Lower Bounds for Higher-Order Convex Optimization

Naman Agarwal NAMANA@CS.PRINCETON.EDU
Department of Computer Science, Princeton University, Princeton, NJ
Elad Hazan EHAZAN@CS.PRINCETON.EDU
Department of Computer Science, Princeton University, Princeton, NJ
Google Brain

Editors: Sebastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

State-of-the-art methods in mathematical optimization employ higher-order derivative information. We explore the limitations of higher-order optimization and prove that even for convex optimization, a polynomial dependence on the approximation guarantee and higher-order smoothness parameters is necessary. This refutes the hope that higher-order smoothness and higher-order derivatives can lead to dimension free polynomial time algorithms for convex optimization. As a special case, we show Nesterov's accelerated cubic regularization method and higher-order methods to be nearly tight.

Keywords: Convex Optimization, Second-Order Optimization, Higher-Order Optimization, Newton's method, Lower Bounds.

1. Introduction

Linearly-converging[1] convex optimization algorithms fall into two categories. The first are methods whose iteration complexity scales with the dimension. This includes the ellipsoid (Khachiyan, 1980; Grötschel et al., 2012), cutting plane (Vaidya, 1989; Lee et al., 2015) and random-walk (Bertsimas and Vempala, 2004; Kalai and Vempala, 2006; Lovász and Vempala, 2006) based methods. They solve convex optimization in the general membership oracle model.

The other category of linearly-converging algorithms is iterative derivative-based methods. These achieve fast dimension-free iteration rates for certain types of convex functions, namely those that are strongly convex and smooth. Indeed gradient descent and further extensions have therefore been extremely successful for optimization especially in machine learning.

More recently state-of-the-art optimization for machine learning has shifted from gradient based methods, namely stochastic gradient descent and its derivatives (Duchi et al., 2011; Johnson and Zhang, 2013), to methods that are based on higher moments. Notably, the fastest theoretical running times for both convex (Agarwal et al., 2016; Xu et al., 2016; Bollapragada et al., 2016) and non-convex (Agarwal et al., 2017; Carmon et al., 2016) optimization are attained by algorithms that either explicitly or implicitly exploit second-order information and third-order smoothness.

Of particular interest is Newton's method, due to recent efficient implementations that run in near-linear time in the input representation. The hope was that Newton's method or higher-order methods can achieve logarithmic in error iteration complexity which is independent of the dimen-

1. methods for which the dependence on the additive error ε is logarithmic

sionality, which is extremely high for many large-scale applications, and without requiring strong convexity.

Method	Dim. Free	Order	Assumptions	Upper Bound
Ellipsoid (Grötschel et al., 2012) (Vaidya, 1989) Random Walks (Kalai and Vempala, 2006) Interior Point (Nesterov and Nemirovskii, 1994) (Abernethy and Hazan, 2015)	No	NA	None	$poly\left(d, \log(1/\varepsilon)\right)$
Gradient Descent (Nesterov, 2004)	Yes	k=1	Bounded L_2	$O\left(\left(\frac{L_2}{\varepsilon}\right)^{1/2}\right)$
			$\lambda_{\min} > 0$	$O\left(\left(\frac{L_2}{\lambda_{\min}}\right)^{1/2} \log(\frac{1}{\varepsilon})\right)$
Newton's method (Monteiro and Svaiter, 2013)	Yes	k=2	Bounded L_3	$O\left(\left(\frac{L_3}{\varepsilon}\right)^{2/7}\right)$
			$\lambda_{\min} > 0$	$\tilde{O}\left(\left(\frac{L_3}{\lambda_{\min}}\right)^{\frac{2}{7}} + \log\log(\frac{1}{\varepsilon})\right)$
Higher-Order (Baes, 2009)	Yes	$k > 2$	Bounded L_k	$O\left(\left(\frac{L_{k+1}}{\varepsilon}\right)^{1/(k+1)}\right)$

Table 1: Table summarizing known results for convex optimization.

Table 1 surveys the known algorithms for convex optimization and their iteration complexity. Polynomial time algorithms admit *linear convergence*, i.e. $O(\log \frac{1}{\varepsilon})$, and invariably depend on the dimension. Dimension free polynomial-time algorithms require both an upper bound on the smoothness and a lower bound on the strong convexity.

The main question we set to answer is **whether there exists linearly-converging iterative algorithms for (high-order) smooth functions that are not strongly convex?**

In this paper we show that unfortunately, these hopes cannot be attained without stronger assumptions on the underlying optimization problem. In particular Theorem 1 shows that even if the functions are k^{th}-order smooth, for arbitrarily large k, and the iterative algorithm uses k^{th}-order derivative information, the answer is negative. To the best of our knowledge, our results are the first lower bound for k^{th}-order optimization for $k \geq 2$ that include higher-order smoothness.[2]

1.1. Statement of Result

We consider the problem of k^{th}-order optimization. We model a k^{th}-order algorithm as follows. Given a k-times differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, at every iteration i, the algorithms outputs a point x_i and receives as input the tuple $[f(x_i), \nabla f(x_i), \nabla^2 f(x_i) \ldots \nabla^k f(x_i)]$, i.e. the value of the

2. After the writing of the first manuscript we were made aware of the work by Arjevani et al. (2017) which provides lower bounds for these settings as well.

function and its k derivatives at x_i.[3] The goal of the algorithm is to output a point x_T such that

$$f(x_T) - \min_{x \in \mathbb{R}^d} f(x) \leq \varepsilon.$$

For the k^{th}-order derivatives to be informative, one needs to bound their rate of change, or equivalently the Lipschitz constant of its derivative. This is called k^{th}-order smoothness, and we denote it by L_{k+1}. In particular we assume that

$$\|\nabla^k f(x) - \nabla^k f(y)\| \leq L_{k+1}\|x - y\|,$$

where $\|\cdot\|$ is defined as the induced operator norm with respect to the Euclidean norm. Our main theorem shows the limitation of k^{th}-order iterative optimization algorithms:

Theorem 1 *For every number L_{k+1} and k^{th}-order algorithm* ALG *(deterministic or randomized), there exists a number $\varepsilon_0(L_{k+1})$ such that for all $\varepsilon \leq \varepsilon_0(L_{k+1})$, there exists a k-differentiable convex function $f : B_d \to \mathbb{R}$ with k^{th}-order smoothness coefficient L_{k+1} such that* ALG *cannot output a point x_T such that*

$$f(x_T) \leq \min_{x \in B_d} f(x) + \varepsilon,$$

in number of iterations T fewer than

$$c_k \left(\frac{L_{k+1}}{\varepsilon} \right)^{\Omega(1/k)},$$

where c_k is a constant depending on k and B_d is defined to be the unit ball in d dimensions.

The above lower bound is known to be tight up to constants in the exponent: for $k > 2$, Baes (2009) proves an upper bound of $O\left(\left(\frac{L_{k+1}}{\varepsilon}\right)^{\frac{1}{k+1}}\right)$.

