s|} = f_s(I_s(-1)) = f_s(I_s(+1))$ is the function value on the endpoints of I_s. This is symmetric on I_s as $I_s(0)$ is the midpoint of I_s.
Motivation: recursive structure: repeat absolute value function on small intervals.

(F-5) For $t \sqsubseteq s$, we have $f_s(x) < b_{|t|}$ if and only if $x \in \operatorname{int}(I_t)$.
Motivation: level sets encode substrings.

Construction of the function family

We start with the empty string $\perp$, and define $f_\perp(x) := |x|$ and $I_\perp := [-1, 1]$. In particular, $b_0 = 1$. The further b_k we define via the recursion $b_{k+1} := b_k - 2 \cdot (1/4)^{k+1} \cdot 2^{-k}$.

Given f_s and I_s, we define f_{s0} and I_{s0} to be the *right modification* of f_s via

$$I_{s0} := I_s\left[-\frac{1}{2}, 0\right]$$

$$f_{s0}(x) := \begin{cases} b_{|s|+1} - 2^{-3(|s|+1)} + 2^{-|s|-1}\left|x - I_s\left(-\frac{1}{4}\right)\right|, & \text{if } x \in I_s\left[-\frac{1}{2}, 1\right] \\ f_s(x), & \text{otherwise.} \end{cases}$$

Similarly, the *left modification* f_{s1} of f_s is the reflection of f_{s0} with respect to $I_s(0)$, and I_{s1} is the reflection of I_{s0} with respect to $I_s(0)$, i.e.,

$$I_{s1} := I_s\left[0, \frac{1}{2}\right]$$

$$f_{s1}(x) := \begin{cases} b_{|s|+1} - 2^{-(3|s|+1)} + 2^{-|s|-1}\left|x - I_s\left(\frac{1}{4}\right)\right|, & \text{if } x \in I_s\left[-1, \frac{1}{2}\right] \\ f_s(x), & \text{otherwise.} \end{cases}$$

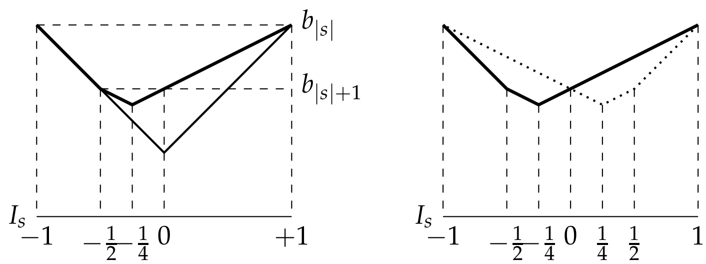

Figure 2: Right modification on the left side: the solid normal line is before the modification, the solid thick line after it. On the right side: right modification is the solid thick line; left modification is the dotted line.

Observe that $I_{s0}, I_{s1} \subseteq I_s$ and $\operatorname{int}(I_{s0}) \cap \operatorname{int}(I_{s1}) = \emptyset$.

This finishes the definition of the f_s. Clearly, these functions are convex and Lipschitz continuous with Lipschitz constant 1, satisfying (F-1)–(F-5).

We establish the packing property for $\mathcal{F}$.

Lemma 4.3. *The family $\mathcal{F}$ satisfies the packing property for $M = \lfloor (1/3)\log(1/\varepsilon) \rfloor$.*

Proof. Note that f_S has its minimum at the midpoint of I_S, and the function value at the endpoints of I_S are at least $(1/2)^{3M} \geq \varepsilon$ larger than the value at the midpoint. Therefore every ε-optimal solution lies in the interior of I_S, i.e., $\mathcal{S}_\varepsilon(f_S) \subseteq \operatorname{int}(I_S)$. Therefore by (F-2), the $\mathcal{S}_\varepsilon(f_S)$ are pairwise disjoint. □

In the following $F \in \mathcal{F}$ will be an instance picked uniformly at random. The random variable S will be the associated string of length M so that $F = f_S$ and S is also distributed uniformly.

Reduction to the String Guessing Problem

We will now provide an oracle for family $\mathcal{F}$ that can be emulated by the String Guessing Oracle. As a first step, we relate the query point x with the indexing strings of the functions. At a high level, the lemma below shows the existence of a prefix of the unknown string determining most of the local behavior of the function at a given query point. From this we will prove in Lemma 4.5 that the oracle answer only reveals this prefix.

Lemma 4.4. *Let $x \in [-1, +1]$ be a query point. Then there is a non-empty binary string s with $l := |s| \leq M$ with the following properties.*

(i) $f_{s^{\oplus(1)}}(x) \geq b_1 > f_{s^{\oplus(2)}}(x) \geq \cdots \geq b_{l-1} > f_{s^{\oplus(l)}}(x) \geq f_s(x)$. *If $l < M$ then also $f_s(x) \geq b_l$.*

(ii) Every binary string t of length M has a unique prefix p from $\{s^{\oplus(1)}, \ldots, s^{\oplus(l)}, s\}$. Moreover, $f_t(x) = f_p(x)$.

Proof. Let s_0 be the longest binary string of length less than M, such that x lies in the interior of I_{s_0}. We choose s to be the one of the two extensions of s_0 by 1 bit, for which f_s has the smaller function value at x (if the two values are equal, then either extension will do). Let $l := |s|$, thus $f_{s^{\oplus(l)}}(x) \geq f_s(x)$.

Note that by the choice of s_0, the point x is not an interior point of I_s unless $l = M$. By (F-2), the point x is neither an interior point of any of the $I_{s^{\oplus(1)}}, \ldots, I_{s^{\oplus(l)}}$.

To prove (ii), let t be any binary string of length M. The existence and uniqueness of a prefix p of t from the set $\{s^{\oplus(1)}, \ldots, s^{\oplus(l)}, s\}$ is clear. In particular, unless $p = t = s$ and $l = M$, the point x is not an interior point of I_p, hence $f_t(x) = f_p(x)$ follows from (F-3). When $p = t$, then $f_t(x) = f_p(x)$ is obviously true.

Now we prove (i). Recall that $f_{s^{\oplus(l)}}(x) \geq f_s(x)$ by the choice of s. First, if $l < M$ then $x \notin \operatorname{int}(I_s)$ by choice, hence $f_s(x) \geq b_l$ by (F-5). Second, let us prove $f_{s^{\oplus(i)}}(x) \geq b_i > f_{s^{\oplus(i+1)}}(x)$ for all $i \leq l$. As $x \notin \operatorname{int}(I_{s^{\oplus(i)}})$, by (F-5) we have $f_{s^{\oplus(i)}}(x) \geq b_i$. Finally, since $x \in \operatorname{int}\left(I_{s|_i}\right)$ and $s|_i \sqsubseteq s^{\oplus(i+1)}$, again by (F-5) we get $f_{s^{\oplus(i+1)}}(x) < b_i$. □

Our construction of instances encodes prefixes in level sets of the instance. The previous lemma indicates that algorithms in this case need to identify a random string, where the oracle reveals prefixes of such string. The following lemma formally shows an emulation by the String Guessing Oracle.

Lemma 4.5. *There is a single-coordinate local oracle $\tilde{\mathcal{O}}$ for the family $\mathcal{F}$ above, which is emulated by the String Guessing Oracle $\mathcal{O}_S$ on strings of length M.*

Proof. We define the emulation functions first, as they determine the emulated oracle $\tilde{\mathcal{O}}$. Let $x \in [-1, 1]$ and s the string from Lemma 4.4. We define the query emulation function as $q(x) := (s, \mathrm{id})$. Moreover, let $l = |s|$.

Now we need to emulate the oracle answer. From Lemma 4.4 (ii) there exists a prefix P of S such that $f_S(x) = f_P(x)$. We define the following function p of the $\mathcal{O}_S$ oracle answer

$$p(x, \texttt{EQUAL}) := s,$$
$$p(x, k) := s^{\oplus(k)}.$$

Note that $P = p(x, \mathcal{O}_S(q(x))$. We claim that p depends on f_S only locally around x. First, if $f_s(x) < f_{s^{\oplus(l)}}(x)$ then by Lemma 4.4 (i) $f_S(x)$ determines P (and thus p). Otherwise, depending on whether f_S is increasing or decreasing around x, we can determine if $P_l = s_l$.

