ON THE LOWER BOUND OF MINIMIZING POLYAK-ŁOJASIEWICZ FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Polyak-Łojasiewicz (PL) (Polyak, 1963) condition is a weaker condition than the strong convexity but suffices to ensure a global convergence for the Gradient Descent algorithm. In this paper, we study the lower bound of algorithms using first-order oracles to find an approximate optimal solution. We show that any first-order algorithm requires at least $\Omega\left(\left(\frac{L}{\mu}\right)^{1-\alpha}\right)$ gradient costs to find an ε-approximate optimal solution for a general L-smooth function that has an μ-PL constant for any $\alpha > 0$. This result demonstrates the near *optimality* of the Gradient Descent algorithm to minimize smooth PL functions in the sense that there exists a "hard" PL function such that no first-order algorithm can be faster by a *polynomial* order. In contrast, it is well-known that the momentum technique, e.g. (Nesterov, 2003, chap. 2) can provably accelerate Gradient Descent to $O\left(\sqrt{\frac{L}{\hat{\mu}}}\log\frac{1}{\varepsilon}\right)$ gradient costs for functions that are L-smooth and $\hat{\mu}$-strongly convex. Therefore, our result distinguishes the hardness of minimizing a smooth PL function and a smooth strongly convex function as the complexity of the former cannot be improved by any polynomial order in general.

1 INTRODUCTION

We consider the problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{1}$$

where the function f is L-smooth and satisfies the Polyak-Łojasiewicz condition. A function f is said to satisfy the Polyak-Łojasiewicz condition if (2) holds for some $\mu > 0$:

$$\|\nabla f(\mathbf{x})\|^2 \geq 2\mu\left(f(\mathbf{x}) - \inf_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})\right), \quad \forall \mathbf{x}\in\mathbb{R}^d. \tag{2}$$

We refer to (2) as the μ-PL condition and simply denote $\inf_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})$ by f^*. The PL condition may be originally introduced by Polyak dating back to 1963 (Polyak, 1963). It is also a special case of the Łojasiewicz condition (Lojasiewicz, 1963) . The PL condition is strictly weaker than strong convexity as one can show that any $\hat{\mu}$-strongly convex function that admits:

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle\nabla f(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle + \frac{\hat{\mu}}{2}\|\mathbf{x}-\mathbf{y}\|^2$$

is also $\hat{\mu}$-PL by minimizing both sides with respect to $\mathbf{x}$ (Karimi et al., 2016). However, the PL condition does not even imply convexity. From a geometric view, the PL condition suggests that the sum of the squares of the gradient dominates the optimal function value gap, which implies that any local stationary point is a global minimizer. Because it is relatively easy to obtain an approximate local stationary point by first-order algorithms, the PL condition serves as an ideal and weaker alternative to strong convexity.

In machine learning, the PL condition has received wide attention recently. Lots of models are found to satisfy this condition under different regimes. Examples include, but are not limited to, matrix decomposition and linear neural networks under a specific initialization (Hardt & Ma, 2016;

Li et al., 2018), nonlinear neural networks in the so-called neural tangent kernel regime (Liu et al., 2022), reinforcement learning with linear quadratic regulator (Fazel et al., 2018). Compared with strong convexity, the PL condition is much easier to hold since the reference point in the latter only is a minimum point such that $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{y}} f(\mathbf{y})$, instead of any $\mathbf{y}$ in the domain.

Turning to the theoretic side, it is known (Karimi et al., 2016) that the standard Gradient Descent algorithm admits a linear converge to minimize a L-smooth and μ-PL function. To be specific, in order to find an ε-approximate optimal solution $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) - f^* \leq \varepsilon$, Gradient Decent needs $O(\frac{L}{\mu} \log \frac{1}{\varepsilon})$ gradient computations. However, it is still not clear whether there exist algorithms that can achieve a provably faster convergence rate. In the optimization community, it is perhaps well-known that the momentum technique, e.g. Nesterov (2003, chap. 2), can provably accelerate Gradient Descent from $O(\frac{L}{\hat{\mu}} \log \frac{1}{\varepsilon})$ to $O\left(\sqrt{\frac{L}{\hat{\mu}}} \log \frac{1}{\varepsilon}\right)$ for functions that are L-smooth and $\hat{\mu}$-strongly convex. Even though some works (J Reddi et al., 2016; Lei et al., 2017) have considered accelerations under different settings, probably faster convergence of first-order algorithms for PL functions is still not obtained up to now.

In this paper, we study the first-order complexities to minimize a generic smooth PL function and ask the question:

"Is the Gradient Decent algorithm (nearly) optimal or can we design a much faster algorithm?"