Although the bound is stated for constrained optimization over the unit ball, it can be extended to an unconstrained setting via the addition of an appropriate scaled multiple of $\|x\|^2$. We leave this adaptation for a full version of this paper. Further as is common with lower bounds the underlying dimension d is assumed to be large enough and differs for the deterministic vs randomized version. Theorems 10 and 13 make the dependence precise.

Table 2 surveys the known lower bounds for derivative based optimization. For the case of $k = 2$, the most efficient methods known are the cubic regularization technique proposed by Nesterov (2008) and an accelerated hybrid proximal extra-gradient method proposed by Monteiro and Svaiter (2013). The best known upper bound in this setting is $O\left(\frac{L_3}{\varepsilon}\right)^{2/7}$ (Monteiro and Svaiter, 2013). We show a lower bound of $\Omega\left(\left(\frac{L_3}{\varepsilon}\right)^{2/11}\right)$ (c.f. Theorem 6) demonstrating that the upper bound is nearly tight.

3. An iteration is equivalent to an oracle call to k^{th}-order derivatives in this model.

Oracle	Lower bound
First-Order Oracle	$\Omega\left(\left(\frac{L_2}{\varepsilon}\right)^{1/2}\right)$ (Nemirovsky and Yudin, 1978)
Second-Order Oracle	$\Omega\left(\left(\frac{L_3}{\varepsilon}\right)^{2/11}\right)$ (This paper - c.f. Theorem 6)
k^{th}-Order Oracle	$\Omega\left(\left(\frac{L_{k+1}}{\varepsilon}\right)^{2/(5k+1)}\right)$ (This paper - c.f. Theorem 6)

Table 2: Lower bounds for higher order oracles

1.2. Related work.

The literature on convex optimization is too vast to survey; the reader is referred to Boyd and Vandenberghe (2004); Nesterov (2004).

Lower bounds for convex optimization were studied extensively in the seminal work of Nemirovsky and Yudin (1978). In particular, tight first-order optimization lower bounds were established assuming first-order smoothness.(Also see Nesterov (2004) for a concise presentation of the lower bound). In a recent work, Arjevani and Shamir (2016) presented a lower bound when given access to second-order derivatives. However a key component (as remarked by the authors themselves) missing from the bound established by Arjevani and Shamir (2016) was that the constructed function was not third-order smooth. Indeed the lower bound established by Arjevani and Shamir (2016) can be overcome when the function is third-order smooth (ref. Nesterov (2008)). The upper bounds for higher-order oracles (assuming appropriate smoothness) was established by Baes (2009).

Higher order smoothness has been leveraged recently in the context of non-convex optimization (Agarwal et al., 2017; Carmon et al., 2016; Allen-Zhu, 2017). In a surprising new discovery, Carmon et al. (2017) show that assuming higher-order smoothness the bounds for first-order optimization can be improved without having explicit access to higher-order oracles. This is a property observed in our lower bound too. Indeed as shown in the proof the higher order derivatives at the points queried by the algorithm are always 0. For further details regarding first-order lower bounds for various different settings we refer the reader to Agarwal et al. (2009); Woodworth and Srebro (2016b); Arjevani and Shamir (2016); Arjevani et al. (2015) and the references therein. The work of Guzmán and Nemirovski (2015) studies a different kind of smoothing namely inf-convolution to obtain first-order smoothness in arbitrary norms, however it does not provide guarantees with higher-order smoothness.

In parallel and independently, Arjevani et al. Arjevani et al. (2017) also obtain lower bounds for deterministic higher-order optimization. In comparison, their lower bound is stronger in terms of the exponent than the ones proved in this paper, and matches the upper bound for $k = 2$. However, our construction and proof are simple (based on the well known technique of ball smoothing) and our bounds hold for randomized algorithms as well, as opposed to their deterministic lower bounds.

1.3. Overview of Techniques

Our lower bound is inspired by the lower bound presented in Clarkson et al. (2012). In particular we construct the function as a piece-wise linear convex function defined by $f(x) = \max_i\{a_i^T x\}$ with carefully constructed vectors a_i and restricting the domain to be the unit ball. The key idea here is

that querying a point reveals information about at most one hyperplane. The optimal point however can be shown to require information about all the hyperplanes.

Unfortunately the above function is not differentiable. We now smooth the function by the ball smoothing operator (defined in Definition 3) which averages the function in a small Euclidean ball around a point. We show (c.f. Corollary 5) that iterative application of the smoothing operator ensures k-differentiability as well as boundedness of the k^{th}-order derivative.

Two key issues arise due to smoothing described above. Firstly although the smoothing operator leaves the function unchanged around regions far away from the intersection of the hyperplanes, it is not the case for points lying near the intersection. Indeed querying a point near the intersection of the hyperplanes can potentially lead to leak of information about multiple hyperplanes at once. To avoid this, we carefully shift the linear hyperplanes making them affine and then arguing that this shifting indeed forces sufficient gap between the points queried by the algorithm and the intersections leaving sufficient room for smoothing.

Secondly such a smoothing is well known to introduce a dependence on the dimension d in the smoothness coefficients. Our key insight here is that for the class of functions being considered for the lower bound (c.f. Definition 2) smoothing can be achieved without a dependence on the dimension(c.f. Theorem 4). This is essential to achieving dimension free lower bounds and we believe this characterization can be of intrinsic interest.

1.4. Organization of the paper

We begin by providing requisite notation and definitions for the smoothing operator and proving the relevant lemmas regarding smoothing in Section 2. Section 3 provides a quantitative statement of our main theorem. In Section 4 we provide the construction of our hard function. In Section 5 we state and prove our main theorem (Theorem 10) showing the lower bound against deterministic algorithms. We also prove Theorem 1 based on Theorem 10 in this Section. In Section 6 we state and prove Theorem 13, showing the lower bound against randomized algorithms.

2. Preliminaries

2.1. Notation

We use B_d to refer to the d-dimensional ℓ_2 unit ball. We suppress the d from the notation when it is clear from the context. Let Γ be an r-dimensional linear subspace of $\mathbb{R}^d$. We denote by M_Γ, an $r \times d$ matrix which contains an orthonormal basis of Γ as rows. Let $\Gamma^\perp$ denote the orthogonal complement of Γ. Given a vector v and a subspace Γ, let $v \perp \Gamma$ denote the perpendicular component of v w.r.t Γ. We now define the notion of a Γ-invariant function.

Definition 2 (Γ-invariance) *Let Γ be an r dimensional linear subspace of $\mathbb{R}^d$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be Γ-invariant if for all $x \in \mathbb{R}^d$ and y belonging to the subspace $\Gamma^\perp$, i.e. $M_\Gamma y = 0$, we have that*

$$f(x) = f(x + y)$$

Equivalently there exists a function $g : \mathbb{R}^r \to \mathbb{R}$ such that for all x, $f(x) = g(M_\Gamma x)$.