Since $f_S(x) = f_P(x)$ and $f_S \geq f_P$, a valid oracle answer to the query point x is $f_P(x)$ as function value and a subgradient $\nabla f_P(x)$ of f_P at x as $\nabla f_S(x)$. Therefore we define the answer emulation as $a(x, R) := (f_{p(x,R)}(x), \nabla f_{p(x,R)}(x))$. This provides a single-coordinate local oracle $\tilde{\mathcal{O}}$ for the family $\mathcal{F}$ (the single-coordinate condition is trivially satisfied when $n = 1$) that can be emulated by the String Guessing Oracle $\mathcal{O}_S$. □

The previous lemma together with Lemma 3.5 leads to a straightforward proof of Theorem 4.2 in the one dimensional case.

Proof of Theorem 4.2 for $n = 1$. Let A be a black box optimization algorithm for $\mathcal{F}$ accessing the oracle $\tilde{\mathcal{O}}$. As $\mathcal{F}$ satisfies the packing property by Lemma 4.3, in order to find an ε-minimum the algorithm A has to identify the string s defining the function $f = f_s$ (and from an ε-minimum the string s can be recovered).

Let $F = f_S$ be the random instance chosen with uniform distribution. Together with the emulation defined in Lemma 4.5, algorithm A solves the String Guessing Problem for strings of length M, hence requiring at least $[(1 - P_e)M - 1]/2$ queries in expectation with error probability at most P_e by Proposition 3.3. Moreover, with probability $1 - P_e - o(1)$, the number of queries is at least $\Omega(M)$. This proves the theorem for $n = 1$ by the choice of M. □

4.2 Multidimensional case

Construction of function family

In the general n-dimensional case the main difference is using a larger indexing string. Therefore we choose $M = \lfloor (1/3) \log(1/\varepsilon) \rfloor$, and consider n-tuples $s_1, \ldots, s_n$ of binary strings of length M as indexing set for the function family $\mathcal{F}$, and define the member functions via

$$f_{s_1,\ldots,s_n}(x_1,\ldots,x_n) := \max_{i\in[n]} f_{s_i}(x_i), \tag{4}$$

where the f_{s_i} are the functions from the one dimensional case. This way, the size of $\mathcal{F}$ is 2^{nM}. Note that as the f_{s_i} are 1-Lipschitz, the $f_{s_1,\ldots,s_n}$ are 1-Lipschitz in the L^∞ norm, too. We prove that $\mathcal{F}$ satisfies the packing property.

Lemma 4.6. *The family $\mathcal{F}$ above satisfies the packing property for $M = \lfloor (1/3)\log(1/\varepsilon) \rfloor$.*

Proof. As the minimum values of all the one dimensional f_{s_i} coincide, obviously the set of ε-minima of $f_{s_1,\ldots,s_n}$ is the product of its components:

$$\mathcal{S}_\varepsilon(f_{s_1,\ldots,s_n}) = \prod_{i\in[n]} \mathcal{S}_\varepsilon(f_{s_i}).$$

Hence the claim reduces to the one dimensional case, proved in Lemma 4.3. ☐

Let $S = (S_1,\ldots,S_n)$ denote the tuple of strings indexing the actual instance, hence the S_i are mutually independent uniform binary strings; and let $F = f_{S_1,\ldots,S_n}$.

Reduction to the String Guessing Problem

We argue as in the one dimensional case, but now the string for the String Guessing Oracle is the concatenation of the strings $S_1,\ldots,S_n$, and therefore has length nM.

Lemma 4.7. *There is a single-coordinate oracle $\tilde{\mathcal{O}}$ for family $\mathcal{F}$ that can be emulated by the String Guessing Oracle $\mathcal{O}_S$ with associated string the concatenation of the $S_1,\ldots,S_n$.*

Before proving the result, let us motivate our choice for the first-order oracle. The general case arises from an interleaving of the case $n = 1$. As we have seen in the proof of Lemma 4.5, for $n = 1$ querying the first-order oracle leads to querying prefixes. By (F-3), if S is the string defining the function f_S, then for any prefix S' of S we have $f_{S'} \le f_S$; this gives a lower bound on the unknown instance. By querying a point x we obtain such a prefix with the additional property $f_{S'}(x) = f_S(x)$, which localizes the minimizer in an interval, and thus provides an upper bound on its value.

Now, for general n we want to upper bound the maximum as well by prefixes of the hidden strings. In particular, there is no use to querying any potential prefixes u for coordinate i such that $f_u(x_i)$ is strictly smaller than the candidate maximum; they are not revealed by the oracle.

The query string for the String Guessing Oracle now arises by interleaving the query strings for each coordinate. In particular, if we restrict the query string to the substring consisting only of prefixes for a specific coordinate i, then these substrings should be ordered by $\sqsubseteq$, which is precisely the ordering we used for the case $n = 1$ as a necessary condition. Thus, a natural way of interleaving these query strings is by their objective function value. Moreover, refining this order by the lexicographic order on coordinates will induce a single-coordinate oracle.

Proof. Let $x = (x_1,\ldots,x_n)$ be a query point. For a family of strings $\{S_i\}_i$ regard S as their concatenation, and for notational convenience let $S_{i,h}$ denote the h-th bit of S_i. Applying Lemma 4.4 to each coordinate $i \in [n]$, there is a number l_i and a string s_i of length l_i associated to the point x_i.

We define the confidence order $\prec$ of labels (i,h) with $i \in [n]$ and $h \in [l_i]$ as the one induced by the lexicographic order on the pairs $(-f_{s_i^{\oplus(h)}}(x_i), i)$ i.e.,

$$(i_1,h_1) \prec (i_2,h_2) \iff \begin{cases} f_{s_{i_1}^{\oplus(h_1)}}(x_{i_1}) > f_{s_{i_2}^{\oplus(h_2)}}(x_{i_2}) & \text{or} \\ f_{s_{i_1}^{\oplus(h_1)}}(x_{i_1}) = f_{s_{i_2}^{\oplus(h_2)}}(x_{i_2}) \ \wedge\ i_1 \leq i_2. \end{cases}$$

We restrict to the labels (i,h) with $f_{s_i^{\oplus(h)}}(x_i) \geq \max_{j\in[n]} f_{s_j}(x_j)$ (there is no use to query the rest of labels, as pointed out above). Let $(i_1,h_1),\ldots,(i_k,h_k)$ be the sequence of these labels in $\prec$-increasing confidence order. Let t be the string of length k with $t_m = s_{i_m,h_m}$ for all $m \in [k]$. We define the query emulation as $q(x) = (t,\sigma)$ with $\sigma_m := (i_m,h_m)$.

Now, when queried with this string the String Guessing Oracle returns the index of the first mismatch. This string corresponds to a prefix of S (in the order given by $\prec$). To be precise, we define a coordinate j and a prefix p as helper functions in x and the oracle answer for the answer emulation a (with the intent of having $f_S(x) = f_p(x_j)$ and p a prefix of S_j). If the oracle answer is `EQUAL`, then we choose $j = i_k$, and set $p := s_j|_{h_k}$. If the oracle answer is a number m then we set $j := i_m$ and $p := s_{i_m}^{\oplus(h_m)}$.

Analogously as in the proof of Lemma 4.5, both p and j depend only on x and on the local behavior of f_S around x. Moreover, it is easy to see that $f_S(x) = f_p(x_j)$ and $f_S(y) \geq f_{S_j}(y_j) \geq f_p(y_j)$for all y, which means that $\nabla f_p(x_j)e_j$ is a subgradient of f_S at x.

We now define the answer emulation

$$a(x,R) = (f_{p(x,R)}(x_{j(x,R)}), \nabla f_{p(x,R)}(x_{j(x,R)})e_{j(x,R)}),$$

and thus the oracle $\tilde{\mathcal{O}}(x) = a(x, \mathcal{O}_S(q(x)))$ is a first-order local oracle for the family $\mathcal{F}$ that can be emulated by the String Guessing Oracle. Finally, the single-coordinate condition is satisfied from the confidence order of the queries, which proves our result. ☐

We are ready to prove Theorem 4.2.