We answer the question in the language of min-max lower bound complexity for minimizing the L-smooth and μ-PL function class. We analyze the worst complexity of minimizing any function that belongs to the class using first-order algorithms. Excitingly, we construct a hard instance function showing that any first-order algorithm requires at least $\Omega\left(\left(\frac{L}{\mu}\right)^{1-\alpha}\right)$ gradient costs to find an ε-approximate optimal solution, where α is a positive number. This answers the aforementioned question in an explicit way: due to the arbitrariness of the positive factor α, the Gradient Descent algorithm is already nearly *optimal* in the sense that no first-order algorithm can achieve a provably faster convergence rate by a polynomial order in general. For the first time, we distinguish the hardness of minimizing a PL function and a strongly convex function in terms of first-order complexities, as the momentum technique for smooth and strongly convex functions provably accelerates Gradient Descent by a certain polynomial order.

It is worth mentioning that the optimization problem under our consideration is high-dimensional and the goal is to obtain the complexity bounds that do not have an explicit dependency on the dimension.

Our technique to establish the lower bound follows from the previous lower bounds in convex (Nesterov, 2003) and non-convex optimization (Carmon et al., 2021). The main idea is to construct a so-called "zero-chain" function ensuring that any first-order algorithm per-iteratively can only solve one coordinate of the optimization variable. Then for a "zero-chain" function that has a sufficiently high dimension, some number of entries will never reach their optimal values after the execution of any first-order algorithm in certain iterations. To obtain the desired $\Omega\left(\left(\frac{L}{\mu}\right)^{1-\alpha}\right)$ lower bound, we propose a "zero-chain" function similar to Carmon et al. (2020), which is composed of the worst convex function designed by Nesterov (2003) and a separable function in the form as $\sigma \sum_{i=1}^{T} \upsilon_{T,c}(\mathbf{x}_i)$ to destroy the convexity. Different from their separable function, the one that we introduce has a very large Lipshictz constant. This property helps us to estimate the PL constant in a convenient way. This new idea gives new insights into the constructions and analyses of instance functions, which might be potentially generalized to establish the lower bounds for other non-convex problems.

NOTATION

We use bold letters, such as $\mathbf{x}$, to denote vectors in the Euclidean space $\mathbb{R}^d$, and bold capital letters, such as $\mathbf{A}$, to denote matrices. $\mathbf{I}_d$ denotes the identity matrix of size $d \times d$. We omit the subscript and simply denote $\mathbf{I}$ as the identity matrix when the dimension is clear from context. For $\mathbf{x} \in \mathbb{R}^d$, we use $\mathbf{x}_i$ to denote its ith coordinate. We use $\operatorname{supp}(\mathbf{x})$ to denote the subscripts of non-zero entries of $\mathbf{x}$, i.e. $\operatorname{supp}(\mathbf{x}) = \{i : \mathbf{x}_i \neq 0\}$. We use $\operatorname{span}\left\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}\right\}$ to denote the linear subspace spanned by $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}$, i.e. $\left\{\mathbf{y} : \mathbf{y} = \sum_{i=1}^{n} a_i \mathbf{x}^{(i)}, a_i \in \mathbb{R}\right\}$. We call a function f L-smooth if

∇f is L-Lipschitz continuous, i.e. $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$. We denote $f^* = \inf_{\mathbf{x}} f(\mathbf{x})$. We let $\mathbf{x}^*$ be any minimizer of f, i.e., $\mathbf{x}^* = \mathrm{argmin} f$. We always assume the existence of $\mathbf{x}^*$. We say that $\mathbf{x}$ is an ε-approximate optimal point of f when $f(\mathbf{x}) - f^* \leq \varepsilon$.

2 Related Work

Lower Bounds There has been a line of research concerning the lower bounds of algorithms on certain function classes. To the best of our knowledge, (Nemirovskij & Yudin, 1983) defines the oracle model to measure the complexity of algorithms. For convex functions and first-order oracles, the lower bound is studied in (Nesterov, 2003). (Woodworth & Srebro, 2016) extends the result to a finite-sum of convex functions. When the function is non-convex, it is generally NP-hard to find its global minima, or to test whether a point is a local minimum or a saddle point (Murty & Kabadi, 1985). Instead of finding ε-approximate optimal points, an alternative measure is finding ε-stationary points where $\|\nabla f(\mathbf{x})\| \leq \varepsilon$. Sometimes, additional constraints on the Hessian matrices of second-order stationary points are needed. Results of this kind include Carmon et al. (2020; 2021); Fang et al. (2018); Zhou & Gu (2019); Arjevani et al. (2019; 2020). Though a PL function may be non-convex, it is tractable to find an ε-approximate optimal point, as local minima of a PL function must be global minima. In this paper, we give the lower complexity bound for finding ε-approximate optimal points.