A function $f : \mathbb{R}^d \to \mathbb{R}$ is defined to be c-Lipschitz with respect to a norm $\|\cdot\|$ if it satisfies

$$f(x) - f(y) \leq c\|x - y\|$$

Lipschitzness for the rest of the paper will be measured in the ℓ_2 norm.

2.2. Smoothing

In this section we define the smoothing operator and derive the requisite properties.

Definition 3 (Smoothing operator) *Given an r-dimensional subspace $\Gamma \in \mathbb{R}^d$ and a parameter $\delta > 0$, define the operator $S_{\delta,\Gamma} : (\mathbb{R}^d \to \mathbb{R}) \to (\mathbb{R}^d \to \mathbb{R})$ (referred henceforth as the smoothing operator) as*

$$S_{\delta,\Gamma} f(x) \triangleq \mathbb{E}_{v \in \Gamma, \|v\| \leq 1}[f(x + \delta v)],$$

where the expectation is over sampling a unit vector from the subspace Γ uniformly randomly.

As a shorthand we define $f_{\delta,\Gamma} \triangleq S_{\delta,\Gamma} f$. Further for any $t \in \mathbb{N}$ define $S^t_{\delta,\Gamma} f \triangleq S_{\delta,\Gamma}(...S_{\delta,\Gamma}(f))$ i.e. the smoothing operator applied on f iteratively t times.

When $\Gamma = \mathbb{R}^d$ we suppress the notation from $f_{\delta,\Gamma}$ to f_δ. Following is the main lemma we prove regarding the smoothing operator.

Lemma 4 *Let Γ be an r-dimensional linear subspace of $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ be Γ-invariant and G-Lipschitz. Let $f_{\delta,\Gamma} \triangleq S_{\delta,\Gamma} f$ be the smoothing of f. Then we have the following properties.*

1. *$f_{\delta,\Gamma}$ is differentiable and also G-Lipschitz and Γ-invariant.*
2. *$\nabla f_{\delta,\Gamma}$ is $\frac{rG}{\delta}$-Lipschitz.*
3. *$\forall\ x : |f_{\delta,\Gamma}(x) - f(x)| \leq \delta G$.*

Following is a corollary of the above lemma.

Corollary 5 *Given a G-Lipschitz continuous function f and an r-dimensional subspace Γ such that f is Γ-invariant, we have that the function $S^k_{\delta,\Gamma} f$ is k-times differentiable $\forall\ k$. Moreover we have that for any x, y*

$$\forall i \in [k]\ \ \|\nabla^i S^k_{\delta,\Gamma} f(x) - \nabla^i S^k_{\delta,\Gamma} f(y)\| \leq \left(\frac{r}{\delta}\right)^i G\|x - y\|,$$

$$|S^k_{\delta,\Gamma} f(x) - f(x)| \leq G\delta k.$$

The proofs of Lemma 4 and Corollary 5 are included in the appendix.

3. Main Theorem Statement

The main result we prove in the paper is given by the following theorem. The theorem is a restatement of Theorem 1.

Theorem 6 *For every number L_{k+1} and k^{th}-order algorithm ALG (deterministic or randomized), there exists an $\varepsilon_0(L_{k+1})$ such that for all $\varepsilon \leq \varepsilon_0(L_{k+1})$, there exists a k-differentiable convex function $f \in B_d \to \mathbb{R}$ with k^{th}-order smoothness coefficient L_{k+1} such that ALG cannot output a point x_T such that*

$$f(x_T) \leq \min_{x \in B_d} f(x) + \varepsilon,$$

in number of iterations T fewer than

$$c_k \left(\frac{L_{k+1}}{\varepsilon} \right)^{\frac{2}{5k+1}}.$$

where c_k is a constant depending on k.

The rest of the paper is dedicated to the proof of the above theorem.

4. Construction of the hard function

In this section we describe the construction of our hard function $f^\dagger$. Our construction is inspired by the information-theoretic hard instance based on zero-sum games proposed by Clarkson et al. (2012). The construction of the function will be characterized by a sequence of vectors $X^{1\to r} = \{x_1 \dots x_r\}$, $x_i \in B_d$ and parameters k, γ, δ, m. We assume $d > m \geq r$. To make the dependence explicit we denote the hard function as

$$f^\dagger(X^{1\to r}, \gamma, k, \delta, m) : B_d \to \mathbb{R}.$$

For brevity in the rest of the section we suppress $X^{1\to r}, \gamma, k, \delta, m$ from the notation, however all the quantities defined in the section depend on them. To define $f^\dagger$ we will define auxiliary vectors $\{a_1 \dots a_r\}$ and auxiliary functions $f, \tilde{f}$.

Given a sequence of vectors $\{x_1, x_2, \dots x_r\}, x_i \in B_d$, let X_i for $i \leq r$ be defined as the subspace spanned by the vectors $\{x_1 \dots x_i\}$. Further inductively define vectors $\{a_1 \dots a_r\}$ as follows. If $x_i \notin X_{i-1}$, define

$$a_i \triangleq \frac{\hat{a}_i}{\|\hat{a}_i\|} \text{ where } \hat{a}_i \triangleq x_i \perp X_{i-1}.$$

If indeed $x_i \in X_i$, then a_i is defined to be an arbitrary unit vector in the orthogonal component $X_{i-1}^\perp$. Further define an auxiliary function

$$f(x) \triangleq \max_{i \in [r]} f_i(x) \text{ where } f_i(x) \triangleq a_i^T x.$$

Given the parameter γ, now define the following functions

$$\tilde{f}(x) \triangleq \max_{i \in [r]} \tilde{f}_i(x) \text{ where } \tilde{f}_i(x) \triangleq f_i(x) + \left(1 - \frac{i}{m}\right)\gamma \triangleq a_i^T x + \left(1 - \frac{i}{m}\right)\gamma.$$

With these definitions in place we can now define the hard function parametrized by k, δ. Let A_r be the subspace spanned by $\{a_1 \dots a_r\}$

$$f^\dagger(X^{1\to r}, k, \gamma, \delta, m) \triangleq S^k_{\delta, A_r}\ \tilde{f}(X^{1\to r}, \gamma, m), \tag{4.1}$$

i.e. $f^\dagger$ is constructed by smoothing $\tilde{f}$ k-times with respect to the parameters δ, A_r. We now collect some important observations regarding the function $f^\dagger$.

Observation 7 *$f^\dagger$ is convex and continuous. Moreover it is 1-Lipschitz and is invariant with the respect to the r-dimensional subspace A_r.*

Note that $\tilde{f}$ is a $\max$ of linear functions and hence convex. Since smoothing preserves convexity we have that $f^\dagger$ is convex. 1-Lipschitzness follows by noting that by definition $\|a_i\| = 1$ and it can be seen that $\tilde{f}$ is A_r-invariant and therefore by Theorem 4 we get that $f^\dagger$ is A_r-invariant.