Proof of Theorem 4.2. The proof is analogous to the case $n = 1$. However, by the emulation via Lemma 4.7 we solve the String Guessing Problem for strings of length nM. Thus by Proposition 3.3 we obtain the claimed bounds the same way as in the case $n = 1$. ☐

5 Single-coordinate oracle complexity for L^p-Balls

In this section we examine the complexity of convex nonsmooth optimization on the unit ball $B_p(0,1)$ in the L^p norm for $1 \leq p < \infty$. Again, we restrict our analysis to the case of single-coordinate oracles. We distinguish the large-scale case (i.e., $\varepsilon \geq 1/n^{\max\{p,2\}}$), and low-scale case (i.e., $\varepsilon \leq n^{-1/\max\{p,2\}-\delta}$, for fixed $\delta > 0$).

5.1 Large-scale case

Theorem 5.1. *Let $1 \leq p < \infty$ and $\varepsilon \geq 1/\sqrt[p]{n}$. There exists a finite family $\mathcal{F}$ of convex Lipschitz continuous functions in the L^p norm with Lipschitz constant 1 on the n-dimensional unit ball $B_p(0,1)$, and a single-coordinate local oracle $\tilde{\mathcal{O}}$ for $\mathcal{F}$, such that both the distributional and the high-probability oracle complexity of finding an ε-minimum under the uniform distribution are $\Omega\left(1/\varepsilon^{\max\{p,2\}}\right)$.*

For bounded-error algorithms with error probability at most P_e, the distributional complexity is $\Omega\left((1-P_e)/\varepsilon^{\max\{p,2\}}\right)$, while the high probability complexity is $\Omega\left(1/\varepsilon^{\max\{p,2\}}\right)$.

Remark 5.2 (The case $p = 1$). For $p = 1$, the lower bound can be improved to $\Omega\left(\frac{\ln n}{\varepsilon^2}\right)$ by a nice probabilistic argument, see [Nemirovski and Yudin, 1983, Section 4.4.5.2].

As in the previous section, we will construct a single-coordinate oracle that can be emulated by the String Guessing Oracle. As the lower bound does not depend on the dimension, we shall restrict our attention to the first $M = \Omega(1/\varepsilon^{\max\{p,2\}})$ coordinates. For these coordinates, it will be convenient to work in an orthogonal basis of vectors with maximal ratio of L^p norm and L^2 norm, to efficiently pack functions in the L^p-ball. For $p \geq 2$ the standard basis vectors e_i already have maximal ratio, but for $p < 2$ it requires a basis of vectors with all coordinates of all vectors being ± 1, see Figure 3. In particular, in our working basis the L^p norm might look different than in the standard basis. We shall present the two cases uniformly, keeping the differences to a bare minimum.

Figure 3: Unit vectors of maximal L^p norm together with the unit Euclidean ball in gray and the unit L^p-ball in black.

The exact setup is as follows. Let $r := \max\{p, 2\}$ for simplicity. We define M and the working basis for the first M coordinates, such that the coordinates as functions will have Lipschitz constant at most 1.

Case 1: $2 \leq p < \infty$. We let $M := \left\lfloor \frac{1}{\varepsilon^p} \right\rfloor - 1$. The working basis is chosen to be the standard basis.

Case 2: $1 \leq p < 2$. Let l be the largest integer with $1/\varepsilon^2 > 2^l$, and define $M := 2^l$. Since $\varepsilon \geq 1/n^2$, obviously $M < 1/\varepsilon^2 \leq n$. In the standard basis, the space $\mathbb{R}^2$ has an orthogonal basis of ± 1 vectors, e.g., $(1, 1)$ and $(1, -1)$. Taking l-fold tensor power, we obtain an orthogonal basis of $\mathbb{R}^M$ consisting of ± 1 vectors v_i in the standard basis. We shall work in the scaled orthogonal basis $\xi_i := v_i / \sqrt[q]{M}$, where q is chosen such that $1/p + 1/q = 1$. Note that the coordinate functions have Lipschitz constant at most 1, as $\langle \xi_i, x \rangle \leq \|\xi_i\|_q \|x\|_p$ for all x, and $\|\xi_i\|_q = 1$.

Clearly in both cases, $M \leq n$ and $M = \Omega(1/\varepsilon^r)$, but $M < 1/\varepsilon^r$. From now on, we shall use $\|\cdot\|_p$ for the p norm in the original basis, and $\|\cdot\|_2$ for the 2 norm in the working basis. Note that $\|x\|_p \leq \|x\|_2$ if $p < 2$.

Construction of function family

We define our functions $f_s : B_p(0, 1) \to \mathbb{R}$ as maximum of (linear) coordinate functions:

$$f_s(x) = \max_{i \in [M]} s_i x_i, \tag{5}$$

where the x_i are the coordinates of x in our working basis.

We parameterize the family $\mathcal{F} = \{f_s : s \in \{-1, +1\}^M\}$ via sequences $s = (s_1, \ldots, s_M)$ of signs ± 1 of length M. By the above this family satisfies the requirements of Theorem 5.1. We establish the packing property for $\mathcal{F}$.

Lemma 5.3. *The family $\mathcal{F}$ satisfies the packing property.*

Proof. Let $x = (x_1, \dots, x_n)$ be an ε-minimum of f_s. We compare it with

$$x^* := \left(-\frac{s_1}{\sqrt[r]{M}}, \dots, -\frac{s_M}{\sqrt[r]{M}}, 0, \dots, 0\right).$$

Recall that $r = \max\{p, 2\}$. The vector x^* lies in the unit L^p-ball. This is obvious for $p \geq 2$, while for $p < 2$ this follows from $\|x^*\|_p \leq \|x^*\|_2 = 1$.

Therefore, as $M < 1/\varepsilon^r$, we obtain for all $i \in [M]$

$$s_i x_i \leq f_s(x_1, \dots, x_n) \leq f_s^* + \varepsilon \leq f_s(x^*) + \varepsilon = -\frac{1}{\sqrt[r]{M}} + \varepsilon < 0,$$

i.e., $s_i = -\operatorname{sgn} x_i$. Hence every ε-minimum x uniquely determines s, proving the packing property. □

Let $F \in \mathcal{F}$ be chosen uniformly at random, and let S be the associated string of length M so that $F = f_S$ and thus $S \in \{-1, +1\}^M$ is uniformly distributed.

Reduction to the String Guessing Problem

The main idea is that the algorithm learns solely some entries S_i of the string S from an oracle answer.

Lemma 5.4. *There is a single-coordinate local oracle $\tilde{\mathcal{O}}$ that can be emulated by the String Guessing Oracle $\mathcal{O}_S$.*

Proof. To better suit the present problem, we now use ± 1 for the values of bits of strings.

Given a query x, we introduce an ordering $\prec$ on the set of coordinates $\{1, 2, \dots, M\}$: we map each coordinate i to the pair $(-|x_i|, i)$, and take the lexicographic order on these pairs, i.e.,

$$i_1 \prec i_2 \iff \begin{cases} |x_{i_1}| > |x_{i_2}| & \text{or} \\ |x_{i_1}| = |x_{i_2}| \ \wedge \ i_1 \leq i_2. \end{cases}$$

Let $\sigma(1), \dots, \sigma(k)$ be the indices $i \in [M]$ put into $\prec$-increasing order with k the minimum between M and the $\prec$-first i s.t. $x_i = 0$. Let s be the string of length k with $s_j = -\operatorname{sgn} x_{\sigma(j)}$. If $x_{\sigma(k)} = 0$, we put $s_k = +1$. (The value -1 would also do.) The query emulation q is defined via $q(x) := (s, \sigma)$.

We now define helper functions J and p in x and a query of $\mathcal{O}_S$. We set

$$J(x, \texttt{EQUAL}) := k, \qquad p(x, \texttt{EQUAL}) := s_k, \qquad J(x, j) := j, \qquad p(x, j) := -s_j.$$

For the remainder of the proof we drop the arguments of these functions and simply write J and p instead of $J(x, \mathcal{O}_S(q(x)))$ and $p(x, \mathcal{O}_S(q(x)))$, respectively to ease readability.

