A Lower Bound for the Optimization of Finite Sums

Alekh Agarwal ALEKHA@MICROSOFT.COM
Microsoft Research NYC, New York, NY.

Léon Bottou LEON@BOTTOU.ORG
Facebook AI Research, New York, NY.

Abstract

This paper presents a lower bound for optimizing a finite sum of n functions, where each function is L-smooth and the sum is μ-strongly convex. We show that no algorithm can reach an error ε in minimizing all functions from this class in fewer than $\Omega(n+\sqrt{n(\kappa-1)}\log(1/\varepsilon))$ iterations, where $\kappa = L/\mu$ is a surrogate condition number. We then compare this lower bound to upper bounds for recently developed methods specializing to this setting. When the functions involved in this sum are not arbitrary, but based on i.i.d. random data, then we further contrast these complexity results with those for optimal first-order methods to directly optimize the sum. The conclusion we draw is that a lot of caution is necessary for an accurate comparison, and identify machine learning scenarios where the new methods help computationally.

1. Introduction

Many machine learning setups lead to the minimization a convex function of the form

$$x_f^* = \arg\min_{x\in\mathcal{X}} f(x), \text{ with } f(x) = \frac{\mu}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^{n} g_i(x), \quad (1)$$

where $\mathcal{X}$ is a convex, compact set. When the functions g_i are also convex, then the overall optimization problem is convex, and can in principle be solved using any off-the-shelf convex minimization procedure. In the machine learning literature, two primary techniques have typically been used to address such convex optimization problems. The first approach (called the batch approach) uses the ability to evaluate the function f along with its gradients, Hessian etc. and applies first- and second-order methods to minimize the objective. The second approach (called the stochastic approach) interprets the average in Equation (1) as an expectation and uses stochastic gradient methods, randomly sampling a g_i and using its gradient and Hessian information as unbiased estimates for those of the function f.[1] Both these classes of algorithms have extensive literature on upper bounds for the complexities of specific methods. More fundamentally, there are also lower bound results on the minimum black-box complexity of the *best-possible algorithm* to solve convex minimization problems. In several broad problem classes, these lower bounds further coincide with the known upper bounds for specific methods, yielding a rather comprehensive general theory.

However, a recent line of work in the machine learning literature, recognizes that the specific problem (1) of interest has additional structure beyond a general convex minimization problem. For instance, the average in defining the function f is over a fixed number n of functions, whereas typical complexity results on stochastic optimization allow for the expectation to be with respect to a continuous random variable. Recent works (Le Roux et al., 2012; Shalev-Shwartz & Zhang, 2013; Johnson & Zhang, 2013) make further assumptions that the functions g_i involved in this sum are smooth, and the function f is of course strongly convex by construction. Under these conditions, the algorithms studied in these works have the following properties: (i) the cost of each iteration is identical to stochastic optimization methods, and (ii) the convergence rate of the method is linear.[2] The results are surprising since the existing lower bounds on stochastic optimization dictate that the error can decrease no faster than $\Omega(1/k)$ after k iterations under such assumptions (Nemirovsky & Yudin, 1983), leaving an exponential gap compared to these new results. It is of course not a contradiction due to the finite

[1] There is a body of literature that recognizes the ability of stochastic optimization to minimize testing error rather than training error in machine learning contexts (see e.g. Bottou & Bousquet, 2008), but we will focus on training error for this paper.

[2] An optimization algorithm is linearly convergent if it reduces the sub-optimality by a constant factor at each iteration.

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

sum structure of the problem (1) (following the terminology of Bertsekas (2012), we will call the setup of optimizing a finite sum *incremental optimization* hereafter).

Given this recent and highly interesting line of work, it is natural to ask just how much better can one do in this model of minimizing finite sums. Put another way, can we specialize the existing lower bounds for stochastic or batch optimization, to yield results for this new family of functions. The aim of such a result would be to understand the fundamental limits on any possible algorithm for this family of problems, and whether better algorithms are possible at all than the existing ones. Answering such questions is the goal of this work. To this end, we define the Incremental First-order Oracle (IFO) complexity model, where an algorithm picks an index $i \in \{1,2,\ldots,n\}$ and a point $x \in \mathcal{X}$ and the oracle returns $g_i'(x)$. We consider the setting where each function g_i is L-smooth (that is, it has L-Lipschitz continuous gradients). In this setting, we demonstrate that no method can achieve $\|x_K - x_f^*\| \leq \varepsilon \|x_f^*\|$ for all functions f of the form (1), without performing $K = \Omega\big(n + \sqrt{n(L/\mu - 1)}\log(1/\varepsilon)\big)$ calls to the IFO. As we will discuss following this main result, this lower bound is not too far from upper bounds for IFO methods such as SAG, SVRG and SAGA (Schmidt et al., 2013; Johnson & Zhang, 2013; Defazio et al., 2014) whose iteration complexity is $\mathcal{O}((n + L/\mu)\log(1/\varepsilon))$. Some dual coordinate methods such as ASDCA and SPDC (Shalev-Shwartz & Zhang, 2014; Zhang & Xiao, 2014) get even closer to the lower bound, but are not IFO algorithms. Overall, there is no method with a precisely matching upper bound on its complexity, meaning that there is further room for improving either the upper or the lower bounds for this class of problems.

Following the statement of our main result, we will also discuss the implications of these lower bounds for the typical machine learning problems that have inspired this line of work. In particular, we will demonstrate that caution is needed in comparing the results between the standard first-order and IFO complexity models, and worst-case guarantees in the IFO model might not adequately capture the performance of the resulting methods in typical machine learning settings. We will also demonstrate regimes in which different IFO methods as well as standard first-order methods have their strengths and weaknesses.