Observation 8 *$f^\dagger$ is k-differentiable with the Lipschitz constants $L_{i+1} \leq \left(\frac{r}{\delta}\right)^i$ for all $i \leq k$.*

Above is a direct consequence of Corollary 5 and the fact that $\tilde{f}$ is 1-Lipschitz and invariant with respect to the r-dimensional subspace A_r. Corollary 5 also implies that

$$\forall x \;\; |f^\dagger(x) - \tilde{f}(x)| \leq k\delta. \tag{4.2}$$

Setting $\hat{x} \triangleq -\sum_{i=1}^{r} \frac{a_i}{\sqrt{r}}$, we get that $f(\hat{x}) = \frac{-1}{\sqrt{r}}$. Therefore the following inequality follows from Equation (4.2) and by noting that $\|f(x) - \tilde{f}(x)\|_\infty \leq \gamma$

$$\min_{x \in B_d} f^\dagger(x) \leq f^\dagger(\hat{x}) \leq \frac{-1}{\sqrt{r}} + \gamma + k\delta \tag{4.3}$$

The following lemma provides a characterization of the derivatives of $f^\dagger$ at the points x_i.

Lemma 9 *Given a sequence of vectors $\{x_1 \dots x_r\}$ and parameters δ, γ, r, m, let $\{g_1 \dots g_r\}$ be a sequence of functions defined as*

$$\forall\, i \;\; g_i \triangleq f^\dagger(X^{1\to i}, k, \gamma, \delta, m).$$

If the parameters are such that $2k\delta \leq \frac{\gamma}{m}$ then we have that

$$\forall\, i \in [r]\; \forall j \in [k] \;\; g_i(x_i) = g_r(x_i),\;\; \nabla^j g_i(x_i) = \nabla^j g_r(x_i).$$

Proof

We will first note the following about the smoothing operator S_δ^k. At any point x, all the k derivatives and the function value of $S_\delta^k f$ for any function f depend only on the value of the function f in a ball of radius at most $k\delta$ around the point x. Consider the function g_r and g_i for any $i \in [r]$. Note that by definition of the functions g_i, for any x such that

$$\operatorname*{argmax}_{j \in [r]} \;\; a_j^T x + \left(1 - \frac{j}{m}\right)\gamma \leq i$$

we have that $g_i(x) = g_r(x)$. Therefore to prove the lemma it is sufficient to show that

$$\forall\, i, x \in \|x - x_i\| \leq k\delta \quad \operatorname*{argmax}_{j \in [r]} \;\; a_j^T x + \left(1 - \frac{j}{m}\right)\gamma \leq i.$$

Let us first note the following facts. By construction we have that $\forall j > i, a_j^T x_i = 0$. This immediately implies that

$$\max_{j > i} \;\; a_j^T x_i + \left(1 - \frac{j}{m}\right)\gamma = \left(1 - \frac{i+1}{m}\right)\gamma. \tag{4.4}$$

Further using the fact that $\|a_j\| \leq 1, \forall j \in [r]$ we have that

$$\forall x \;\text{ s.t. }\; \|x - x_i\| \leq k\delta \;\text{ we have }\; \max_{j > i} \;\; a_j^T x + \left(1 - \frac{j}{m}\right)\gamma \leq \left(1 - \frac{i+1}{m}\right)\gamma + k\delta. \tag{4.5}$$

Further note that by construction $a_i^T x_i \geq 0$ which implies $a_i^T x + \left(1 - \frac{i}{m}\right)\gamma \geq \left(1 - \frac{i}{m}\right)\gamma$. Again using the fact that $\|a_j\| \leq 1, \forall j \in [r]$ we have that

$$\forall x \text{ s.t. } \|x - x_i\| \leq k\delta \text{ we have } \max_{j \leq i} a_j^T x + \left(1 - \frac{j}{m}\right) \geq \left(1 - \frac{i}{m}\right)\gamma - k\delta. \tag{4.6}$$

The above equations in particular imply that as long as $2k\delta < \frac{\gamma}{m}$, we have that

$$\forall x \text{ s.t. } \|x - x_i\| \leq k\delta \quad \operatorname*{argmax}_{j \in [r]} a_j^T x + \left(1 - \frac{j}{m}\right) \leq i \tag{4.7}$$

which as we argued before is sufficient to prove the lemma. ■

5. Main Theorem and Proof

The following theorem (Theorem 10) proves the existence of the required hard function. Theorem 6 for the deterministic case is a simple derivation which we provide after the theorem statement.

Theorem 10 *For any integer k, any $T > 5k$, and $d > T$ and any k^{th}-order deterministic algorithm, there exists a convex function $f^\dagger : B_d \to \mathbb{R}$ for every $d > T$, such that for T steps of the algorithm every point $y \in B_d$ queried by the algorithm is such that*

$$f^\dagger(y) \geq \min_{x \in B_d} f^\dagger(x) + \frac{1}{2\sqrt{T}}.$$

Moreover the function is guaranteed to be k-differentiable with Lipschitz constants L_{i+1} bounded as

$$\forall\, i \leq k \;\; L_{i+1} \leq (10k)^i T^{2.5i}. \tag{5.1}$$

We first prove Theorem 6 in the deterministic case using Theorem 10.

Proof [Proof of Theorem 6 Deterministic case] Given an algorithm ALG and numbers L_{k+1}, k define $\varepsilon_0(L_{k+1}, k) \triangleq L_{k+1}/(10k)^k$. For any $\varepsilon \leq \varepsilon_0$ pick a number $\mathcal{T}$ such that

$$\varepsilon = \frac{L_{k+1}}{2(10k)^k \mathcal{T}^{(2.5k+0.5)}}.$$

Let $f^\dagger$ be the function constructed in Theorem 10 for parameters $k, \mathcal{T}$, ALG and define the hard function $h : B_d \to \mathbb{R}$

$$h(x) \triangleq \frac{L_{k+1}}{(10k)^k \mathcal{T}^{2.5k}} f^\dagger(x).$$

Note that by the guarantee in Equation (5.1) we get that $h(x)$ is k^{th}-order smooth with coefficient at most L_{k+1}. Note that since this is a scaling of the original hard function $f^\dagger$ the lower bound applies directly and therefore ALG cannot achieve accuracy

$$\frac{L_{k+1}}{2(10k)^k \mathcal{T}^{2.5k}\sqrt{\mathcal{T}}} \triangleq \varepsilon,$$

in less than $\mathcal{T} = c_k \left(\frac{L_{k+1}}{\varepsilon}\right)^{\frac{2}{5k+1}}$ iterations where c_k is a constant only depending on k. This finishes the proof of the theorem. ■

We now provide the proof of Theorem 10.

Proof [Proof of Theorem 10]

Define the following parameters $\gamma \triangleq \frac{1}{3\sqrt{T}}$ and $\delta_T \triangleq \frac{\gamma}{3kT}$.