PL Condition The PL condition may be first proposed in (Polyak, 1963), It is also a special case of the Łojasiewicz condition in (Lojasiewicz, 1963). Besides the PL condition, there are other relaxations of the strong convexity, including error bounds (Luo & Tseng, 1993), essential strong convexity (Liu et al., 2014), weak strong convexity (Necoara et al., 2019), restricted secant inequality (Zhang & Yin, 2013), and quadratic growth (Anitescu, 2000). Karimi et al. (2016) discussed the relationships between these conditions. To be specific, all these relaxations implies the PL condition except for the quadratic growth, which implies that the PL condition is quite general. There are many other papers that study designing event algorithms to optimize a PL objective function under different scenarios, for example, Bassily et al. (2018); Nouiehed et al. (2019); Hardt & Ma (2016); Fazel et al. (2018); J Reddi et al. (2016); Lei et al. (2017).

3 Preliminaries

3.1 Upper Bound on PL functions

Although the PL condition is a weaker condition than strong convexity, it guarantees linear convergence for Gradient Descent. The result can be found in Polyak (1963) and Karimi et al. (2016). We present it here for completeness.

Theorem 1. *If f is L-smooth and satisfies μ-PL condition, then the Gradient Descent algorithm with a constant step-size $\frac{1}{L}$:*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{1}{L}\nabla f(\mathbf{x}^{(k)}), \tag{3}$$

has a linear convergence rate. We have:

$$f(\mathbf{x}^{(k)}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(\mathbf{x}^{(0)}) - f^*). \tag{4}$$

Theorem 1 shows that the Gradient Descent algorithm finds the ε-approximate optimal point of f in $O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ gradient computations. This gives an upper complexity bound for first-order algorithms. However, it remains open to us whether there are faster algorithms for smooth PL functions. We will establish a lower complexity bound on first-order algorithms, which nearly matches the upper bound.

3.2 Definitions of Algorithm Classes and Function Classes

An algorithm is a mapping from real-valued functions to sequences. For algorithm A and $f : \mathbb{R}^d \to \mathbb{R}$, we define $\mathsf{A}[f] = \{\mathbf{x}^{(i)}\}_{i\in\mathbb{N}}$ to be the sequence of algorithm A acting on f, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$.

Note here, the algorithm under our consideration works on function defined on any Euclidean space. We call it the dimension-free property of the algorithm.

The definition of algorithms abstracts away from the the optimization process of a function. We consider algorithms which only make use of the first-order information of the iteration sequence. We call them first-order algorithms. If an algorithm is a first-order algorithm, then

$$\mathbf{x}^{(i)} = \mathsf{A}^{(i)}\left(\mathbf{x}^{(0)}, \nabla f(\mathbf{x}^{(0)}), \cdots, \mathbf{x}^{(i-1)}, \nabla f(\mathbf{x}^{(i-1)})\right), \tag{5}$$

where $\mathsf{A}^{(i)}$ is a function depending on A. Perhaps the simplest example of first-order algorithms is Gradient Descent.

We are interested in finding an ε-approximate point of a function f. Given a function $f : \mathbb{R}^d \to \mathbb{R}$ and an algorithm A, the complexity of A on f is the number of queries to the first-order oracle needed to find an ε-approximate point. We denote $T_\varepsilon(\mathsf{A}, f)$ to be the gradient complexity of A on f, then

$$T_\varepsilon(\mathsf{A}, f) = \min_t \left\{ t : f(\mathbf{x}^{(t)}) - f^* \leq \varepsilon \right\}. \tag{6}$$

In practice, we do not have the full information of the function f. We only know that f is in a particular function class $\mathcal{F}$, such as L-smooth functions. Given an algorithm A. We denote $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ to be the complexity of A on $\mathcal{F}$, and define $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ as follows:

$$\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F}) = \sup_{f \in \mathcal{F}} T_\varepsilon(\mathsf{A}, f). \tag{7}$$

Thus, $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ is the worst-case complexity of functions $f \in \mathcal{F}$.