Recent work of Arjevani (2014) also studies the problem of lower bounds on smooth and strongly convex optimization methods, although their development focuses on certain restricted subclasses of first-order methods (which includes SDCA but not the accelerated variants, for instance). Discussion on the technical distinctions in the two works is presented following our main result.

As a prerequisite for our result, we need the result on black-box first-order complexity of minimizing smooth and strongly convex functions. We provide a self-contained proof of this result in our paper in Appendix A which might be of independent interest. In fact, we establish a slight variation on the original result, in order to help prove our main result. Our main result will invoke this construction multiple times to design each of the components g_i in the optimization problem (1).

The remainder of this paper is organized as follows. The next section formally describes the complexity model and the structural assumptions. We then state the main result, followed by a discussion of consequences for typical machine learning problems. The proofs are deferred to the subsequent section, with the more technical details in the supplement.

2. Setup and main result

Let us begin by formally describing the class of functions we will study in this paper. Recall that a function g is called L-smooth, if it has L-Lipschitz continuous gradients, that is

$$\forall x, y \in \mathcal{X} \quad \|g'(x) - g'(y)\|_* \leq L\|x - y\| \,,$$

where $\|\cdot\|_*$ is the norm dual to $\|\cdot\|$. In this paper, we will only concern ourselves with scenarios where $\mathcal{X}$ is a convex subset of a separable Hilbert space, with $\|\cdot\|$ being the (self-dual) norm associated with the inner product. A function g is called μ-strongly convex if

$$\forall x, y \in \mathcal{X} \quad g(y) \geq g(x) + \langle g'(x), y - x\rangle + \frac{\mu}{2}\|x - y\|^2.$$

Given these definitions, we now define the family of functions being studied in this paper.

Definition 1 *Let $\mathcal{F}_n^{\mu,L}(\Omega)$ denote the class of all convex functions f with the form (1), where each g_i is $(L-\mu)$-smooth and convex.*

Note that f is μ-strongly convex and L-smooth by construction, and hence $\mathcal{F}_n^{\mu,L}(\Omega) \subseteq \mathcal{S}^{\mu,L}(\Omega)$ where $\mathcal{S}^{\mu,L}(\Omega)$ is the set of all μ-strongly convex and L-smooth functions. However, as we will see in the sequel, it can often be a much smaller subset, particularly when the smoothness of the global function is much better than that of the local functions. We now define a natural oracle for optimization of functions with this structure, along with admissible algorithms.

Definition 2 (Incremental First-order Oracle (IFO)) *For a function $f \in \mathcal{F}_n^{\mu,L}(\Omega)$, the Incremental First-order Oracle (IFO) takes as input a point $x \in \mathcal{X}$ and index $i \in \{1,2,\ldots,n\}$ and returns the pair $(g_i(x), g_i'(x))$.*

Definition 3 (IFO Algorithm) *An optimization algorithm is an IFO algorithm if its specification does not depend on the cost function f other than through calls to an IFO.*

For instance, a standard gradient algorithm would take the current iterate x_k, and invoke the IFO with (x_k, i) in turn with $i = \{1, 2, \ldots, n\}$, in order to assemble the gradient of f. A stochastic gradient algorithm would take the current iterate x_k along with a randomly chosen index i as inputs to IFO. Most interesting to our work, the recent SAG, SVRG and SAGA algorithms (Le Roux et al., 2012; Johnson & Zhang, 2013; Defazio et al., 2014) are IFO algorithms. On the other hand, dual coordinate ascent algorithms require access to the gradients of the conjugate of f_i, and therefore are not IFO algorithms.

We now consider IFO algorithms that invoke the oracle K times (at $x_0, \ldots, x_{K-1}$) and output an estimate x_K of the minimizer x_f^*. Our goal is to bound the smallest number of queries K needed for any method to ensure an error $\|x_K - x_f^*\| \leq \varepsilon \|x_f^*\|$, uniformly for all $f \in \mathcal{F}_n^{\mu,L}(\Omega)$. This complexity result will depend on the ratio $\kappa = L/\mu$ which is analogous to the condition number that usually appears in complexity bounds for the optimization of smooth and strongly convex functions. Note that κ is strictly an upper bound on the condition number of f, but also the best one in general given the structural information about $f \in \mathcal{F}_n^{\mu,L}(\Omega)$.

In order to demonstrate our lower bound, we will make a specific choice of the problem domain $\mathcal{X}$. Let ℓ_2 be the Hilbert space of real sequences $x = (x_{[i]})_{i=1}^{\infty}$ with finite norm $\|x\|^2 = \sum_{i=1}^{\infty} {x_{[i]}}^2$, and equipped with the standard inner product $\langle x, y \rangle = \sum_{i=1}^{\infty} x_{[i]} y_{[i]}$. We are now in a position to state our main result over the complexity of optimization for the function class $\mathcal{F}_n^{\mu,L}(\ell_2)$.