Consider a deterministic algorithm ALG. Since ALG is deterministic let the first point played by the algorithm be fixed to be x_1. We now define a series of functions $f_i^\dagger$ inductively for all $i = \{1, \dots T\}$ as follows

$$X^{1\to i} \triangleq \{x_1 \dots x_i\} \qquad f_i^\dagger \triangleq f^\dagger(X^{1\to i}, \gamma, k, \delta_T, T) \tag{5.2}$$

$$Inp_i^x \triangleq \{f_i^\dagger(x_i), \nabla f_i^\dagger(x_i) \dots \nabla^k f_i^\dagger(x_i)\} \qquad x_{i+1} \triangleq \text{ALG}(Inp_1^x, \dots Inp_i^x) \tag{5.3}$$

The above definitions *simulate* the deterministic algorithm ALG with respect to changing functions $f_i^\dagger$. Inp_i^x is the input the algorithm will receive if it queried point x_i and the function was $f_i^\dagger$. x_{i+1} is the next point the algorithm ALG will query on round $i+1$ given the inputs $\{Inp_1^x \dots Inp_i^x\}$ over the previous rounds. Note that thus far these quantities are tools defined for analysis. Since ALG is deterministic these quantities are all deterministic and well defined. We will now prove that the function $f_T^\dagger$ defined in the series above satisfies the properties required by the Theorem 10.

Bounded Lipschitz Constants Using Corollary 5, the fact that $f^\dagger$ has Lipschitz constant bounded by 1 and that $f_T^\dagger$ is invariant with respect to a T dimensional subspace, we get that the function $f_T^\dagger$ has higher order Lipschitz constants bounded above as

$$\forall i \le k \quad L_{i+1} \le \left(\frac{T}{\delta_T}\right)^i \le \left(10kT^{2.5}\right)^i.$$

Suboptimality

Let $\{y_0 \dots y_T\}$ be the points queried by the algorithm ALG when executed on $f_T^\dagger$. We need to show that

$$\forall i \in [1 \dots T] \qquad f_T^\dagger(y_i) \ge \min_{x \in B_d} f_T^\dagger(x) + \frac{1}{2\sqrt{T}}. \tag{5.4}$$

Equation 5.4 follows as a direct consequence of the following two claims.

Claim 11 *We have that for all $i \in [1, T]$, $y_i = x_i$ where x_i is defined by Equation (5.3).*

Claim 12 *We have that*

$$\forall i \in [1 \dots T] \qquad f_T^\dagger(x_i) \ge \min_{x \in B_d} f_T^\dagger(x) + \frac{1}{2\sqrt{T}}.$$

To remind the reader, x_i (Equation (5.3)) are variables which were defined by simulating the algorithm on a changing function where as y_i are the points played by the algorithm ALG when run on $f_T^\dagger$. Claim 11 shows that even though $f_T^\dagger$ was constructed using x_i the outputs produced by the algorithm does not change.

Claim 11 and Claim 12 derive Equation 5.4 in a straightforward manner thus finishing the proof of Theorem 10.

■

We now provide the proofs of Claim 11 and Claim 12.

Proof [Proof of Claim 11] Note that since the algorithm is deterministic y_1 is fixed and y_i for $i \geq 2$ is defined inductively as follows.

$$Inp_i^y \triangleq \{f_T^\dagger(y_i), \nabla f_T^\dagger(y_i), \dots \nabla^k f_T^\dagger(y_i)\} \qquad y_{i+1} = \text{ALG}(Inp_1^y, \dots Inp_T^y) \tag{5.5}$$

We will prove the claim via strong induction. The base case $x_1 = y_1$ is immediate because ALG is deterministic and therefore the first point queried by it is always the same.

Assume now that the claim holds for all $j \leq i$. Since by definition $2k\delta_T \leq \gamma/T$, we can see as a direct consequence of Lemma 9, that

$$\{\forall j \leq i \;\; x_j = y_j\} \Rightarrow \{\forall j \leq i \;\; Inp_i^{y_i} = Inp_i^{x_i}\} \tag{5.6}$$

where Inp_i^x is as defined in Equation (5.3). Note that $Inp_i^{x_i}$ is the set of derivatives of $f_i^\dagger$ at x_i and $Inp_i^{y_i}$ is the set of derivatives of $f_T^\dagger$ at y_i. Also by definition we have that

$$\{\forall j \leq i \;\; Inp_i^{y_i} = Inp_i^{x_i}\} \Rightarrow \{x_{i+1} = y_{i+1}\}.$$

Putting the above two together we have that $\{\forall j \leq i \;\; x_j = y_j\} \Rightarrow \{x_{i+1} = y_{i+1}\}$ which finishes the induction. ■

Proof [Proof of Claim 12] Using Lemma 9 we have that $f_i^\dagger(x_i) = f_T^\dagger(x_i)$. Further Equation (4.6) implies that

$$f_i^\dagger(x_i) \geq \left(1 - \frac{i}{T}\right)\gamma - k\delta_T.$$

Now using (4.3) using we get that every point in $\{x_1 \dots x_T\}$ is such that

$$f_T^\dagger(x_i) - \min_{x \in B} f_T^\dagger(x) \geq \left(\frac{1}{\sqrt{T}} - \frac{i\gamma}{T} - 2k\delta_T\right) \geq \frac{1}{2\sqrt{T}}.$$

The above follows by the choice of parameters and T being large enough. This finishes the proof of Claim 12. ■

6. Lower Bounds against Randomized Algorithms

In this section we prove the version of Theorem 10 for randomized algorithms. The key idea underlying the proof remains the same. However since we cannot *simulate* the algorithm anymore we choose the vectors $\{a_i\}$ forming the subspace randomly from $\mathbb{R}^d$ for a large enough d. This ensures that no algorithm with few queries can discover the subspace in which the function is non-invariant with reasonable probability. Naturally the dimension required for Theorem 10 now is larger than the tight $d > T$ we achieved as in the case of deterministic algorithms.

The proof of Theorem 6 for randomized algorithms follows in exactly the same way as the proof for the deterministic case using Theorem 10.

Theorem 13 *For any integer k, any $T > 5k$, $\delta \in [0,1]$, and any k-order (potentially randomized algorithm), there exists a k-differentiable convex function $f^\dagger : B_d \to \mathbb{R}$ for $d = \Omega(T^3 \log(T^2/\delta))$, such that with probability at least $1-\delta$ (over the randomness of the algorithm) for T steps of the algorithm every point y queried by the algorithm is such that*

$$f^\dagger(y) \geq \min_{x \in B_d} f^\dagger(x) + \frac{1}{2\sqrt{T}}.$$

Moreover the function $f^\dagger$ is guaranteed to be k-differentiable with Lipschitz constants L_i bounded as

$$\forall\, i \leq k \;\; L_{i+1} \leq (20kT^{2.5})^i.$$

Due to space constraints the proof of Theorem 13 is included in the appendix.