Actually, J is the $\prec$-smallest index j with $f_S(x) = S_j x_j$. If $j \neq \sigma(k)$ then $p = S_j$; in the case $J = \sigma(k)$, the value of p is $+1$ if f_S is partially locally increasing in x in the J-th coordinate, and it is -1 if it is decreasing. In other words, J and p are local. Moreover, $f_S(x) = p x_J$ and $f_S(y) \geq p y_J$ for all y, therefore $p e_J$ is a subgradient of f_S at x.

We define the query emulation a via $a(x, R) := (p(x, R) x_{J(x,R)}, p(x, R) e_{J(x,R)})$. Oracle $\tilde{\mathcal{O}}$ is defined by the emulation $\tilde{\mathcal{O}}(x) = a(x, \mathcal{O}_S(q(x)))$, which is clearly single-coordinate. Thus $\tilde{\mathcal{O}}(x)$ is a valid answer to query x. □

We are ready to prove Theorem 5.1

Proof of Theorem 5.1. The proof is analogous to the proof of Theorem 4.2. Given the oracle $\mathcal{O}$ in Lemma 5.4, every black box algorithm A having access to this oracle solves the String Guessing Problem for strings of length $M = \Theta(1/\varepsilon^{\max\{p,2\}})$ using the String Guessing Oracle only. Hence the claimed lower bounds are obtained by Proposition 3.3. □

5.2 The low-scale case: reduction to the box case

We show that for small accuracies, the L^p-ball lower bound follows from Theorem 4.2. Before we establish this result, let us observe that for technical reasons the optimal lower bound when $1 \leq p < 2$ will be postponed until Section 6.

Proposition 5.5. *Let $1 \leq p < \infty$, and $\varepsilon \leq n^{-\frac{1}{p}-\delta}$ with $\delta > 0$. There exists a family $\mathcal{F}$ of convex Lipschitz continuous functions in the L^p norm with Lipschitz constant 1 on the n-dimensional unit Euclidean ball $B_p(0,1)$, and a single-coordinate oracle for family $\mathcal{F}$, such that both the distributional and the high-probability oracle complexity of finding an ε-minimum under the uniform distribution is $\Omega\left(n \log \frac{1}{\varepsilon}\right)$.*

For algorithms with error probability at most P_e, the distributional complexity is $\Omega\left((1-P_e) n \log \frac{1}{\varepsilon}\right)$ and the high probability complexity is $\Omega\left(n \log \frac{1}{\varepsilon}\right)$.

Proof. The proof is based on a rescaling argument.

We have $[-\frac{1}{\sqrt[p]{n}}, \frac{1}{\sqrt[p]{n}}]^n \subseteq B_p(0,1)$ and thus by Theorem 4.2 there exists a family of convex Lipschitz continuous functions with Lipschitz constant 1 (in the L^∞ norm, therefore also in the L^p norm), and a single-coordinate oracle for $\mathcal{F}$, with both distributional oracle complexity and high-probability oracle complexity $\Omega\left(n \log \frac{1}{\varepsilon \sqrt[p]{n}}\right) = \Omega\left(n \log \frac{1}{\varepsilon}\right)$ for large n, where the last equality follows from the fact that for $\varepsilon \leq n^{-1/p-\delta}$ with $\delta > 0$ we have $\varepsilon \sqrt[p]{n} \leq \varepsilon^{\frac{\delta}{1/p+\delta}}$.

□

For the case of the L^p-ball with $1 \leq p < \infty$, we thus close the gap exhibited in Figure 1 for arbitrary small but fixed $\delta > 0$.

Remark 5.6 (Understanding the dimensionless speed up in terms of entropy). The observed (dimensionless) performance for the L^p-ball, for $2 \leq p < \infty$, has a nice interpretation when comparing the total entropy of the function families. Whereas in the unit box we could pack up to roughly $2^{n \log \frac{1}{\varepsilon}}$ instances with nonintersecting ε-solutions, we can only pack roughly $2^{1/\varepsilon^p}$ into the L^p-ball. This drop in entropy alone can explain the observed speed up.

We give some intuition by comparing the volume of the unit box with the volume of the inscribed unit L^p-ball. Suppose that there are $K_n \approx 2^{n \log 1/\varepsilon}$ 'equidistantly' packed instances in the box; this number is roughly the size of the function family used above. Intersecting with

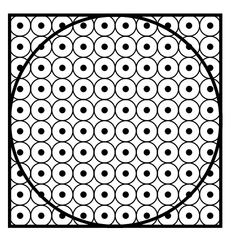

Figure 4: Equidistantly packed points with a neighbourhood in a ball and a box. The number of points in each is proportional to its volume.

the L^p-ball, see Figure 4 for an illustration, we end up with roughly $K_n V_n$ instances, where $V_n = (2\Gamma(1/p+1))^n / \Gamma(n/p+1)$ is the volume of the unit ball. For the boundary case $\varepsilon = 1/\sqrt[p]{n}$:

$$\begin{aligned}
\mathbb{H}[F] &\approx \log K_n V_n \\
&\approx n \log n^{1/p} + n\left(1 + \frac{1}{p}\log\frac{1}{p} - \frac{1}{p} + \log\sqrt{2\pi}\right) - \left(\frac{n}{p}\log\frac{n}{p} - \frac{n}{p} + \log\sqrt{2\pi}\right) \\
&\approx n\left(1 + \log\sqrt{2\pi}\right) \approx \frac{1 + \log\sqrt{2\pi}}{\varepsilon^p},
\end{aligned}$$

i.e., the entropy of the function family in the ball drops significantly, being in line with the existence of fast methdods in this case.

6 Lower bounds for arbitrary local oracles

We extend our results in Sections 4 and 5 to arbitrary local oracles. The key observation is that for query points where the instance is locally linear any local oracle reduces to the the single-coordinate oracle studied in previous sections. Thus, we can prove lower bounds by perturbing our instances in such a way that we avoid singular[1] query points with probability one.

We present full proofs for expectation (distributional) lower bounds, however observe that lower bounds w.h.p. (and with bounded error) follow analogous arguments by averaging on conditional probabilities, instead of conditional expectations.

Before going into the explicit constructions, let us present a useful tool for analyzing arbitrary local oracles. We will show there exists a *maximal* oracle $\mathcal{O}$ such that any local oracle can be emulated by $\mathcal{O}$. This way, it suffices to show lower bounds on $\mathcal{O}$ to deduce lower bounds for arbitrary local oracles. More precisely, let $\mathcal{O}$ be the oracle answering for query $x \in X$ the family of functions $g \in \mathcal{F}$ such that there exists a neighborhood around x (possibly depending on g) where $f = g$. Clearly $\mathcal{O}$ is a local oracle.

Lemma 6.1 (Maximal Oracle)**.** *Let $\mathcal{F}$ be a finite family of functions. Then the maximal oracle $\mathcal{O}$ is such that any local oracle $\mathcal{O}'$ can be emulated by $\mathcal{O}$*

Proof. Let $\mathcal{O}'$ be any local oracle, and x be a query point. Let the query emulation be the identity. Now, for the answer emulation, by definition, for instances $f, g \in \mathcal{F}$, we have $\mathcal{O}_f(x) = \mathcal{O}_g(x)$ if and only if $f = g$ around x. Therefore the function $a(x, \mathcal{O}_f(x)) = \mathcal{O}'_f(x)$ is well-defined; this defines an oracle emulation of $\mathcal{O}'$ by $\mathcal{O}$, proving the result. ☐

For the rest of the section, let $\tilde{\mathcal{O}}$ be the single-coordinate oracle studied in previous sections, and let $\mathcal{O}$ be the maximal oracle. Note that we state the theorems below for $\mathcal{O}$ an arbitrary local oracle, but from Lemma 6.1 w.l.o.g. we may choose for the proofs $\mathcal{O}$ to be the maximal oracle.

6.1 Large-scale complexity for L^p-Balls

Recall that in Section 5.1, different function families were used for the case $1 \le p < 2$ and $2 \le p < \infty$. However, the proof below is agnostic to which family is used, by following the notation from (5).