For searching an ε-approximate optimal point of a function in $\mathcal{F}$, we need to find an algorithm which have a low complexity on $\mathcal{F}$. Denote an algorithm class by $\mathcal{A}$. The lower bound of an algorithm class on $\mathcal{F}$ describes the efficiency of algorithm class $\mathcal{A}$ on function class $\mathcal{F}$, which is defined to be

$$\mathcal{T}_\varepsilon(\mathcal{A}, \mathcal{F}) = \inf_{\mathsf{A} \in \mathcal{A}} \mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F}) = \inf_{\mathsf{A} \in \mathcal{A}} \sup_{f \in \mathcal{F}} T_\varepsilon(\mathsf{A}, f). \tag{8}$$

3.3 Linear-span Algorithm

Among all the algorithms, a special algorithm class is called linear-span algorithms. If A is a linear-span algorithm and $\mathsf{A}[f] = \left\{\mathbf{x}^{(t)}\right\}_{t \in \mathbb{N}}$, then the following condition holds for all $f : \mathbb{R}^d \to \mathbb{R}$:

$$\mathbf{x}^{(n)} \in \mathbf{x}^{(0)} + \operatorname{span}\left\{\nabla f(\mathbf{x}^{(0)}), \cdots, \nabla f(\mathbf{x}^{(n-1)})\right\}, \quad \forall n \in \mathbb{N}. \tag{9}$$

That is to say, $\mathbf{x}^{(n)} - \mathbf{x}^{(0)}$ lies in the linear subspace spanned by $\nabla f(\mathbf{x}^{(0)}), \cdots, \nabla f(\mathbf{x}^{(n-1)})$. We denote the collection of first-order linear-span algorithms with $\mathbf{x}^{(0)} = \mathbf{0}$ by $\mathcal{A}_{\text{ls}}$. It is shown by Nemirovskij & Yudin (1983) that a lower complexity bound on first-order linear-span algorithms are also a lower complexity bound on all the first-order algorithm when the function class satisfies the orthogonal invariance property.

3.4 Zero-chain

A zero-chain f is a function that safisfies the following condition:

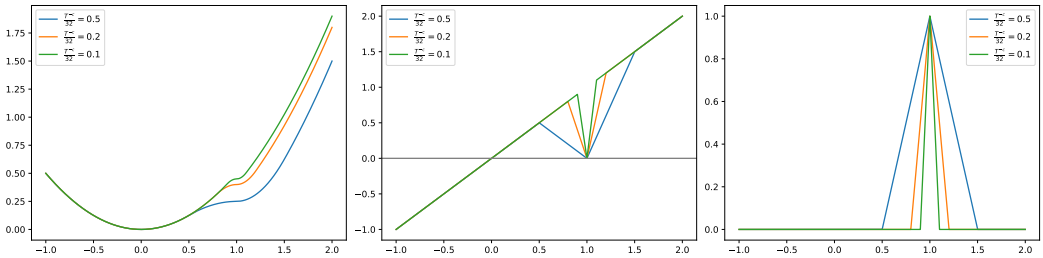
$$\operatorname{supp}(\mathbf{x}) \subseteq \{1, 2, \cdots, k\} \implies \operatorname{supp}(\nabla f(\mathbf{x})) \subseteq \{1, 2, \cdots, k+1\}, \quad \forall \mathbf{x}. \tag{10}$$

In other words, the support of $\nabla f(\mathbf{x})$ lies in a restricted linear subspace depending on the support of $\mathbf{x}$.

The "worst function in the (convex) world" in Nesterov (2003) defined as

$$f_k(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_1 - 1)^2 + \sum_{i=1}^{k-1} (\mathbf{x}_{i+1} - \mathbf{x}_i)^2 \tag{11}$$

is a zero-chain, because if $\mathbf{x}_i = 0$ for $i > n$, then $(\nabla f_k(\mathbf{x}))_{i+1} = 0$ for $i > n$. A zero-chain is difficult to optimize for linear-span algorithms, because linear-span algorithms only discover one coordinate by one gradient computation.



(a) $v_{T,c}$ (b) $v'_{T,c}$ (c) $b_{T,c}$

Figure 1: Illustration of $v_{T,c}$, $v'_{T,c}$ and $b_{T,c}$.

4 Main Results

According to Theorem 1, we already have an upper complexity bound $O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ by applying Gradient Descent to all the PL functions. In this section, we establish the lower complexity bound of first-order algorithms on PL functions. Let $\mathcal{P}(\Delta,\mu,L)$ be the collection of all L-smooth and μ-PL functions f with $f(\mathbf{x}^{(0)}) - f^* \leq \Delta$. We establish a lower bound of $\mathcal{T}_\varepsilon\left(\mathcal{A}_{\text{ls}}, \mathcal{P}(\Delta,\mu,L)\right)$ by constructing a function which is hard to optimize for linear-span algorithms, and extend the result to first-order algorithms.