7. Conclusion

We have considered the problem of achieving dimension free polynomial time algorithms for minimizing convex functions where the function is guaranteed to be k-differentiable and k^{th}-order smooth and the algorithm is allowed to have access to k derivatives at every iteration. We showed an oracle complexity lower bound for convex optimization under these conditions demonstrating that the number of points queried by any deterministic/randomized algorithm should have at least an inverse polynomial dependence on the desired error. This rules out linearly-converging derivative-based algorithms even under these assumptions.

While we provide precise guarantees for the dependence on the exponent, our bounds are weaker than those proved independently and concurrently by Arjevani et al. (2017) (which only applies to deterministic algorithms). We believe that our construction (or potentially a similar one) might be able to achieve the improved bounds and leave this direction as immediate future work. Furthermore we remark that while the known upper and lower bounds are tight for first and second order optimization, they are not tight for $k > 2$ and this is an intriguing open question.

Acknowledgments

The authors would like to acknowledge and thank Ohad Shamir for providing insightful comments on the first draft of this manuscript and Brian Bullins and Gopi Sivakanth for helpful suggestions. The authors are supported by NSF grant 1523815.

References

J Abernethy and E Hazan. Faster convex optimization: Simulated annealing with an efficient universal barrier. arxiv 1507.02528, 2015.

Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization for machine learning in linear time. *arXiv preprint arXiv:1602.03943*, 2016.

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1195–1199, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4528-6. doi: 10.1145/3055399.3055464. URL http://doi.acm.org/10.1145/3055399.3055464.

Zeyuan Allen-Zhu. Natasha: Faster stochastic non-convex optimization via strongly non-convex parameter. *arXiv preprint arXiv:1702.00763*, 2017.

Yossi Arjevani and Ohad Shamir. Oracle complexity of second-order methods for finite-sum problems. *arXiv preprint arXiv:1611.04982*, 2016.

Yossi Arjevani, Shai Shalev-Shwartz, and Ohad Shamir. On lower and upper bounds for smooth and strongly convex optimization problems. *arXiv preprint arXiv:1503.06833*, 2015.

Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *arXiv preprint arXiv:1705.07260v2*, 2017.

Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *arXiv preprint arXiv:1609.08502*, 2016.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *arXiv preprint 1611.00756*, 2016.

Yair Carmon, Oliver Hinder, John C Duchi, and Aaron Sidford. " convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. *arXiv preprint arXiv:1705.02766*, 2017.

Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *Journal of the ACM*, 59(5):23:1–23:49, October 2012. ISSN 00045411. doi: 10.1145/2371656.2371658.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

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

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

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

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Adam Tauman Kalai and Santosh Vempala. Simulated annealing for convex optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

Leonid G Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1049–1065. IEEE, 2015.

László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 57–68. IEEE, 2006.

Renato DC Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

Arkadi Nemirovsky and David Yudin. *Problem complexity and method efficiency in optimization.* Nauka Publishers, Moscow (in Russian), 1978. John Wiley, New York (in English) 1983.

Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Pravin M Vaidya. A new algorithm for minimizing convex functions over convex sets. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 338–343. IEEE, 1989.

Blake Woodworth and Nati Srebro. Tight Complexity Bounds for Optimizing Composite Objectives. In *NIPS*, 2016a.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems*, pages 3639–3647, 2016b.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

Appendix A. Proofs regarding Smoothing

A.1. Proof of Lemma 4

Proof As stated before f being Γ-invariant implies that there exists a function g such that $f(x) = g(\Gamma x)$. Therefore we have that

$$f_{\delta,\Gamma}(x) = \mathbb{E}_{v\in\Gamma,\|v\|\leq 1}[f(x+\delta v)] = \mathbb{E}_{v\in\Gamma,\|v\|\leq 1}[g(M_\Gamma x + \delta M_\Gamma v)] = g_\delta(M_\Gamma x)$$

where $g_\delta(x) \triangleq S_{\delta,\mathbb{R}^r} g(x)$. The representation of $f_{\delta,\Gamma}$ as $g_\delta(M_\Gamma x)$ implies that $f_{\delta,\Gamma}$ is Γ-invariant. Further the above equality implies that $\nabla f_{\delta,\Gamma}(x) = M_\Gamma^T \nabla g_\delta(M_\Gamma x)$. A standard argument using Stokes' theorem shows that g_δ is differentiable even when g is not [4] and that $\nabla g_\delta(y) = \frac{r}{\delta}\mathbb{E}_{v\sim S_r}[g(y+\delta v)v]$(Lemma 1 Flaxman et al. (2005)), where S_r is the r-dimensional sphere, i.e. $S_r = \{x \in \mathbb{R}^r | \|x\| = 1\}$

$$\begin{aligned}\|\nabla g_\delta(x) - \nabla g_\delta(y)\| &= \frac{r}{\delta}\|\mathbb{E}_{v\sim S_r}[g(x+\delta v)v] - \mathbb{E}_{v\sim S_r}[g(y+\delta v)v]\| \\ &\leq \frac{r}{\delta}\mathbb{E}_{v\sim S_r}[|g(x+\delta v) - g(y+\delta v)|\|v\|] \leq \frac{rG}{\delta}\|x-y\|.\end{aligned}$$

The first inequality follows from Jensen's inequality and the second inequality follows from noticing that f being G-Lipschitz implies that g is G-Lipschitz. We now have that

$$\|\nabla f_{\delta,\Gamma}(x) - \nabla f_{\delta,\Gamma}(y)\| \leq \|M_\Gamma\left(\nabla g_\delta(M_\Gamma x) - \nabla g_\delta(M_\Gamma y)\right)\| \leq \frac{rG}{\delta}\|M_\Gamma\|\|(x-y)\| \leq \frac{rG}{\delta}\|(x-y)\|.$$

f being G-Lipschitz immediately gives us $\forall\ x : |f_{\delta,\Gamma}(x) - f(x)| \leq \delta G$. ■

A.2. Proof of Corollary 5

Proof We will argue inductively. The base case ($k = 0$) is a direct consequence of the function f being G-Lipschitz. Suppose the theorem holds for $k-1$. To argue about $\|\nabla^i S^k_{\delta,\Gamma} f(x) - \nabla^i S^k_{\delta,\Gamma} f(y)\|$ we will consider the function $q_{i,v}(x) = \nabla^i S^{k-1}_{\delta,\Gamma} f(x)[v^{\otimes i}]$ for $i \in [k]$ and for a unit vector v. We will first consider the case $i < k$. Using the inductive hypothesis and the fact that smoothing and derivative commute for differentiable functions we have that

$$S_{\delta,\Gamma}\ q_{i,v}(x) = \nabla^i S^k_{\delta,\Gamma} f(x)[v^{\otimes i}]\ .$$