Theorem 1 *Consider an IFO algorithm for problem (1) that performs $K \geq 0$ calls to the oracle and output a solution x_K. Then, for any $\gamma > 0$, there exists a function $f \in \mathcal{F}_n^{\mu,L}(\ell_2)$ such that $\|x_f^*\| = \gamma$ and*

$$\|x_f^* - x_K\| \;\geq\; \gamma q^{2t} \quad \text{with} \quad q = \frac{\sqrt{1 + \frac{\kappa-1}{n}} - 1}{\sqrt{1 + \frac{\kappa-1}{n}} + 1}\,, \quad \kappa = \frac{L}{\mu}\,,$$

and with $t = \begin{cases} 0 & \text{if } K < n. \\ K/n & \text{otherwise.} \end{cases}$

In order to better interpret the result of the theorem, we state the following direct corollary which lower bounds the number of steps need to attain an accuracy of $\varepsilon \|x_f^*\|$.

Corollary 1 *Consider an IFO algorithm for problem (1) that guarantees $\|x_f^* - x_K\| \leq \varepsilon \|x_f^*\|$ for any $\varepsilon < 1$. Then there is a function $f \in \mathcal{F}_n^{\mu,L}(\ell_2)$ on which the algorithm must perform at least $K = \Omega(n + \sqrt{n(\kappa-1)}\log(1/\varepsilon))$ IFO calls.*

The first term in the lower bound simply asserts that any optimization method needs to make at least one query per g_i, in order to even see each component of f which is clearly necessary. The second term, which is more important since it depends on the desired accuracy ε, asserts that the problem becomes harder as the number of elements n in the sum increases or as the problem conditioning worsens. Again, both these behaviors are qualitatively expected. Indeed as $n \to \infty$, the finite sum approaches an integral, and the IFO becomes equivalent to a generic stochastic-first order oracle for f, under the constraint that the stochastic gradients are also Lipschitz continuous. Due to $\Omega(1/\varepsilon)$ complexity of stochastic strongly-convex optimization (with no dependence on n), we do not expect the linear convergence of Corollary 1 to be valid as $n \to \infty$. Also, we certainly expect the problem to get harder as the ratio L/μ degrades. Indeed if all the functions g_i were identical, whence the IFO becomes equivalent to a standard first-order oracle, the optimization complexity similarly depends on $\Omega(\sqrt{\kappa - 1}\log(1/\varepsilon))$.

Whenever presented with a lower bound, it is natural to ask how it compares with the upper bounds for existing methods. We now compare our lower bound to upper bounds for standard optimization schemes for $\mathcal{S}^{\mu,L}(\ell_2)$ as well as specialized ones for $\mathcal{F}_n^{\mu,L}(\ell_2)$. We specialize to $\|x_f^*\| = 1$ for this discussion.

Comparison with optimal gradient methods: As mentioned before, $\mathcal{F}_n^{\mu,L}(\ell_2) \subseteq \mathcal{S}^{\mu,L}(\ell_2)$, and hence standard methods for optimization of smooth and strongly convex objectives apply. These methods need n calls to the IFO for getting the gradient of f, followed by an update. Using Nesterov's optimal gradient method (Nesterov, 2004), one needs at most $\mathcal{O}(\sqrt{\kappa}\log(1/\varepsilon))$ gradient evaluations to reach ε-optimal solution for $f \in \mathcal{S}^{\mu,L}(\ell_2)$, resulting in at most $\mathcal{O}(n\sqrt{\kappa}\log(1/\varepsilon))$ calls to the IFO. Comparing with our lower bound, there is a suboptimality of at most $\mathcal{O}(\sqrt{n})$ in this result. Since this is also the best possible complexity for minimizing a general $f \in \mathcal{S}^{\mu,L}(\ell_2)$, we conclude that there might indeed be room for improvement by exploiting the special structure here. Note that there is an important caveat in this comparison. For f of the form (1), the smoothness constant for the overall function f might be much smaller than L, and the strong convexity term might be much higher than μ due to further contribution from the g_i. In such scenarios. the optimal gradient methods will face a much smaller condition number κ in their complexity. This issue will be discussed in more detail in Section 3.

Comparison with the best known algorithms: At least three algorithms recently developed for problem setting (1) offer complexity guarantees that are close to our lower bound. SAG, SVRG and SAGA (Le Roux et al., 2012; Johnson & Zhang, 2013; Defazio et al., 2014) all reach an optimization error ε after less than $\mathcal{O}((n+\kappa)\log(1/\varepsilon))$ calls to the oracle. There are two differences from our lower bound. The first term of n multiplies the $\log(1/\varepsilon)$ term in the upper bounds, and the condition number dependence is $O(\kappa)$ as opposed to $O(\sqrt{n\kappa})$. This suggests that there is room to either improve the lower bound, or for algorithms with a better complexity. As observed earlier, the ASDCA and SPDC methods (Shalev-Shwartz & Zhang, 2014; Zhang & Xiao, 2014) reach a closer upper bound of $\mathcal{O}((n+\sqrt{n(\kappa-1)})\log(1/\varepsilon))$, but these methods are not IFO algorithms.

Room for better lower bounds? One natural question to ask is whether there is a natural way to improve the lower bound. As will become clear from the proof, a better lower bound is not possible for the *hard problem instance* which we construct. Indeed for the quadratic problem we construct, conjugate gradient descent can be used to solve the problem with a nearly matching upper bound. Hence there is no hope to improve the lower bounds without modifying the construction.

It might appear odd that the lower bound is stated in the infinite dimensional space ℓ_2. Indeed this is essential to rule out methods such as conjugate gradient descent solving the problem exactly in a finite number of iterations depending on the dimension only (without scaling with ε). An alternative is to rule out such methods, which is precisely the approach Arjevani (2014) takes. On the other hand, the resulting lower bounds here are substantially stronger, since they apply to a broader class of methods. For instance, the restriction to stationary methods in Arjevani (2014) makes it difficult to allow any kind of adaptive sampling of the component functions f_i as the optimization progresses, in addition to ruling out methods such as conjugate gradient.