4.1 Hard Instance

We introduce a hard function $f_{T,c,\sigma} : \mathbb{R}^T \to \mathbb{R}$ for first-order algorithms:

$$f_{T,c,\sigma}(\mathbf{x}) = q(\mathbf{x}) + \sigma \sum_{i=1}^{T} v_{T,c}(\mathbf{x}_i), \tag{12}$$

where

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}_1^2 + \frac{1}{2}\sum_{i=1}^{T-1}\left(\mathbf{x}_{i+1} - \mathbf{x}_i\right)^2 \tag{13}$$

is a quadratic function, and $v_{T,c}$ is defined in (14). The parameter T is a positive integer, and c, σ are positive real numbers.

$$v_{T,c}(x) = \begin{cases} \frac{1}{2}x^2, & x \leq 1 - T^{-c}, \\ \frac{1}{2}x^2 - 16T^c\left(x - 1 + T^{-c}\right)^2, & 1 - \frac{1}{32}T^{-c} < x \leq 1, \\ \frac{1}{2}x^2 - \frac{1}{32}T^{-c} + 16T^c\left(x - 1 - T^{-c}\right)^2, & 1 < x \leq 1 + \frac{1}{32}T^{-c}, \\ \frac{1}{2}x^2 - \frac{1}{32}T^{-c}, & x > 1 + \frac{1}{32}T^{-c}. \end{cases} \tag{14}$$

The function q can be rewritten as

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}, \tag{15}$$

where

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \tag{16}$$

is a positive-definite symmetric matrix.

From the definition of $v_{T,c}$ in (14), we obtain the derivative of $v_{T,c}$, which is:

$$v'_{T,c}(x) = \begin{cases} x, & x \leq 1 - \frac{1}{32}T^{-c}, \\ x - 32T^c\left(x - 1 + \frac{1}{32}T^{-c}\right), & 1 - \frac{1}{32}T^{-c} < x \leq 1, \\ x + 32T^c\left(x - 1 - \frac{1}{32}T^{-c}\right), & 1 < x \leq 1 + \frac{1}{32}T^{-c}, \\ x, & x > 1 + \frac{1}{32}T^{-c}. \end{cases} \tag{17}$$

As we can see in (17), $v'_{T,c}$ is a piecewise linear function. To further simplify notations, we define $b_{T,c}(x)$ in (18):

$$b_{T,c}(x) = \begin{cases} 1 - 32T^c|x-1|, & 1 - \frac{1}{32}T^{-c} \leq x \leq 1 + \frac{1}{32}T^{-c}, \\ 0, & \text{otherwise}. \end{cases} \tag{18}$$

Then $v'_{T,c}(x) = x - b_{T,c}(x)$. Figure 1 provides geometric view of $v_{T,c}$ and $b_{T,c}$.

The quadratic part q is a translation of "the worst function in the (convex) world" in Nesterov (2003), and the definition of $v_{T,c}$ is inspired by the hard instance in Carmon et al. (2021). Our hard instance differs from previous ones mainly in the large Lipschitz constant of its gradient. We note that the controlled degree of nonsmoothness is crucial for our estimate of PL constant. In Lemma 1 we list some important properties of $f_{T,c,\sigma}$.

Lemma 1. *If $\sigma < 1$ and $T^c \geq \frac{1}{2}\sigma^{-1}$, then $f_{T,c,\sigma}$ satisfies the following.*

1. *$g_{T,c,\sigma}(\mathbf{x}) = f_{T,c,\sigma}(\mathbf{1} - \mathbf{x})$ is a zero-chain.*
2. *$\mathbf{x}^* = \mathbf{0}$, $f^*_{T,c,\sigma} = 0$, $f_{T,c,\sigma}(x) \leq \frac{1}{2}\mathbf{x}^T(\mathbf{A} + \sigma\mathbf{I})\mathbf{x}$.*
3. *$f_{T,c,\sigma}$ is $(3 + 32\sigma T^c)$-smooth.*
4. *$f_{T,c,\sigma}$ satisfies the $\frac{1}{C_1 T^{1+5c}}$-PL condition, where C_1 is a universal constant.*

We give a proof sketch of Lemma 1 in Section 5.1, and defer the formal proof in Appendix A .

4.2 Lower Complexity Bounds

We study the lower bound of linear-span algorithms first. Let $\tilde{f}$ denote the following function:

$$\tilde{f}(\mathbf{x}) = \frac{LT^{-1}D^2}{42\sigma T^c} f_{T,c,\sigma}\left(\mathbf{1} - T^{1/2}D^{-1}\mathbf{x}\right), \tag{19}$$

where T and σ are parameters to be specified later.

In Lemma 2, we show that $\tilde{f}$ is difficult to optimize for all first-order linear-span algorithms.