Note that the inductive hypothesis implies that $q_{i,v}(x)$ is $\left(\frac{r}{\delta}\right)^i G$-Lipschitz and so is $S_{\delta,\Gamma}\ q_{i,v}(x)$ via Lemma 4. Therefore we have that

$$\forall i \in [k-1]\ \ \|\nabla^i S^k_{\delta,\Gamma} f(x) - \nabla^i S^k_{\delta,\Gamma} f(y)\| \leq \left(\frac{r}{\delta}\right)^i G\|x-y\|.$$

We now consider the case when $i = k$. By Lemma 4 we know that $S_{\delta,\Gamma}\ q_{k-1,v}(x) = \nabla^{k-1} S^k_{\delta,\Gamma} f(x)[v^{\otimes i}]$ is differentiable and therefore we have that $S^k_{\delta,\Gamma} f(x)$ is k times differentiable. Further we have that

$$\nabla S_{\delta,\Gamma}\ q_{k-1,v}(x) = \nabla^k S^k_{\delta,\Gamma} f(x)[v^{\otimes k-1}].$$

4. We need g to be not differentiable in a measure 0 set which is always the case with our constructions

A direct application of Lemma 4 and the inductive hypothesis which implies that $q_{k-1,v}$ is $\left(\frac{r}{\delta}\right)^{k-1} G$-Lipschitz gives that

$$\|\nabla^k S^k_{\delta,\Gamma} f(x) - \nabla^k S^k_{\delta,\Gamma} f(y)\| \leq \left(\frac{r}{\delta}\right)^k G\|x-y\|.$$

Further it is immediate to see that

$$\inf_{y:\|y-x\|\leq k\delta} f(y) \leq S^k_{\delta,\Gamma} f(x) \leq \sup_{y:\|y-x\|\leq k\delta} f(y)$$

which implies using the fact that f is G Lipschitz that

$$|S^k_{\delta,\Gamma} f(x) - f(x)| \leq G\delta k.$$

■

Appendix B. Lower bound against Randomized Algorithms

B.1. Proof of Theorem 13

Proof We provide a randomized construction for the function $f^\dagger$. The construction is the same as in Section 4 but we repeat it here for clarity. We sample a random T dimensional orthonormal basis $\{a_1 \dots a_T\}$ from the uniform distribution on the orthonormal group $O(T)$. Let A_i be the subspace spanned by $\{a_1 \dots a_i\}$ and $A_i^\perp$ be the orthogonal complement of A_i. Further define an auxiliary function

$$f(x) \triangleq \max_i f_i(x) \text{ where } f_i(x) \triangleq a_i^T x \ .$$

Given a parameter γ, now define the following functions

$$\tilde{f}(x) \triangleq \max_i \tilde{f}_i(x) \text{ where } \tilde{f}_i(x) \triangleq f_i(x) + \left(1 - \frac{i}{T}\right)\gamma \triangleq a_i^T x + \left(1 - \frac{i}{T}\right)\gamma,$$

$$f^\dagger(k,\gamma,\delta_T) \triangleq S^k_{\delta_T, A_T}\, \tilde{f}, \tag{B.1}$$

i.e. smoothing $\tilde{f}$ with respect to δ_T, A_T. The hard function we propose is the random function $f^\dagger$ with parameters set as $\gamma = \frac{1}{3\sqrt{T}}$ and $\delta_T = \frac{1}{20kT^{1.5}}$. We restate facts which can be derived analogously to those derived in Section 4 (c.f. Equations (4.2),(4.3)).

$$\forall x \ \ |f^\dagger(x) - \tilde{f}(x)| \leq k\delta \quad \text{and} \quad \min_{x\in B_d} f^\dagger(x) \leq \frac{-1}{\sqrt{T}} + \gamma + k\delta. \tag{B.2}$$

The following key lemma will be the main component of the proof.

Lemma 14 *Let $\{x_1 \dots x_T\}$ be the points queried by a randomized algorithm throughout its execution on the function $f^\dagger$. With probability at least $1-\delta$ (over the randomness of the algorithm and the selection of $f^\dagger$) the following event $\mathcal{E}$ happens*

$$\mathcal{E} = \left\{\forall i \in [T] \ \ \forall j \geq i \ \ |a_j^T x_i| \leq \frac{1}{20T^{1.5}}\right\}.$$

Using the above lemma we first demonstrate the proof of Theorem 13. We will assume the event $\mathcal{E}$ in Lemma 14 happens.

Bounded Lipschitz Constants Using Corollary 5, the fact that $f^\dagger$ has Lipschitz constant bounded by 1 and that $\tilde{f}$ is invariant with respect to the T dimensional subspace A_T, we get that the function $f_T^\dagger$ has higher order Lipschitz constants bounded above as

$$\forall i \leq k \quad L_{i+1} \leq \left(\frac{T}{\delta_T}\right)^i \leq \left(20kT^{2.5}\right)^i.$$

Sub-optimality : The event $\mathcal{E}$ in the lemma implies that $\tilde{f}_i(x_i) \geq -\frac{1}{20T^{1.5}} + (1-\frac{i}{T})\gamma$ which implies that $\tilde{f}(x_i) \geq -\frac{1}{20T^{1.5}} + (1-\frac{i}{T})\gamma$ and from Equation (B.2) we get that

$$\forall i \in [T] \;\; f^\dagger(x_i) \geq -\frac{1}{20T^{1.5}} + \left(1 - \frac{i}{T}\right)\gamma - k\delta_T.$$

Now using Equation (B.2) we get that every x_i is such that

$$\forall i \in [T] \quad f^\dagger(x_i) - \min_{x \in B} f^\dagger(x) \geq \frac{1}{\sqrt{T}} - \frac{i}{T}\gamma - 2k\delta_T - \frac{1}{20T^{1.5}} \geq \frac{1}{2\sqrt{T}}.$$

The last inequality follows by the choice of parameters. This finishes the proof of Theorem 13. ■

Proof [Proof of Lemma 14] We will use the following claims to prove the lemma. For any vector x, define the event $\mathcal{E}_i(x) = \left\{\forall j \geq i \;\; |a_j^T x| \leq \frac{1}{20T^{1.5}}\right\}$. The event we care about then is

$$\mathcal{E} \triangleq \bigcap_{i=1 \to T} \mathcal{E}_i(x_i).$$

Claim 15 *Let $x \in \mathbb{R}^d$. If $\mathcal{E}_i(x)$ holds, then $[f^\dagger(x), \nabla f^\dagger(x) \ldots \nabla^k f^\dagger(x)]$ all depend only on $\{a_1 \ldots a_i\}$.*

Claim 16 *Let $i \in [T]$. Assume that $\forall j < i$, $\mathcal{E}_j(x_j)$ holds. Then $\mathcal{E}_i(x_i)$ holds with probability at least $1 - \frac{\delta}{T}$(over the choice of a_i and the randomness of the algorithm).*

Claim 15 is a robust version of the argument presented in the proof of Theorem 10. Claim 16 is a byproduct of the fact that in high dimensions the correlation between a fixed vector and a random small basis is small. Claim 15 is used to prove the Claim 16.