Theorem 6.2. *Let $1 \le p < \infty$, $\varepsilon \ge 1/n^{\max\{p,2\}}$ and let $\mathcal{O}$ be an arbitrary local oracle for the family $\mathcal{F}$ of convex Lipschitz continuous functions in the L^p norm with Lipschitz constant 1 on the n-dimensional unit ball $B_p(0,1)$. Then both the distributional and the high-probability oracle complexity of finding an ε-minimum is $\Omega(1/\varepsilon^{\max\{p,2\}})$.*

For bounded-error algorithms with error bound P_e, the distributional complexity is $\Omega\left((1-P_e)/\varepsilon^{\max\{p,2\}}\right)$, and the high probability complexity is $\Omega\left(1/\varepsilon^{\max\{p,2\}}\right)$.

Before proving this Theorem let us introduce the hard function family, which is a perturbed version of the hard instances in Section 5.1.

[1] In our framework, singular points are defined as the ones where a subgradient depends on more than one bit encoding the instance. This coincides with points of nonsmoothness in the large-scale case, but in the box case there is a more subtle property, see Lemma 6.5.

Construction of function family

Let $1 \le p < \infty$, $\varepsilon \ge 1/n^{\max\{p,2\}}$, and $X := B_p(0,1)$. Let M and f_s be defined as in the proof of Theorem 5.1, and $\bar{\delta} := \varepsilon/(KM)$, where $K > 0$ is a constant. Consider the infinite family $\mathcal{F} := \{f_{s,\delta}(x) : s \in \{-1,+1\}^M,\ \delta \in [0,\bar{\delta}]^M\}$, where

$$f_{s,\delta}(x) = f_s(x+\delta).$$

Finally, we consider the random variable $F = f_{S,\Delta}$ on $\mathcal{F}$ where $S \in \{-1,1\}^M$ and $\Delta \in [0,\bar{\delta}]^M$ are chosen independently and uniformly at random.

Proof. The proof requires two steps: first, showing that the subfamily of instances with a fixed perturbation δ is as hard as the unperturbed one for the single-coordinate oracle. Second, by properly averaging over δ we obtain the expectation lower bound.

Lower bound for fixed perturbation under oracle $\tilde{\mathcal{O}}$ Let $\delta \in [0,\bar{\delta}]^M$ be a fixed vector, and $\tilde{\mathcal{F}} = \{f_{s,\delta} \colon s \in \{-1,+1\}^M\}$. Since $f_{s,\delta}(x) = f_s(x+\delta)$, for a fixed perturbation the subfamily of instances is just a re-centering of the unperturbed ones. We claim that the complexity of this family under $\tilde{\mathcal{O}}$ is lower bounded by $\mathbb{E}[T] \ge \frac{M(1-\varepsilon/K)}{2}$.

In fact, consider the ball $B_p(-\delta, r)$, where $r = 1 - \varepsilon/K$. Let $x \in B_p(-\delta, r)$, then

$$\|x\|_p \le \|x+\delta\|_p + M\bar{\delta} \le 1 - \varepsilon/K + \varepsilon/K = 1,$$

so $x \in B_p(0,1)$. Therefore, $B_p(-\delta,r) \subseteq B_p(0,1)$, and thus the complexity of $\tilde{\mathcal{F}}$ over $B_p(0,1)$ can be lower bounded by the complexity of the same family over $B_p(-\delta,r)$ (optimization on a subset is easier in terms of oracle complexity). Now observe that the problem of minimizing $\tilde{\mathcal{F}}$ over $B_p(-\delta,r)$ under $\tilde{\mathcal{O}}$ is equivalent to the problem studied in Section 5.1, only with the radius scaled by r. This re-scaled problem has the same complexity as the original one, only with an extra r factor. Thus,

$$\mathbb{E}[T] \ge \frac{Mr}{2} = \frac{M(1-\varepsilon/K)}{2} \quad \forall \delta \in [0,\bar{\delta}]^M.$$

Lower Bounds for $\mathcal{F}$ under oracle $\mathcal{O}$ To conclude our proof, we need to argue that oracle $\mathcal{O}$ does not provide more information than $\tilde{\mathcal{O}}$ with probability 1. Let A be an algorithm and T the number of queries it requires to determine S (which is a random variable in both S and Δ).

We will show first that throughout its trajectory $(X^1, \ldots, X^T)$, algorithm A queries singular points of $f_{S,\Delta}$ with probability zero. Formally, we have

Lemma 6.3 (on unpredictability, large-scale case)**.** *For an $\mathcal{O}$-based algorithm solving family F with queries $X^1, \ldots, X^T$ we define, for $t \ge 0$, the set of maximizer coordinates as*

$$I^t := \{i \in [M] : S_i(X_i^t + \Delta_i) = f_{S,\Delta}(X^t)\}$$

if $t \le T$, and $I^t = \emptyset$ otherwise, and let us consider the event E where the set of maximizers include at most one new coordinate at each iteration

$$E := \left\{ \left| I^t \setminus \bigcup_{s<t} I^s \right| \le 1, \quad \forall t \le T \right\}.$$

Then $\mathbb{P}[E] = 1$.

Proof. We prove by induction that before every query $t \geq 1$ the set of 'unseen' coordinates $I_c^t := [M] \setminus (\cup_{s<t} I^s)$ is such that perturbations $(\Delta_i)_{i \in I_c^t}$ are absolutely continuous (w.r.t. the Lebesgue measure). Moreover, from this we can prove simultaneously that

$$\mathbb{P}\left[\left| I^t \setminus \bigcup_{s<t} I^s \right| > 1 \,\middle|\, \Pi_{<t} \right] = 0.$$

We start from the base case $t = 0$, which is evident since the distribution on Δ is uniform. Now, since singular points (for all possible realizations of S) lie in a smaller dimensional manifold, then $|I^1| = 1$ almost surely. In the inductive step, suppose the claim holds up to t and consider the $(t+1)$-th query. Then what the transcript provides for coordinates in I_c^{t+1} are upper bounds for the perturbations Δ_i given S_i. In fact, from the $(t+1)$-th oracle answer all we obtain are S_j and Δ_j, where j is such that $f_{S,\Delta}(X^{t+1}) = S_j X_j^{t+1} + \Delta_j$; note that such j is almost surely unique among $j \in I_c^t$, by induction. For the rest of the coordinates $i \neq j$ we implicitly know

$$S_i X_i^{t+1} + \Delta_i \leq S_j X_j^{t+1} + \Delta_j,$$

i.e., $\Delta_i \leq D_{i,+}$ if $S_i = 1$, and $\Delta_i \leq D_{i,-}$ if $S_i = -1$; where $D_{i,\pm}$ are constants depending on $(X^0, \ldots, X^{t+1})$, S_j, Δ_j, but not depending on any of the other unknowns. Thus, at every iteration we obtain for non-maximizer coordinates upper bounds on the perturbation Δ_i, conditionally on the sign of S_i. These bounds are such that $\Delta_i = D_{i,\pm}$ with probability zero, as the distribution on $(\Delta_i)_{i \in I_c^t}$ (conditionally on the transcript), which is the one described above, is absolutely continuous. Moreover, by absolute continuity,

$$\mathbb{P}\left[\left| I^{t+1} \setminus \bigcup_{s \leq t} I^s \right| > 1 \,\middle|\, \Pi_{\leq t} \right] = 0,$$

proving the inductive step.

Finally, by the union bound

$$\mathbb{P}\left[\bar{E}\right] \leq \sum_{t=1}^{M} \mathbb{P}\left[\left| I^{t+1} \setminus \bigcup_{s \leq t} I^s \right| > 1 \right].$$

And from the previous argument,

$$\mathbb{P}\left[\left| I^{t+1} \setminus \bigcup_{s \leq t} I^s \right| > 1 \right] = \mathbb{E}_{\Pi_{\leq t}}\left[\mathbb{P}\left[\left| I^{t+1} \setminus \bigcup_{s \leq t} I^s \right| > 1 \,\middle|\, \Pi_{\leq t} \right] \right] = 0.$$

□

With the Lemma on unpredictability the proof becomes straightforward. We claim that on event E, the oracle answer provided by $\mathcal{O}$ can be emulated by the answer provided by $\tilde{\mathcal{O}}$ on the same point; thus, the trajectory of A is equivalent to the trajectory of some algorithm querying $\tilde{\mathcal{O}}$.