Lemma 2. *Under a specific choice or parameters T and σ, a first-order linear-span algorithm with $\mathbf{x}^{(0)} = \mathbf{0}$ needs at least $\frac{T}{2}$ gradient computations to find a point $\mathbf{x}$ satisfying $\tilde{f}(\mathbf{x}) - \tilde{f}^* \leq \frac{1}{16}(\tilde{f}(\mathbf{x}^{(0)}) - \tilde{f}^*)$.*

We give the full proof of Lemma 2 in Appendix A . The key point is that each gradient access of a linear-span algorithm only reveals one coordinate of a zero-chain. For $i > \frac{T}{2}$, $\mathbf{x}_i^{(k)}$ remains unchanged when $k \leq \frac{T}{2}$, which gives a lower bound of the function value after the first $\frac{T}{2}$ gradient computations.

With Lemma 1 and Lemma 2 in hand, we are ready to give our lower complexity bound for linear-span first-order algorithms.

Theorem 2. *Given $L \geq \mu > 0$ and $c < 0.01$. When $\kappa = \frac{L}{\mu} > C_2$ where C_2 is a universal constant, there exists T and σ such that $\tilde{f}$ is L-smooth and μ-PL. Moreover, any first-order linear-span algorithm with $\mathbf{x}^{(0)} = \mathbf{0}$ needs at least $O\left(\kappa^{1/(1+6c)}\right)$ gradient computations to find a point $\mathbf{x}$ satisfying $\tilde{f}(\mathbf{x}) - \tilde{f}^* \leq \frac{1}{16}(\tilde{f}(\mathbf{x}^{(0)}) - \tilde{f}^*)$.*

The proof of Theorem 2 is provided in Section 5.2. For any $0 < a < 1$, let $c = \frac{a}{6(1-a)}$, then by Theorem 2, any linear-span algorithm needs at least $O(\kappa^{1-a})$ gradient computations to find an ε-approximate optimal point of the function $\tilde{f}$.

Using the technique of Nemirovskij & Yudin (1983), for specific function classes such as PL functions, a lower complexity bound on first-order linear-span algorithms is also a lower complexity bound on all the first-order algorithms. Denoting the set of all first-order algorithms by $\mathcal{A}^{(1)}$, we have the following lemma:

Lemma 3.

$$\mathcal{T}_\varepsilon\left(\mathcal{A}^{(1)}, \mathcal{P}(\Delta, \mu, L)\right) \geq \mathcal{T}_\varepsilon\left(\mathcal{A}_{\text{ls}}, \mathcal{P}(\Delta, \mu, L)\right). \tag{20}$$

Note that given L and μ, the value $\tilde{f}(\mathbf{0})$ can be controlled by choosing an appropriate D. Lemma 2 and 3 leads to the main result of the paper:

Theorem 3. *For any $0 < a < 1$, when $\varepsilon \leq \frac{1}{16}\Delta$,*

$$\mathcal{T}_\varepsilon\left(\mathcal{A}^{(1)}, \mathcal{P}(\Delta, \mu, L)\right) \geq O(\kappa^{1-a}) \tag{21}$$

This lower bound nearly matches the upper bound $O\left(\kappa \log \frac{1}{\varepsilon}\right)$ in the term of κ.

5 Proof of Lemma 1 and Theorem 2

5.1 Proof Sketch of Lemma 1

Properties 1, 2 and 3 in Lemma 1 follow from direct computation. The key to proving property 4 is the T^{5c} term in the denominator of the PL constant. From the assumptions $\sigma < 1$ and $T \geq \frac{1}{2}\sigma^{-1/c}$ in Lemma 1, we know that T is large enough compared to σ, which makes the desired PL constant relatively small. To prove property 4, we consider the following cases:

- if $\max|\mathbf{x}_i| < 1 - \frac{1}{32}T^{-c}$, then for every i, $b_{T,c}(\mathbf{x}_i) = 0$. Now we estimate the PL constant. From its definition, we have
$$\frac{\|\nabla f_{T,c,\sigma}(\mathbf{x})\|^2}{2f_{T,c,\sigma}(\mathbf{x})} \geq \frac{\|(\mathbf{A}+\sigma\mathbf{I})\mathbf{x}\|^2}{\mathbf{x}^T(\mathbf{A}+\sigma\mathbf{I})\mathbf{x}} \geq \sigma. \tag{22}$$
This suggests that the PL constant is greater than $\frac{1}{C_1 T^{1+5c}}$ when T is sufficiently large.
- If $\max|\mathbf{x}_i| > 1 + \frac{1}{32}T^{-c}$, let $\mathbf{x}_k = \max|\mathbf{x}_i| := m$. In this case, we have the following lemma:
Lemma 4. *If $\mathbf{x}_k = \max|\mathbf{x}_i| := m$ and $m \geq 1 + \frac{1}{32}T^{-c}$, then*
$$\|\nabla f_{T,c,\sigma}(\mathbf{x})\| \geq \left|\left(\nabla f_{T,c,\sigma}(\mathbf{x})\right)_k\right| \geq \frac{1}{240}T^{-(1+2c)/2}\sigma\|(\mathbf{A}+\sigma\mathbf{I})\mathbf{x}\|. \tag{23}$$
At the same time, the function value $f_{T,c,\sigma}(\mathbf{x})$ can be upper bounded by $\frac{1}{2}\mathbf{x}^T(\mathbf{A}+\sigma\mathbf{I})\mathbf{x}$. Using the estimate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ and $f_{T,c,\sigma}(\mathbf{x})$, the PL constant can be shown to be greater than $\frac{1}{C_1 T^{1+5c}}$.
- If $1 - \frac{1}{32}T^{-c} \leq \max|\mathbf{x}_i| \leq 1 + \frac{1}{32}T^{-c}$, we first present two auxiliary lemmas below.
Lemma 5. *If $\sigma < 1$, $T^c \geq \frac{1}{2}\sigma^{-1}$, $1 - \frac{1}{32}T^{-c} \leq \max|\mathbf{x}_i| \leq 1 + \frac{1}{32}T^{-c}$, and*
$$\|\nabla f_{T,c,\sigma}(\mathbf{x})\| < \frac{1}{240}T^{-(1+2c)/2}\sigma\|(\mathbf{A}+\sigma\mathbf{I})\mathbf{x}\|, \tag{24}$$
then there exists a consecutive sequence of integers $k, k+1, \cdots, l$, such that
$$\mathbf{x}_{l+1} \geq 1 - \frac{1}{32}T^{-c}, \tag{25}$$
and
$$\mathbf{x}_{i+1} \geq \left(1 + \frac{\sigma}{2}\right)\mathbf{x}_i \tag{26}$$
for $i = k, \cdots, l$.

Proof sketch. If the PL constant is small as (24) assumes, then every coordinate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ must not be too far from 0. Consider a consecutive sequence of positive integers $k, k+1, \cdots, l$, where $\mathbf{x}_i < 1 - \frac{1}{32}T^{-c}$ for $i = k, \cdots, l$, and $\mathbf{x}_{l+1} \geq 1 - T^{-c}$. We have $b_{T,c}(\mathbf{x}_i) = 0$, so the ith coordinate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ is

$$-\mathbf{x}_{i-1} + (2+\sigma)\mathbf{x}_i - \mathbf{x}_{i+1}. \tag{27}$$

We assumed $T^c \geq \frac{1}{2}\sigma^{-1}$ in Lemma 5, which is large enough so that every coordinate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ must be small compared to σ. Therefore, the sequence $\mathbf{x}_k, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}$ grows exponentially, or some of the coordinates of $\nabla f_{T,c,\sigma}(\mathbf{x})$ would be too large. □

Lemma 6. *If $\sigma < 1$, $T \geq \frac{1}{2}\sigma^{-1/c}$, $\mathbf{x}_{l+1} \geq 1 - \frac{1}{32}T^{-c}$ and $\mathbf{x}_{l+1} \geq \left(1 + \frac{\sigma}{2}\right)\mathbf{x}_l$, then*

$$\|\nabla f_{T,c,\sigma}(\mathbf{x})\| \geq (\nabla f_{T,c,\sigma}(\mathbf{x}))_{l+1} \geq \frac{\sigma}{16}. \tag{28}$$

Proof sketch. The $(l+1)$th coordinate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ is

$$(\nabla f_{T,c,\sigma}(\mathbf{x}))_{l+1} = -\mathbf{x}_l + 2\mathbf{x}_{l+1} - \mathbf{x}_{l+2} + \sigma\mathbf{x}_{l+1} - \sigma b_{T,c}(\mathbf{x}_{l+1}). \tag{29}$$

Because $\mathbf{x}_{l+1} \geq 1 - \frac{1}{32}T^{-c}$,

$$\begin{aligned} \mathbf{x}_{l+1} - \mathbf{x}_l &\geq \left(1 - \frac{1}{1+\sigma/2}\right)\mathbf{x}_{l+1} \\ &\geq \frac{\sigma}{4}\mathbf{x}_{l+1}. \end{aligned} \tag{30}$$

The gap between $\mathbf{x}_{l+1}$ and $\mathbf{x}_l$ depends only on σ, while the gap between $\mathbf{x}_{l+2}$ and $\mathbf{x}_{l+1}$ is smaller than $\frac{1}{16}T^{-c}$. As we have assumed that T is large compared to σ, $\mathbf{x}_{l+1} - \mathbf{x}_l$ dominates $\mathbf{x}_{l+2} - \mathbf{x}_{l+1}$ and $-\sigma(\mathbf{x}_{l+1} - b_{T,c}(\mathbf{x}_{l+1}))$. Therefore, $-\mathbf{x}_l + 2\mathbf{x}_{l+1} - \mathbf{x}_{l+2} + \sigma\mathbf{x}_{l+1} - \sigma b_{T,c}(\mathbf{x}_{l+1})$ will be greater than 0 by a margin depending only on σ. The absolute value of the $(l+1)$th coordinate of $\nabla f_{T,c,\sigma}(\mathbf{x})$ gives a lower bound of $\|\nabla f_{T,c,\sigma}(\mathbf{x})\|$. □