3. Consequences for optimization in machine learning

With all the relevant results in place now, we will compare the efficiency of the different available methods in the context of solving typical machine learning problems. Recall the definitions of the constants L and μ from before. In general, the full objective f (1) has its own smoothness and strong convexity constants, which need not be the same as L and μ. To that end, we define L_f to be the smoothness constant of f, and μ_f to the strong convexity of f. It is immediately seen that L provides an upper bound on L_f, while μ provides a lower bound on μ_f.

Algorithm	Batch complexity	Adaptive?
ASDCA, SDPC (Shalev-Shwartz & Zhang, 2014) (Zhang & Xiao, 2014)	$\tilde{\mathcal{O}}\left(\left(1+\sqrt{\frac{L-\mu}{\mu n}}\right)\log\frac{1}{\varepsilon}\right)$	no
SAG (Schmidt et al., 2013)	$\tilde{\mathcal{O}}\left(\left(1+\frac{L}{\mu_f n}\right)\log\frac{1}{\varepsilon}\right)$	to μ_f
AGM† (Nesterov, 2007)	$\tilde{\mathcal{O}}\left(\sqrt{\frac{L_f}{\mu_f}}\log\frac{1}{\varepsilon}\right)$	to μ_f, L_f

Table 1. A comparison of the batch complexities of different methods. A method is adaptive to μ_f or L_f, if it does not need the knowledge of these parameters to run the algorithm and obtain the stated complexity upper bound. †Although the simplest version of AGM does require the specification of μ_f and L_f, Nesterov also discusses an adaptive variant with the same bound up to additional logarithmic factors.

In order to provide a meaningful comparison for incremental as well as batch methods, we follow Zhang & Xiao (2014) and compare the methods in terms of their *batch complexity*, that is, how many times one needs to perform n calls to the IFO in order to ensure that the optimization error for the function f is smaller than ε. When defining batch complexity, Zhang & Xiao (2014) observed that the incremental and batch methods have dependence on L versus L_f, but did not consider the different strong convexities that play a part for different algorithms. In this section, we also include the dual coordinate methods in our comparison since they are computationally interesting for the problem (1) even though they are not admissible in the IFO model. Doing so, the batch complexities can be summarized as in Table 1.

Based on the table, we see two main points of difference. First, the incremental methods rely on the smoothness of the individual components. That this is unavoidable is clear, since even the worst case lower bound of Theorem 1 depends on L and not L_f. As Zhang & Xiao (2014) observe, L_f can in general be much smaller than L. They attempt to address the problem to some extent by using non-uniform sampling, thereby making sure that the best of the g_i and the worst of the g_i have a similar smoothness constant under the reweighing. This does not fully bridge the gap between L and L_f as we will show next. However, more striking is the difference in the lower curvature across methods. To the best of our knowledge, all the existing analyses of coordinate ascent require a clear isolation of strong convexity, as in the function definition (1). These methods then rely on using μ as an estimate of the curvature of f, and cannot adapt to any additional curvature when μ_f is much larger than μ. Our next example shows this can be a serious concern for many machine learning problems.

In order to simplify the following discussion we restrict ourselves to perhaps the most basic machine learning optimization problem, the regularized least-squares regression:

$$f(x) = \frac{\mu}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^{n} g_i(x) \text{ with } g_i(x) = (\langle a_i, x\rangle - b_i)^2, \tag{2}$$

where a_i is a data point and b_i is a scalar target for prediction. It is then easy to see that $g_i''(x) = a_i a_i^\top$ so that $f \in \mathcal{F}_n^{\mu,L}(\Omega)$ with $L = \max_i(\mu + \|a_i\|^2)$. To simplify the comparisons, assume that $a_i \in \mathbb{R}^d$ are drawn independently from a distribution defined on the sphere $\|a_i\| = R$. This ensures that $L = \mu + R^2$. Since each function g_i has the same smoothness constant, the importance sampling techniques of Zhang & Xiao (2014) cannot help.

In order to succinctly compare algorithms, we use the notation Γ_{ALG} to represent the batch complexity of ALG without the $\log(1/\varepsilon)$ term, which is common across all methods. Then we see that the upper bound for Γ_{ASDCA} is

$$\Gamma_{\text{ASDCA}} \;=\; 1 + \sqrt{\frac{\kappa - 1}{n}} \;=\; 1 + \sqrt{\frac{R^2}{\mu n}}. \tag{3}$$

In order to follow the development of Table 1 for SAG and AGM, we need to evaluate the constants μ_f and L_f. Note that in this special case, the constants L_f and μ_f are given by the upper and lower eigenvalues respectively of the matrix $\mu I + \hat{\Sigma}$, where $\hat{\Sigma} = \sum_{i=1}^n a_i a_i^\top / n$ represents the empirical covariance matrix. In order to understand the scaling of this empirical covariance matrix, we shall invoke standard results on matrix concentration.