To prove our claim, let $\mathcal{O}$ be the maximal oracle for the family of perturbed instances $\mathcal{F}$. We observe that on event E, oracle $\tilde{\mathcal{O}}$ is as powerful as $\mathcal{O}$, since the oracle answer of $\mathcal{O}$ for instance $f_{s,\delta}$ is the set $\{f_{r,\gamma} \colon r_j = s_j,\ \gamma_j = \delta_j\}$, where j is the unique maximizer coordinate of $f_{s,\delta}$ on x. Note that this oracle answer can be trivially emulated from the answer by $\tilde{\mathcal{O}}$, which is essentially (s_j, δ_j).

By the claim we conclude that for all δ excluding the measure zero set $\bar{E}$, $\mathbb{E}\left[T \,|\, \Delta = \delta\right] \geq \frac{M(1-\varepsilon/K)}{2}$. By averaging over δ we obtain $\mathbb{E}\left[T\right] \geq \frac{M(1-\varepsilon/K)}{2}$. By choosing $K > 0$ arbitrarily large we obtain the desired lower bound. □

6.2 Complexity for the box

For the box case we will first introduce the family construction, which turns out to be slightly more involved than the one in Section 4. Similarly as in the large-scale case, we first analyze the perturbed family for a fixed perturbation under the single-coordinate oracle, and then we prove the Lemma on unpredictability. With this the rest of the proof is analogous to the large-scale case and thus left as an exercise.

Theorem 6.4. *Let $L, R > 0$, and let $\mathcal{O}$ be an arbitrary local oracle for the family $\mathcal{F}$ of Lipschitz continuous convex functions on the L^∞-ball $B_\infty(0, R)$ with Lipschitz constant L in the L^∞ norm. Then both the distributional and the high-probability oracle complexity for finding an ε-minimum is $\Omega\left(n \log \frac{LR}{\varepsilon}\right)$.*

For bounded-error algorithms with error bound P_e, the distributional complexity is $\Omega\left((1 - P_e)n \log \frac{LR}{\varepsilon}\right)$, and the high-probability complexity is $\Omega\left(n \log \frac{LR}{\varepsilon}\right)$.

As in Section 4, w.l.o.g. we prove the Theorem for $L = R = 1$, and recall that w.l.o.g. $\mathcal{O}$ is the maximal oracle.

One dimensional construction of function family

First we define the perturbed instances for the one dimensional family. The multidimensional family will be defined simply as the maximum of one dimensional functions, as in (4).

We will utilize different perturbations for each level (in the recursive definition) of the function. For this reason, in order to preserve convexity, and in order to not reveal the behavior of lower levels through perturbations, we need to patch the perturbations of consecutive levels in a consistent way.

Given $0 < \varepsilon \le 1$, let $M := \lfloor \frac{1}{3 - \ln \alpha} \ln(1/\varepsilon) \rfloor$ and $\bar{\delta} := \frac{1-\alpha}{4} (\frac{\alpha}{8})^M$, where $\alpha := 1 - 8\varepsilon/(5KM)$, and K is a large constant. Note that for K large enough $\alpha > 1/e$, independently of the values $\varepsilon \in (0, 1]$ and $M \ge 1$; this way, we guarantee that $M \ge \lfloor \frac{1}{4} \ln(1/\varepsilon) \rfloor$. Once we have defined our function family we justify our choice for these parameters.

Let us recall from Section 4 the recursive definition of intervals $(I_s)_{s \in \{0,1\}^M}$ and properties (F-1)–(F-5). We will prove there exists a family $\tilde{\mathcal{F}} = \{f_{s,\delta} : [-1, 1] \to \mathbb{R} :\ s \in \{0, 1\}^l,\ 0 < \delta_i \le \bar{\delta},\ i = 1, \ldots, M\}$, satisfying properties (F-1), (F-2), and the analogues of (F-3)–(F-5) described below

(G-3) $f_{s,\delta} \ge f_{s|_l,\delta}$ with $f_{s,\delta}(x) = f_{s|_l,\delta}(x)$ if and only if $x \in [-1, 1] \setminus \operatorname{int}\left(I^\delta_{s|_l}\right)$, where

$$I^\delta_{s|_l} := I_{s|_l}\left[-1 + \left(\frac{2}{\alpha}\right)^l \frac{\delta_{l+1}}{1 - \alpha}, 1 - \left(\frac{2}{\alpha}\right)^l \frac{\delta_{l+1}}{1 - \alpha/2}\right].$$

(G-4) The function $f_{s,\delta}$ restricted to the interval I_s is of the form

$$f_{s,\delta}(x) = b_{|s|,\delta} - \left(\frac{\alpha}{8}\right)^{|s|} + \left(\frac{\alpha}{2}\right)^{|s|} |x - I_s(0)| \qquad x \in I_s,$$

where $b_{|s|,\delta} = f_{s,\delta}(I_s(-1)) = f_{s,\delta}(I_s(+1))$ is the function value on the endpoints of I_s (defined inductively on $|s|$ and δ_i, $i \le |s|$).

(G-5) For $t \sqsubseteq s$, we have $f_{s,\delta}(x) < b_{|t|,\delta}$ if and only if $x \in \operatorname{int}(I_t)$.

We construct our instance inductively, the case $|s| = 0$ being trivial ($f_\perp(x) = |x|$; note this function does not depend on the perturbations δ). Moreover, let $b_{0,\delta} = 1$, and inductively $b_{l+1,\delta} := b_{l,\delta} - \frac{\alpha}{2}\left(\frac{\alpha}{8}\right)^l - \delta_{l+1}$. Suppose now $|s| = l$ and $\delta \in [0, \bar{\delta}]^M$, and for simplicity let $s_{l+1} = 0$ (the case

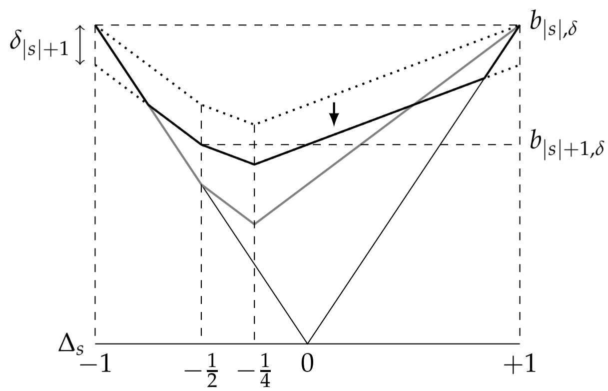

Figure 5: Comparison between instance from Section 4.1 (grey line) and perturbed one (thick line).

$s_{l+1} = 1$ is analogous). By inductive hypothesis $f_{s,\delta}(I_s(-1)) = f_{s,\delta}(I_s(+1)) = b_{|s|,\delta}$. We consider the perturbed extension given by

$$g_{s0,\delta}(x) := \begin{cases} b_{l+1,\delta} - \left(\frac{\alpha}{8}\right)^{l+1} + \left(\frac{\alpha}{2}\right)^{l+1} \left| x - I_s\left(-\frac{1}{4}\right)\right|, & \text{if } x \in I_s\left[-\frac{1}{2}, 1\right] \\ b_{l,\delta} + \alpha\left[f_{s,\delta}(x) - b_{l,\delta}\right] - \delta_{l+1}, & \text{otherwise.} \end{cases}$$

We define the new perturbed instance as follows

$$f_{s0,\delta}(x) = \max\{g_{s0,\delta}(x), f_{s,\delta}(x)\} \qquad x \in [-1,1].$$

Note for example that at $x = I_s(-1/2)$ the function $g_{s0,\delta}$ is continuous, and moreover $g_{s0,\delta}(x) = b_{l+1,\delta} > b_{l,\delta} - \frac{1}{2}\left(\frac{\alpha}{8}\right)^l = f_{s,\delta}(x)$, where the strict inequality holds by definition of $\bar{\delta}$; similarly, for $x = I_s(0)$, $g_{s0,\delta}(x) = b_{l+1,\delta} > f_{s,\delta}(x)$. This way, we guarantee that at the interval I_{s0} the maximum defining $f_{s0,\delta}$ is only achieved by $g_{s0,\delta}$.