Detailed presentation and proof of Lemma 5 and Lemma 6 is provided in Appendix A . For $\mathbf{x} \in \mathbb{R}^T$, we may use one of the lemmas above to compute the PL constant at $\mathbf{x}$, and give a universal lower bound of PL constant which leads to Lemma 1.

5.2 Proof of Theorem 2

Proof. Define $C_2 = 4200C_1$, where C_1 is the constant defined in Lemma 1. Given c and κ, we let

$$T \in \left(\left(\frac{100}{C_2}\kappa\right)^{1/(1+6c)}, \left(\frac{200}{C_2}\kappa\right)^{1/(1+5c)}\right) \cap \mathbb{Z} \tag{31}$$

and

$$\sigma = \frac{100}{C_2}\kappa T^{-(1+6c)}. \tag{32}$$

Because $\kappa > C_2$ and $c < 0.01$, we have

$$\begin{aligned} &\left(\frac{200}{C_2}\kappa\right)^{1/(1+5c)} - \left(\frac{100}{C_2}\kappa\right)^{1/(1+6c)} \\ &\geq 200^{1/1.05} - 100 \\ &> 150 - 100 \\ &= 50, \end{aligned} \tag{33}$$

so there exists T satisfying (31).

It can be verified that T and σ in (31) and (32) satisfy the assumptions in Lemma 1. Therefore, we can use Lemma 1 to calculate the smooth constant and PL constant of $\tilde{f}$.

The smooth constant of $\tilde{f}$ is:

$$\frac{LT^{-1}D^2}{42\sigma T^c} \cdot TD^{-2} \cdot 42\sigma T^c = L, \tag{34}$$

and the PL constant of $\tilde{f}$ is:

$$\begin{aligned} \frac{LT^{-1}D^2}{42\sigma T^c} \cdot TD^{-2} \cdot \frac{1}{C_1 T^{1+5c}} &= \frac{100L}{C_2 T^{1+6c}\sigma} \\ &\overset{a}{=} \frac{100L}{C_2 T^{1+6c}} \cdot \frac{C_2 T^{1+6c}}{100\kappa} \\ &= \frac{L}{\kappa} \\ &= \mu, \end{aligned} \tag{35}$$

where $\overset{a}{=}$ uses (32).

Finally, by lemma 2, any first-order linear-span algorithm needs at least $\frac{T}{2}$ access to gradient to find $\mathbf{x}$ such that $\tilde{f}(x) - \tilde{f}^* \leq \frac{1}{16}\left(\tilde{f}(\mathbf{x}^{(0)}) - \tilde{f}^*\right)$. By (31),

$$\frac{T}{2} \geq \frac{1}{2}\left(\frac{100\kappa}{C_2}\right)^{1/(1+6c)} = O\left(\kappa^{1/(1+6c)}\right), \tag{36}$$

which completes the proof. □

6 Conclusion

We construct a lower complexity bound on optimizing smooth PL functions with first-order methods. For arbitrary $0 < a < 1$, a first-order algorithm needs at least $O\left(\left(\frac{L}{\mu}\right)^{1-a}\right)$ gradient access to find an ε-approximate optimal point of an L-smooth μ-PL function. Our lower bound nearly matches the convergence rate of Gradient Descent.

References

Mihai Anitescu. Degenerate nonlinear programming with a quadratic growth condition. *SIAM Journal on Optimization*, 10(4):1116–1135, 2000.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*, pp. 242–299. PMLR, 2020.

Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pp. 1467–1476. PMLR, 2018.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29, 2016.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. *Advances in Neural Information Processing Systems*, 30, 2017.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47. PMLR, 2018.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 2022.

Ji Liu, Steve Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. In *International Conference on Machine Learning*, pp. 469–477. PMLR, 2014.

Stanislaw Lojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2, 1963.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Technical report, 1985.

Ion Necoara, Yu Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

B.T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. *Advances in neural information processing systems*, 29, 2016.

Hui Zhang and Wotao Yin. Gradient methods for convex minimization: better rates under weaker conditions. *arXiv preprint arXiv:1303.4645*, 2013.

Dongruo Zhou and Quanquan Gu. Lower bounds for smooth nonconvex finite-sum optimization. In *International Conference on Machine Learning*, pp. 7574–7583. PMLR, 2019.