Let $\Sigma = \mathbb{E}[a_i a_i^\top]$ be the second moment matrix of the a_i distribution. Let $\lambda_{\min}$ and $\lambda_{\max}$ be its lowest and highest eigenvalues. Let us define the condition number of the penalized population objective

$$\kappa_f \;\stackrel{\Delta}{=}\; \frac{\mu + \lambda_{\max}}{\mu + \lambda_{\min}}\,.$$

Equation (5.26) in (Vershynin, 2012) then implies that there are universal constants c and C such that the following inequality holds with probability $1 - \delta$:

$$\|\Sigma - \hat{\Sigma}\| \;\leq\; \|\Sigma\| \max\left(z, z^2\right) \text{ with } z = c\sqrt{\frac{d}{n}} + C\sqrt{\frac{\log(2/\delta)}{n}}\,.$$

Let us weaken the above inequality slightly to use $\mu + \|\Sigma\|$ instead of $\|\Sigma\|$ in the bound, which is minor since we typically expect $\mu \ll \lambda_{\max}$ for statistical consistency. Then assuming we have enough samples to ensure that

$$c^2\frac{d}{n} + C^2\frac{\log(d/\delta)}{n} \leq \frac{1}{8\kappa_f^2}, \tag{4}$$

we obtain the following bounds on the eigenvalues of the sample covariance matrix :

$$\mu_f \geq \max\left\{\mu,\, \mu + \lambda_{\min} - \lambda_{\max}\max\left(z, z^2\right)\right\} \geq \tfrac{\mu + \lambda_{\min}}{2},$$
$$L_f \leq \min\left\{L,\, \mu + \lambda_{\max} + \lambda_{\max}\max\left(z, z^2\right)\right\} \leq \tfrac{3(\mu + \lambda_{\max})}{2}\,,$$

Using these estimates in the bounds of Table 1 gives

$$\Gamma_{\text{SAG}} \;=\; 1 + \frac{L}{\mu_f n} \;\leq\; 1 + \frac{2(\mu + R^2)}{n(\mu + \lambda_{\min})} \;=\; \mathcal{O}(1)\,, \tag{5}$$

$$\Gamma_{\text{AGM}} \;=\; \sqrt{\frac{L_f}{\mu_f}} \;\leq\; \sqrt{3\kappa_f} \;=\; \mathcal{O}(\sqrt{\kappa_f})\,. \tag{6}$$

Table 2 compares the three methods under assumption (4) depending on the growth of κ.

Algorithm	$\kappa = \mathcal{O}(n)$	$\kappa \gg n$
ASDCA, SPDC (Eq. (3))	$\tilde{\mathcal{O}}\left(\log \frac{1}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{\frac{\kappa}{n}} \log \frac{1}{\varepsilon}\right)$
SAG (Eq. (5))	$\tilde{\mathcal{O}}\left(\log \frac{1}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\log \frac{1}{\varepsilon}\right)$
AGM (Eq. (6))	$\tilde{\mathcal{O}}\left(\sqrt{\kappa_f} \log \frac{1}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{\kappa_f} \log \frac{1}{\varepsilon}\right)$

Table 2. A comparison of the batch complexities of different methods for the regularized least squares objective (2) when the number of examples is sufficiently large (4). Observe how the ASDCA complexity bound can be significantly worse than the SAG complexity bound, despite its better worst case guarantee.

Problems with $\kappa = \mathcal{O}(n)$: This setting is quite interesting for machine learning, since it corresponds roughly to using $\mu = \mathcal{O}(n)$ when R^2 is a constant. In this regime, all the incremental methods seem to enjoy the best possible convergence rate of $\tilde{\mathcal{O}}(\log(1/\varepsilon))$. When the population problem is relatively well conditioned, AGM obtains a similar complexity since $\kappa_f = \mathcal{O}(1)$. However, for poorly conditioned problems, the population condition number might scale with the dimension d. We conclude that there is indeed a benefit from using the incremental methods over the batch methods in these settings, but it seems hard to distinguish between the complexities of accelerated methods like ASDCA and SPDC compared with SAG or SVRG.

Problems with large κ: In this setting, the coordinate ascent methods seem to be at a disadvantage, because the average loss term provides additional strong convexity, which is exploited by both SAG and AGM, but not by ASDCA or SPDC methods. Indeed, we find that the complexity term Γ_{ASDCA} can be made arbitrarily large as κ_i grows large. However, the contraction factors for both SAG and AGM do not grow with n in this setting, leading to a large gap

between the complexities. Between SAG and AGM, we conclude that SAG has a better bound when the population problem is poorly conditioned.

High-dimensional settings ($n/d \ll 1$): In this setting, the global strong convexity can not really be larger than μ for the function (2), since the Hessian of the averaged loss has a non-trivial null space. It would appear then, that SAG is forced to use the same problem dependent constants as ASDCA/SPDC, while AGM gets no added benefit in strong convexity either. However, in such high-dimensional problems, one is often enforcing a low-dimensional structure in machine learning settings for generalization. In such structures, the global Hessian matrix can still satisfy restricted versions of strong convexity and smoothness conditions, which are often sufficient for batch optimization methods to succeed (Agarwal et al., 2012). In such situations, the comparison might once again resemble that of Table 2, and we leave such development to the reader.

In a nutshell, the superiority of incremental algorithms for the optimization of training error in machine learning is far more subtle than suggested by their worst case bounds. Among the incremental algorithms, SAG has favorable complexity results in all regimes despite the fact that both ASDCA and SPDC offer better worst case bounds. This is largely due to the adaptivity of SAG to the curvature of the problem. This might also explain in some part the empirical observation of Schmidt et al. (2013), who find that on some datasets SDCA (without acceleration) performed significantly poorly compared with other methods (see Figure 2 in their paper for details). Finally, we also observe that SAG does indeed improve upon the complexity of AGM after taking the different problem dependent constants into account, when the population problem is ill-conditioned and the data are appropriately bounded.