The key property of the perturbed instances is the following: Since $\delta_{l+1} > 0$ then $f_{s0,\delta}$ is smooth at $I_s(-1)$ and $I_s(+1)$, and its local behavior does not depend on $\delta_{l+1}, \ldots, \delta_M$. Furthermore, for all $x \in [-1,1] \setminus \operatorname{int}\left(I_s^\delta\right)$, we have $f_{s0,\delta}(x) = f_{s,\delta}(x)$, from which is easy to prove (G-3).

Finally, observe that properties (F-1), (F-2), (G-4) and (G-5) are straightforward to verify. This proves the existence of our family. Moreover, by construction, the function defined above is convex, continuous, and has Lipschitz constant bounded by 1.

To finish our discussion, let us explain the role of these perturbations, and the choice of parameters. First observe that the definition of $g_{s,\delta}$ is obtained by applying two operations to the extension used in Section 4: first we reduce the slope of the extension by a factor α, and then we 'push-down' the function values by an additive perturbation $\delta_{|s|+1}$ (see Figure 5). The motivation for the perturbed family is to provide instances with similar structure than in Section 4; in particular, we preserve the nesting property of level sets. The main difference with the perturbed instance is the smoothness at $I_s(-1)$, $I_s(+1)$: by doing this we hide the behavior (in particular the perturbations) of deeper level sets from its behavior outside the interior of this level set, for any local oracle. In the multidimensional construction the perturbations will have a similar role than in the large-scale case, making the maximizer term unique w.p.1. for any oracle query, as perturbations in different coordinates will be conditionally independent. This process will continue throughout iterations, and the independence of perturbations for deeper level sets is crucial for this to happen.

Multidimensional construction of the family

As in the unperturbed case, the obvious multidimensional extension is to consider the maximum among all coordinates of the one dimensional instance, namely, for a concatenation of (nM)-dimensional strings $\{s_i : i \in [n]\}$, s, and concatenation of (nM)-dimensional vectors $\{\delta_i : i \in [n]\}$, δ, let

$$f_{s,\delta}(x) := \max_{i\in[n]} f_{s_i,\delta_i}(x). \tag{6}$$

Lower bound for fixed perturbation under oracle $\tilde{\mathcal{O}}$ Note that from (F-1) and (G-5) the packing property is satisfied when $M = \lfloor \frac{1}{3-\ln\alpha}\ln(1/\varepsilon)\rfloor$. Next, emulation by the String Guessing Problem comes from analogous results to Lemmas 4.4 and 4.7, considering the obvious modifications due to the perturbations, and whose proofs are thus omitted. This establishes the lower bound $\Omega(n\log(1/\varepsilon))$.

Lower Bounds for $\mathcal{F}$ under oracle $\mathcal{O}$ Similarly as in the large-scale case, the fundamental task is to prove that w.p. 1 at every iteration the information provided by $\mathcal{O}$ can be emulated by the single-coordinate oracle $\tilde{\mathcal{O}}$ studied earlier.

For this, we will analyze the oracle answer, showing that for any nontrivial query the maximizer in (6) is unique w.p. 1. The role of perturbations is crucial for this analysis. With this in hand, the lower bound comes from an averaging argument analogous the large-scale case.

Lemma 6.5 (On unpredictability, box case)**.** *For an $\mathcal{O}$-based algorithm solving family $\mathcal{F}$ with queries $X^1,\dots,X^T$ let the set of maximizer coordinates be*

$$J^t := \{(i,l):\ f_{S,\Delta}(X^t) = f_{S_i,\Delta_i}(X^t),\ b_{l+1,\delta} < f_{S,\Delta}(X^t) \le b_{l,\delta}\}$$

for $t \le T$, and $J^t = \emptyset$ otherwise. For a query $t \le T$ let the i-th depth l_i be such that (i,l_i) is $\prec$-maximal among elements of J^{t-1} with first coordinate i. Finally, let $J_c^t := \{(i,l) : (i,l) \succ (i,l_i)\}$.

Then the distribution of $(\Delta_{i,h})_{J_c^t}$ conditionally on $(\Pi_{<t}, Q_t)$ is absolutely continuous. Moreover, after the oracle answer A_t, w.p. 1 either we only obtain (inexact) lower bounds on some of the $\Delta_{i,h}$, or J^t is a singleton.

Proof. For $t < T$, let the active set be defined as

$$\mathcal{I}^t := \operatorname{int}\left(\prod_{i=1}^{n} I^{\Delta_i}_{s_i|_{l_i+1}}\right).$$

We prove the Lemma by induction on t. The case $t = 1$ clearly satisfies that $(I_{i,l})_{(i,l)\in[n]\times[M]}$ is absolutely continuous. Next, after the first oracle call, there are two cases: first, if the query lies outside the active set $\mathcal{I}^1$, then after the oracle answer all what is learnt are lower bounds on the perturbations (this since the instance behaves as an absolute value function of the maximizer coordinates); by absolute continuity these lower bounds are inexact w.p. 1. If the query lies in $\mathcal{I}^1$ then since the perturbations are absolutely continuous, and since (for all possible realizations of S) the set where the maximizer is not unique is a smaller dimensional manifold, the maximizer in $f_{S,\Delta}$ is unique w.p. 1. In this case all bits preceding this maximizer in the $\prec$-order are learnt, and potentially some perturbations for these bits as well.

Next, let $t \ge 1$, and suppose the Lemma holds up to query t. Then we know that $(\Delta_{i,h})_{J_c^t}$ is absolutely continuous, conditionally on $(\Pi_{<t}, Q_t)$, and that the oracle answer A_t is such that w.p. 1 either we only obtain (inexact) lower bounds on some $\Delta_{i,h}$, or J^t is a singleton. In the first case, $(\Delta_{i,h})_{J_c^t}$ remains absolutely continuous (since lower bounds are inexact), so clearly the statement holds true for $t+1$. In the case J^t is a singleton, note that $(\Delta_{i,h})_{(i,h)\in J_c^{t+1}}$ remains independent and uniform by construction of the function family. This way, by performing the same analysis as in the base case over the set $\prod_{i=1}^n I_{s_i|_{l_i+1}}$ we conclude that the Lemma holds for $t+1$. ☐

Let us define the set

$$E := \bigcap_{t \le T} \{(\Delta_{i,h})_{J_c^t} \text{ is absolutely continuous } \vee |J^t| \le 1\}.$$

By the previous Lemma, $\mathbb{P}[E] = 1$. It is clear that on event E, oracle $\mathcal{O}$ can be emulated by $\tilde{\mathcal{O}}$ by following an analogous approach as in Section 6.1. It is left as exercise to derive from this the lower complexity bound $\Omega(n \log(1/\varepsilon))$, and its variants for expectation, high probability, and bounded error algorithms.

6.2.1 The low-scale case: reduction to the box when $1 \le p < 2$

Finally, as a consequence of our strong lower bounds for arbitrary oracles on the box we derive optimal lower complexity bounds for low-scale optimization over L^p balls for $1 \le p < 2$

Proposition 6.6. *Let $1 \le p < 2$, and $\varepsilon \le n^{-\frac{1}{2}-\delta}$ with $\delta > 0$. There exists a family $\mathcal{F}$ of convex Lipschitz continuous functions in the L^p norm with Lipschitz constant 1 on the n-dimensional unit Euclidean ball $B_p(0,1)$ such that for any local oracle for family $\mathcal{F}$, both the distributional and high-probability oracle complexity of finding an ε-minimum under the uniform distribution is $\Omega\left(n \log \frac{1}{\varepsilon}\right)$.*

For algorithms with error probability at most P_e, the distributional complexity is $\Omega\left((1-P_e) n \log \frac{1}{\varepsilon}\right)$ and the high probability complexity is $\Omega\left(n \log \frac{1}{\varepsilon}\right)$.