Lemma 14 now follows via a simple inductive argument using Claim 16 which is as follows

$$\Pr(\mathcal{E}) = \Pr\left(\bigcap_{i=1 \to T} \mathcal{E}_i(x_i)\right) = \prod_{i=1 \to T} \Pr\left(\mathcal{E}_i(x_i) \middle| \bigcap_{j<i} \mathcal{E}_j(x_j)\right) \geq \left(1 - \frac{\delta}{T}\right)^T \geq 1 - \delta.$$

■

Proof [Proof of Claim 15] As noted before the smoothing operator S_δ^k is such that at any point x all the k derivatives of $S_\delta^k f$ depend only on the value of f in a ball of radius $k\delta$ around the point x.

Therefore it is sufficient to show that for the function $\tilde{f}$, for every y such that $\|y - x\| \leq k\delta_T$ we have that $\tilde{f}(y)$ depends only on $\{a_1 \dots a_i\}$. To ensure this, it is enough to ensure that for every such y we have that $\operatorname{argmin}_{j\in[T]} \tilde{f}_j(y) \leq i$ which is what we prove next.

Let us first note the following facts. By the definition of $\mathcal{E}_i(x)$ we have that $\forall j > i, f_j(x) \leq \frac{1}{20T^{1.5}}$. This immediately implies that

$$\max_{j>i} \; \tilde{f}_j(x_i) \leq \frac{1}{20T^{1.5}} + \left(1 - \frac{i+1}{T}\right)\gamma. \tag{B.3}$$

Now since we know each $\tilde{f}_i$ is 1-Lipschitz [5], this also gives us

$$\forall y \;\text{ s.t. }\; \|y - x\| \leq k\delta_T \;\text{ we have }\; \max_{j>i} \; \tilde{f}_j(y) \leq \frac{1}{20T^{1.5}} + \left(1 - \frac{i+1}{T}\right)\gamma + k\delta_T. \tag{B.4}$$

By the event $\mathcal{E}_i(x)$ we also know that $\tilde{f}_i(x) \geq -\frac{1}{20T^{1.5}} + (1 - \frac{i}{T})\gamma$. This implies as above

$$\forall y \;\text{ s.t. }\; \|y - x\| \leq k\delta_T \;\text{ we have }\; \max_{j\leq i} \; \tilde{f}_j(y) \geq -\frac{1}{20T^{1.5}} + \left(1 - \frac{i}{T}\right)\gamma - k\delta_T. \tag{B.5}$$

The above equations imply that as long as $2k\delta_T + \frac{1}{10T^{1.5}} < \frac{\gamma}{T}$ (which is true by the choice of parameters), we have that

$$\forall y \;\text{ s.t. }\; \|y - x\| \leq k\delta_T \;\; \operatorname*{argmin}_{j\in[t]} \tilde{f}_j(y) \leq i. \tag{B.6}$$

which is sufficient to prove Claim 15. ■

Proof [Proof of Claim 16] Consider any $i \in [T]$. Given $\mathcal{E}_j(x_j)$ is true for all $j < i$, applying Claim 15 for all $j < i$, implies that all the information that the algorithm possesses is only a function of $\{a_1 \dots a_{i-1}\}$ and the internal randomness of the algorithm. Since $\{a_1 \dots a_T\}$ was sampled from the uniform distribution on the orthonormal group, we can assume that it was sampled by the inductive process which picks a_i as a uniformly random unit vector from the subspace $A_{i-1}^{\perp}$, which is defined to be the orthogonal component of A_{i-1}, the subspace spanned by $\{a_1, \dots a_{i-1}\}$.

The above inductive procedure for sampling implies that conditioned on $\{a_1 \dots a_{i-1}\}$, the distribution of the remaining vectors $\{a_i \dots a_T\}$ is uniform on the orthogonal group $O(T - i + 1)$, lying in the $d - i + 1$-dimensional subspace $A_{i-1}^{\perp}$. The above implies that the distribution over $\{a_i \dots a_T\}$ is conditionally independent of any x_i the algorithm might play. Since we wish to bound the absolute value of the inner product we can assume $\|x_i\| = 1$ [6].

Following the above arguments, proving the lemma now reduces to the following. Consider a fixed unit vector y in a $\mathbb{R}^{d-i+1}$. The vector y corresponds to the vector x_i played by the algorithm. Further, consider picking a uniformly random $T-i+1$ dimensional subspace of $\mathbb{R}^{d-i+1}$ represented by the basis $\{y_1 \dots y_{T-i+1}\}$. $\{y_1 \dots y_{T-i+1}\}$ represent the vectors $\{a_i \dots a_T\}$. The lemma now reduces to bounding $\forall\, j$, the probability

$$\Pr\left(|\langle y, y_j\rangle| > \frac{1}{20T^{1.5}}\right).$$

5. $\|a_i\| = 1$
6. Otherwise the absolute value of the inner product is only lower

The rest of the argument follows the argument by Woodworth and Srebro (2016a)(Proof of Lemma 7). Note that for y_1 this probability amounts to the ratio between the surface area of a sphere above the caps of radius $\sqrt{1-(\frac{1}{20T^{1.5}})^2}$ and the surface area of the unit sphere. This surface area is smaller than the ratio between the surface area of a sphere of radius $\sqrt{1-(\frac{1}{20T^{1.5}})^2}$ and the surface area of a unit sphere. Formally this gives us

$$\Pr\left(|y_1^T y| \geq \frac{1}{20T^{1.5}}\right) \leq \sqrt{\left(1-\left(\frac{1}{20T^{1.5}}\right)^2\right)^{d-i+1}}.$$

Applying the argument inductively we get that

$$\forall j \in [1, T-i+1] \quad \Pr\left(|y_j^T y| \geq \frac{1}{20T^{1.5}}\right) \leq \sqrt{\left(1-\left(\frac{1}{20T^{1.5}}\right)^2\right)^{d-i-j+2}}.$$

Using the union bound we have that

$$\begin{aligned}\Pr(\mathcal{E}_i(x_i)) \geq 1-\left(\Pr\left(\bigcup_{j=1\to T-i+1}\left(|y_j^T y| \geq \frac{1}{20T^{1.5}}\right)\right)\right) &\geq 1-(T-i)\left(1-\left(\frac{1}{20T^{1.5}}\right)^2\right)^{\frac{d-T+1}{2}} \\ &\geq 1-(T-i)e^{-(\frac{1}{20T^{1.5}})^2\frac{d-T}{2}} \geq 1-\frac{\delta}{T}. \quad \text{(B.7)}\end{aligned}$$

The last line follows from the choice of $d = \Omega\left(T^3\log(T^2/\delta)\right)$. ■