It is worth observing that all our comparisons are ignoring constants, and in some cases logarithmic factors, which of course play a role in the running time of the algorithms in practice. Note also that the worst case bounds for the incremental methods account for the worst possible choice of the n functions in the sum. Better results might be possible when they are based on i.i.d. random data. Such results would be of great interest for machine learning.

4. Proof of main result

In this section, we provide the proof of Theorem 1. Our high-level strategy is the following. We will first construct the function $f : \ell_2 \mapsto \mathbb{R}$ such that each g_i acts on only the projection of a point x onto a smaller basis, with the bases being disjoint across the g_i. Since the g_i are separable, we then demonstrate that optimization of f under an IFO is equivalent to the optimization of each g_i under a standard first-order oracle. The functions g_i will be constructed so that they in turn are smooth and strongly convex with appropriate constants. Hence, we can invoke the known result for the optimization of smooth and strongly convex objectives under a first-order oracle, obtaining a lower bound on the complexity of optimizing f. We will now formalize this intuitive sketch.

4.1. Construction of a separable objective

We start with a simple definition.

Definition 4 *Let $e_1, e_2, \ldots$ denote the canonical basis vectors of ℓ_2, and let Q_i, $i = 1 \ldots n$, denote the orthonormal families $Q_i = [e_i, e_{n+i}, e_{2n+i}, \ldots, e_{kn+i}, \ldots]$.*

For ease of presentation, we also extend the transpose notation for matrices over operators in ℓ_2 in the natural manner (to avoid stating adjoint operators each time).

Definition 5 *Given a finite or countable orthonormal family $S = [s_1, s_2, \ldots] \subset \ell_2$ and $x \in \ell_2$, let*

$$Sx = \sum_{i=1}^{\infty} x_{[i]} s_i \quad \text{and} \quad S^\top x = (\langle s_i, x \rangle)_{i=1}^{\infty} ,$$

where s_i is assumed to be zero when i is greater than the size of the family.

Remark 1 *Both Sx and $S^\top x$ are square integrable and therefore belong to ℓ_2.*

Using the above notation, we first establish some simple identities for the operators Q_i defined above.

Lemma 1 *Simple calculus yields the following identities:*

$$Q_i^\top Q_i = \sum_{i=1}^{n} Q_i Q_i^\top = I, \text{ and } \|Q_i x\|^2 = \sum_{i=1}^{n} \|Q_i^\top x\|^2 = \|x\|^2 .$$

Proof We start with the first claim. For any basis vector e_j, it is easily checked that $Q_i e_j = e_{(j-1)n+i}$. By definition of $Q_i^\top$, it further follows that $Q_i^\top e_{(j-1)n+i} = e_j$. Linearity now yields $Q_i^\top Q_i x = x$ for any $x \in \ell_2$, giving the first claim. For the second claim, we observe that $Q_i Q_i^\top e_j = 0$ unless $\mathrm{mod}(j,n) = i$, in which case $Q_i Q_i^\top e_j = e_j$. This implies the second claim. The third claim now follows from the first one, since $\langle Q_i x, Q_i x \rangle = \langle x, Q_i^\top Q_i x \rangle = \|x\|^2$. Similarly the final claim follows from the second claim. ■

We now define the family of separable functions that will be used to establish our lower bound.

$$f(x) = \frac{\mu}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^{n} h_i(Q_i^\top x) , \quad h_i(x) \in \mathcal{S}^{0,L-\mu}(\ell_2) \quad (7)$$

Proposition 1 *All functions (7) belong to* $\mathcal{F}_n^{\mu,L}(\ell_2)$.

Proof We simply need to prove that the functions $g_i(x) = h_i(Q_i^\top x)$ belong to $\mathcal{S}^{0,L-\mu}(\ell_2)$. Using $g_i'(x) = Q_i h_i'(Q_i^\top x)$ and Lemma 1, we can write $\|g_i'(x) - g_i'(y)\|^2 = \left\|Q_i\left(h_i'(Q_i^\top x) - h_i'(Q_i^\top y)\right)\right\|^2 = \|h_i'(Q_i^\top x) - h_i'(Q_i^\top y)\|^2 \leq (L-\mu)^2\|Q_i^\top(x-y)\|^2 \leq (L-\mu)^2\|x-y\|^2$. ■

4.2. Decoupling the optimization across components

We would like to assert that the separable structure of f allows us to reason about optimizing its components separately. Since the h_i are not strongly convex by themselves, we first rewrite f as a sum of separated strongly convex functions. Using Lemma 1,

$$\begin{aligned} f(x) &= \frac{\mu}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^n h_i(Q_i^\top x) \\ &= \frac{\mu}{2}\sum_{i=1}^n \|Q_i^\top x\|^2 + \frac{1}{n}\sum_{i=1}^n h_i(Q_i^\top x) \\ &= \frac{1}{n}\sum_{i=1}^n \left[\frac{n\mu}{2}\|Q_i^\top x\|^2 + h_i(Q_i^\top x)\right] \triangleq \frac{1}{n}\sum_{i=1}^n f_i(Q_i^\top x)\,, \end{aligned}$$

By construction, the functions f_i belong to $\mathcal{S}^{n\mu,L-\mu+n\mu}$ and are applied to disjoint subsets of the x coordinates. Therefore, when the function is known to have form (7), problem (1) can be written as

$$x^* = \sum_{i=1}^n Q_i x_i^* \qquad x_i^* = \operatorname*{arg\,min}_{x\in\ell_2} f_i(x)\,. \tag{8}$$

Any algorithm that solves optimization problem (1) therefore implicitly solves all the problems listed in (8).