Proof. This proof is based on convex geometry and it is inspired by Guzmán and Nemirovski [2013].

Let $\varepsilon \le 1/n^{1/2+\delta}$ and $X := B_p(0,1)$. By Dvoretzky's Theorem on the L^p-ball [Pisier, 1989, Theorem 4.15], there exists a universal constant $\alpha \in (0,1)$ (i.e., independent of p and n), such that for $k = \lfloor \alpha n \rfloor$ there exists a subspace $L \subseteq \mathbb{R}^n$ of dimension k, and a centered ellipsoid $E \subseteq L$ such that

$$\frac{1}{2} E \subseteq X \cap L \subseteq E.$$

Let $\{\gamma_i(\cdot) : i = 1, \ldots, k\}$ be linear forms on L such that $E = \{y \in L : \sum_{i=1}^k \gamma_i^2(y) \le 1\}$. By the second inclusion above, for every $i \in [k]$ the maximum of γ_i over $X \cap L$ does not exceed 1, whence, by the Hahn-Banach Theorem, the linear form $\gamma_i(\cdot)$ can be extended from L to $\mathbb{R}^n$ with its maximum over X not exceeding 1. In other words, there exist k vectors $g_i \in \mathbb{R}^n$, $1 \le i \le k$, such that $\gamma_i(y) = \langle g_i, y \rangle$ for every $y \in L$ and $\|g_i\|_{\frac{p}{p-1}} \le 1$, for all $1 \le i \le k$. Now consider the linear mapping

$$x \mapsto Gx := (\langle g_1, x \rangle, \ldots, \langle g_k, x \rangle) \colon \mathbb{R}^n \to \mathbb{R}^k.$$

The operator norm of this mapping induced by the norms $\|\cdot\|_p$ on the domain and $\|\cdot\|_\infty$ on the codomain does not exceed 1. Therefore, for any Lipschitz continuous function $f \colon \mathbb{R}^k \to \mathbb{R}$ with Lipschitz constant 1 in the $\|\cdot\|_\infty$ norm, the function $\tilde{f} \colon \mathbb{R}^n \to \mathbb{R}$ defined by $\tilde{f}(x) = f(Gx)$ is Lipschitz continuous with constant 1 in the L^p norm. We claim (postponing its proof) that the complexity of Lipschitz continuous functions in the L^p norm on $X \subseteq \mathbb{R}^n$ is lower bounded by the complexity of Lipschitz continuous functions in the L^∞ norm on $B_\infty\left(0, \frac{1}{2\sqrt{k}}\right) \subseteq \mathbb{R}^k$ (as $G\left(B_\infty\left(0, \frac{1}{2\sqrt{k}}\right)\right) \subseteq \frac{1}{2}E \subseteq X$). We conclude that the distributional and high probability oracle complexity of the former family is lower bounded by

$$\Omega\left(k \log \frac{1}{2\sqrt{k}\varepsilon}\right) = \Omega\left(n \log \frac{1}{\varepsilon\sqrt{n}}\right) = \Omega\left(n \log \frac{1}{\varepsilon}\right),$$

for large n, since for $\varepsilon \le n^{-1/2-\delta}$ with $\delta > 0$ we have $\varepsilon\sqrt{n} \le \varepsilon^{\frac{\delta}{1/2+\delta}}$.

We finish the proof by proving the claim: let $\mathcal{G}$ be the subfamily of Lipschitz continuous functions with constant 1 for the L^∞ norm given by (6), defined on the box $B_\infty(0, 1/(2\sqrt{k}))$ of $\mathbb{R}^k$, and let $\mathcal{F}$ be the respective family of 'lifted' instances $\tilde{f} : \mathbb{R}^n \to \mathbb{R}$, which are Lipschitz continuous functions with constant 1 for the L^p norm, defined on the unit ball $B_p(0,1)$ of $\mathbb{R}^n$.

Observe that the maximal oracle $\mathcal{O}$ on $\mathcal{G}$ induces the maximal oracle for family $\mathcal{F}$. Namely, if we let $\tilde{\mathcal{O}}$ be the oracle for family $\mathcal{F}$ defined by $\tilde{\mathcal{O}}_{\tilde{f}}(x) = \tilde{\mathcal{O}}_{\tilde{g}}(x)$ if and only if $\mathcal{O}_f(Gx) = \mathcal{O}_g(Gx)$, then it is easy to see that $\tilde{\mathcal{O}}$ is the maximal oracle for $\mathcal{F}$. This way, any oracle for $\mathcal{F}$ can be emulated by an oracle for $\mathcal{G}$, and thus by Lemma 3.5 lower bounds for $\mathcal{G}$ also hold for $\mathcal{F}$.

□

Acknowledgements

The authors would like to thank Arkadi Nemirovski and François Glineur for the valuable discussions. Research reported in this paper was partially supported by NSF grants CMMI-1232623 and CMMI-1300144.

References

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.

S. Arora and B. Barak. *Computational complexity*. Cambridge University Press, Cambridge, 2009.

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009a.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, Mar. 2009b.

A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM J. Optim*, 12:2001, 2001.

G. Braun and S. Pokutta. Common information and unique disjointness. *Proceedings of FOCS*, 2013.

G. Braun, S. Fiorini, and S. Pokutta. Average case polyhedral complexity of the maximum stable set problem. *submitted*, 2013.

M. Braverman and A. Moitra. An information complexity approach to extended formulations. *Electronic Colloquium on Computational Complexity (ECCC)*, 19(131), 2012.

M. Braverman and A. Rao. Information equals amortized communication. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 748–757. IEEE, 2011.

M. Braverman, A. Garg, D. Pankratov, and O. Weinstein. From information to exact communication. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 19, page 171, 2012.

A. Chakrabarti, G. Cormode, R. Kondapally, and A. McGregor. Information cost tradeoffs for augmented index and streaming language recognition. *SIAM Journal on Computing*, 42(1):61–83, 2013.

F. A. Chudak and K. Nagano. Efficient solutions to relaxations of combinatorial problems with submodular penalties via the lovász extension and non-smooth convex optimization. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 79–88. Society for Industrial and Applied Mathematics, 2007.

T. Cover and J. Thomas. *Elements of information theory*. Wiley-interscience, 2006.

A. Dasgupta, R. Kumar, and D. Sivakumar. Sparse and lopsided set disjointness via information theory. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 517–528. Springer, 2012.

D. Garber and E. Hazan. Playing Non-linear Games with Linear Oracles. *Proceedings of FOCS*, 2013.

G. Goel, C. Karande, P. Tripathi, and L. Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 755–764. IEEE, 2009.

C. Guzmán and A. Nemirovski. On Lower Complexity Bounds for Large-Scale Smooth Convex Optimization. arXiv:1307.5001v2, 2013.

S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 671–680. IEEE, 2009.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 53–61, 2013.

G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. *development*, 2:2, 2013.

A. Nemirovski. Efficient methods in convex programming, 1994. URL `http://www2.isye.gatech.edu/~nemirovs/Lect_EMCO.pdf`.

A. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley - Interscience, 1 edition, 1983. ISBN 0 471 10345 4.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Springer Netherlands, 1 edition, 2004. ISBN 1402075537.

Y. Nesterov and A. Nemirovski. On first-order algorithms for ℓ_1/nuclear norm minimization. *Acta Numerica*, 22:509–575, 4 2013.

G. Pisier. *The Volume of Convex Bodies and Banach Space Geometry*. Cambridge University Press, 1 edition, 1989. ISBN 0-521-36465-5; 0-521-66635-X.

M. Pătraşcu. Unifying the landscape of cell-probe lower bounds. *SIAM Journal on Computing*, 40 (3):827–847, 2011.

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

N. Srebro and K. Sridharan. On convex optimization, fat shattering and learning. Unpublished, 2012. URL `http://ttic.uchicago.edu/~karthik/optfat.pdf`.

Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

M. Zhu, S. J. Wright, and F. T. Chan. Duality-based algorithms for total-variation-regularized image restoration. *Comput. Optim. Appl.*, 47(3):377–400, Nov. 2010.