We are almost done, but for one minor detail. Note that we want to obtain a lower bound where the IFO is invoked for a pair (i,x) and responds with $h_i(Q_i^\top x)$ and $\partial h_i(Q_i^\top x)/\partial x$. In order to claim that this suffices to optimize each f_i separately, we need to argue that a first-order oracle for f_i can be obtained from this information, knowing solely the structure of f and not the functions h_i. Since the strong convexity constant μ is assumed to be known to the algorithm, the additional $(n\mu/2)\|x\|^2$ in defining f_i is also known to the algorithm. As a result, given an IFO for f, we can construct a first-order oracle for any of the f_i by simply returning $h_i(Q_i^\top x) + (n\mu/2)\|Q_i^\top x\|^2$ and $\partial h_i(Q_i^\top x)/\partial x + n\mu Q_i Q_i^\top x)$. Furthermore, an IFO invoked with the index i reveals no information about f_j for any other j based on the separable nature of our problem. Hence, the IFO for f offers no additional information beyond having a standard first-order oracle for each f_i.

4.3. Proof of Theorem 1

Based on the discussion above, we can pick any $i \in \{1\ldots n\}$ and view our algorithm as a complicated setup whose sole purpose is to optimize function $f_i \in \mathcal{S}^{n\mu,L-\mu+n\mu}$. Indeed, given the output x_K of an algorithm using an IFO for the function f, we can declare $x_K^i = Q_i^\top x_K$ as our estimate for x_i^*. Lemma 1 then yields

$$\|x_K - x_f^*\|^2 = \sum_{i=1}^n \|Q_i^\top(x_K - x_f^*)\|^2 = \sum_{i=1}^n \|x_K^i - x_i^*\|^2.$$

In order to establish the theorem, we now invoke the classical result on the black-box optimization of functions using a first-order oracle. The specific form of the result stated here is proved in Appendix A.

Theorem 2 (Nemirovsky-Yudin) *Consider a first order black box optimization algorithm for problem (9) that performs* $K \geq 0$ *calls to the oracle and returns an estimate* x_K *of the minimum. For any* $\gamma > 0$*, there exists a function* $f \in \mathcal{S}^{\mu,L}(\ell_2)$ *such that* $\|x_f^*\| = \gamma$ *and*

$$\|x_f^* - x_K\| \geq \gamma\, q^{2K} \quad \text{with } q = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \text{ and } \kappa = \frac{L}{\mu}\,.$$

At a high-level, our oracle will make an independent choice of one of the functions that witness the lower bound in Theorem 2 for each f_i. At a high-level, each function f_i will be chosen to be a quadratic with an appropriate covariance structure such that K_i queries to the function f_i result in the estimation of at most $K_i + 1$ coordinates of x_i^*. By ensuring that the remaining entries still have a substantial norm, a lower bound for such functions is immediate. The precise details on the construction of these functions can be found in Appendix A.[3]

Suppose the IFO is invoked K_i times on each index i, with $K = K_1 + K_2 + \ldots + K_n$. We first establish the theorem for the case $K < n$ in which the algorithm cannot query each functions f_i at least once. After receiving the response x_K, we are still free to arbitrarily choose f_i for any index i that was never queried. No non-trivial accuracy is possible in this case.

Proposition 2 *Consider an IFO algorithm that satisfies the conditions of Theorem 1 with* $K < n$*. Then there is a function* $f \in \mathcal{F}_n^{\mu,L}(\ell_2)$ *such that* $\|x_f^* - x_K\| \geq \gamma$.

Proof Let us execute the algorithm assuming that all the f_i are equal to the function f of Theorem 2 that attains the lower bound with $\gamma = 0$. Since $K < n$, there is at least one function f_j for which $K_j = 0$. Since the IFO has not revealed anything about this function, we can construct function f by redefining function f_j to ensure that

[3] The main difference with the original result of Nemirovsky and Yudin is the dependence on γ instead of $\|x_0 - x_f^*\|$. This is quite convenient in our setting, since it eliminates any possible interaction amongst the starting values of different coordinates for the different functions f_i.

$\|x_K^j - x_j^*\| \geq \|x_j^*\| = \gamma_j$. Since x_j^* is the only part of x^* which is non-zero, we also get $\gamma_j = \gamma$. ■

We can now assume without loss of generality that $K_i > 0$ for each i. Appealing to Theorem 2 for each f_i in turn,

$$\begin{aligned}\|x_K - x_f^*\|^2 &= \sum_{i=1}^n \|x_K^i - x_i^*\|^2 \geq \sum_{i=1}^n \gamma_i^2 q^{4K_i} \\ &= \gamma^2 \sum_{i=1}^n \frac{\gamma_i^2}{\gamma^2} q^{4K_i} \geq \gamma^2 q^{\sum_{i=1}^n \gamma_i^2 4K_i/\gamma^2},\end{aligned}$$

where the last inequality results from Jensen's inequality applied to the convex function $q^{4\alpha}$ for $\alpha \geq 1$. Finally, since the oracle has no way to discriminate amongst the γ_i values when $K_i > 0$, it will end up setting $\gamma_i = \gamma/\sqrt{n}$. With this setting, we now obtain the lower bound

$$\|x_K - x_f^*\|^2 \geq \gamma^2 q^{4K/n},$$

for $K > n$, along with $\|x_K - x_f^*\|^2 \geq \gamma^2$ for $K < n$.

This completes the proof of the Theorem. In order to further establish Corollary 1, we need an additional technical lemma.

Lemma 2 $\forall x > 1$, $\log\left(\frac{\sqrt{x}-1}{\sqrt{x}+1}\right) > \frac{-2}{\sqrt{x-1}}$.

Proof The function $\phi(x) = \log\left(\frac{\sqrt{x}-1}{\sqrt{x}+1}\right) + \frac{2}{\sqrt{x-1}}$ is continuous and decreasing on $(1, +\infty)$ because

$$\begin{aligned}\phi'(x) &= \frac{1}{(\sqrt{x}-1)(\sqrt{x}+1)\sqrt{x}} - \frac{1}{(x-1)\sqrt{x-1}} \\ &= \frac{1}{(x-1)\sqrt{x}} - \frac{1}{(x-1)\sqrt{x-1}} < 0 .\end{aligned}$$

The result follows because $\lim_{x\to\infty} \phi(x) = 0$. ■

Now we observe that we have at least n queries due to the precondition $\varepsilon < 1$ and Proposition 2, which yields the first term in the lower bound. Based on Theorem 1 and this lemma, the corollary is now immediate.

5. Discussion

The results in this paper were motivated by recent results and optimism on exploiting the structure of minimizing finite sums, a problem which routinely arises in machine learning. Our main result provides a lower bound on the limits of gains that might be possible in this setting, allowing us to do a more careful comparison of this setting with regular first-order black box complexity results. As discussed in Section 3, the results seem mixed when the sum consists of n functions based on random data drawn i.i.d. from a distribution. In this statistical setting, we find that the worst-case near-optimal methods like ASDCA can often be much worse than other methods like SAG and SVRG. However, IFO methods like SAG certainly improve upon optimal first-order methods agnostic of the finite sum structure, in ill-conditioned problems. In general, we observe that the problem dependent constants that appear in different methods can be quite different, even though this is not always recognized. We believe that accounting for these opportunities might open door to more interesting algorithms and analysis.

Of course, there is another and a possibly more important aspect of optimization in machine learning which we do not study in this paper. In typical machine learning problems, the goal of optimization is not just to minimize the objective f—usually called the training error—to a numerical precision. In most problems, we eventually want to reason about test error, that is the accuracy of the predictions we make on unseen data. There are existing results (Bottou & Bousquet, 2008) which highlight the optimality of *single-pass stochastic gradient* optimization methods, when test error and not training error is taken into consideration. So far, we do not have any clear results comparing the efficacy of methods designed for the problem (1) in minimizing test error directly. We believe this is an important question for future research, and one that will perhaps be most crucial for the adoption of these methods in machine learning.

We believe that there are some important open questions for future works in this area, which we will conclude with:

1. Is there a fundamental gap between the best IFO methods and the dual coordinate methods in the achievable upper bounds? Or is there room to improve the upper bounds on the existing IFO methods. We certainly found it tricky to do the latter in our own attempts.

2. Is it possible to obtain better complexity upper bounds when the n functions involved in the sum (1) are based on random data, rather than being n arbitrary functions? Can the incremental methods exploit global rather than local smoothness properties in this setting?

3. What are the test error properties of incremental methods for machine learning problems? Specifically, can one do better than just adding up the optimization and generalization errors, and follow a more direct approach as the stochastic optimization literature?

Acknowledgements

We would like to thank Lin Xiao, Sham Kakade and Rong Ge for helpful discussions regarding the complexities of various methods. We also thank the anonymous reviewer who pointed out that the dual coordinate are not valid IFO algorithms.

References

Agarwal, Alekh, Negahban, Sahand, and Wainwright, Martin J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

Arjevani, Y. On Lower and Upper Bounds in Smooth Strongly Convex Optimization - A Unified Approach via Linear Iterative Methods. *ArXiv e-prints*, 2014.

Bertsekas, Dimitri P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. In Sra, S., Nowozin, S., and Wright, S. J. (eds.), *Optimization for Machine Learning*, pp. 85–119. MIT Press, 2012. Extended version: LIDS report LIDS-P2848, MIT, 2010.

Bottou, Léon and Bousquet, Olivier. The tradeoffs of large scale learning. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems*, volume 20, pp. 161–168. NIPS Foundation (http://books.nips.cc), 2008. URL http://leon.bottou.org/papers/bottou-bousquet-2008.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 315–323. 2013.

Le Roux, Nicolas, Schmidt, Mark, and Bach, Francis. A stochastic gradient method with an exponential convergence rate for finite training sets. In Pereira, F., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2663–2671. 2012.

Nemirovsky, Arkadi and Yudin, David B. *Problem Complexity and Method Efficiency in Optimization*. Interscience Series in Discrete Mathematics. Wiley, 1983.

Nesterov, Yurii. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publisher, 2004.

Nesterov, Yurii. Gradient methods for minimizing composite objective function, 2007.

Schmidt, Mark, Le Roux, Nicolas, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 64–72, 2014.

Vershynin, Roman. Introduction to the non-asymptotic analysis of random matrices. In Eldar, Yonina C. and Kutyniok, Gitta (eds.), *Compressed Sensing*, pp. 210–268. Cambridge University Press, 2012.

Zhang, Yuchen and Xiao, Lin. Stochastic primal-dual coordinate method for regularized empirical risk minimization. Technical Report MSR-TR-2014-123, September 2014.