The Min-Max Complexity of Distributed Stochastic Convex Optimization with Intermittent Communication (Extended Abstract)*

Blake Woodworth\textsuperscript{1,2}, Brian Bullins\textsuperscript{2}, Ohad Shamir\textsuperscript{3} and Nathan Srebro\textsuperscript{2}

\textsuperscript{1}Inria
\textsuperscript{2}Toyota Technological Institute at Chicago
\textsuperscript{3}Weizmann Institute of Science

blakewoodworth@gmail.com, bbullins@ttic.edu, ohad.shamir@weizmann.ac.il, nati@ttic.edu

Abstract

We resolve the min-max complexity of distributed stochastic convex optimization (up to a log factor) in the intermittent communication setting, where $M$ machines work in parallel over the course of $R$ rounds of communication to optimize the objective, and during each round of communication, each machine may sequentially compute $K$ stochastic gradient estimates. We present a novel lower bound with a matching upper bound that establishes an optimal algorithm.

1 Introduction

The min-max oracle complexity of stochastic convex optimization in a sequential (non-parallel) setting is very well-understood, and we have provably optimal algorithms that achieve the min-max complexity [Lan, 2012]. However, we do not yet understand the min-max complexity of stochastic optimization in a \textit{distributed} setting, where oracle queries and computation are performed by different workers, with limited communication between them. Perhaps the simplest, most basic, and most important distributed setting is that of \textit{intermittent communication}.

In the (homogeneous) intermittent communication setting, $M$ parallel workers are used to optimize a single objective over the course of $R$ rounds. During each round, each machine sequentially and locally computes $K$ independent unbiased stochastic gradients of the global objective, and then all the machines communicate with each other. This captures the natural setting where multiple parallel “workers” or “machines” are available, and computation on each worker is much faster than communication between workers. It includes applications ranging from optimization using multiple cores or GPUs, to using a cluster of servers, to Federated Learning where workers are edge devices.

The intermittent communication setting has been widely studied for over a decade, with many optimization algorithms proposed and analyzed [Zinkevich et al., 2010; Cotter et al., 2011; Dekel et al., 2012; Zhang et al., 2013b; Shamir and Srebro, 2014], and obtaining new methods and improved analysis is still a very active area of research [Wang et al., 2017; Stich, 2018; Woodworth et al., 2020]. Nevertheless, we do not yet know which methods are optimal, what the min-max complexity is, and what methodological or analytical improvements might allow further progress.

Considerable effort has been made to formalize the setting and establish lower bounds for distributed optimization [Zhang et al., 2013a; Arjevani and Shamir, 2015; Braverman et al., 2016] and here, we follow the graph-oracle formalization of Woodworth et al. [2018]. However, a key issue is that all existing lower bounds for the intermittent communication setting depend only on the product $KR$, i.e. the total number of gradients computed on each machine over the course of optimization, regardless of the number of rounds, $R$, and the number of gradients per round, $K$, separately.

Thus, existing results cannot rule out the possibility that the optimal rate for fixed $T = KR$ can be achieved using only a single round of communication ($R = 1$), since they do not distinguish between methods that communicate very frequently ($R = T$, $K = 1$) and methods that communicate just once ($R = 1$, $K = T$). The possibility that the optimal rate is achievable with $R = 1$ was suggested by Zhang et al. [2013b], and indeed Woodworth et al. [2020] proved that an algorithm that communicates just once is optimal in the special case of quadratic objectives. While it seems unlikely that a single round of communication suffices in the general case, none of our existing lower bounds are able to answer this extremely basic question.

In this paper, we resolve (up to a logarithmic factor) the minimax complexity of smooth, convex stochastic optimization in the (homogeneous) intermittent communication setting. Our main result in Section 3 is a lower bound on the optimal rate of convergence and a matching upper bound. Interestingly, we show that the combination of two extremely simple and naïve methods based on an accelerated stochastic gradient descent (SGD) variant called AC-SA [Lan, 2012] is optimal up to a logarithmic factor. Specifically, we show that the better of the following methods is optimal: “Minibatch Accelerated SGD” which executes $R$ steps of AC-SA using minibatch gradients of size $MK$, and “Single-Machine Accelerated SGD” which executes $KR$ steps of AC-SA on just one of the machines, completely ignoring the other $M - 1$.

*This is an extended abstract of a paper that appeared/won best award at COLT 2021.

\textsuperscript{1}Contact Author
These methods might seem to be horribly inefficient: Minibatch Accelerated SGD only performs one update per round of communication, and Single-Machine Accelerated SGD only uses one of the available workers! This perceived inefficiency has prompted many attempts at developing improved methods which take multiple steps on each machine locally in parallel including, in particular, numerous analyses of Local SGD [Zinkevich et al., 2010; Dekel et al., 2012; Stich, 2018; Woodworth et al., 2020]. Nevertheless, we establish that one or the other is optimal in every regime, so more sophisticated methods cannot yield improved guarantees for arbitrary smooth objectives. Our results therefore highlight an apparent dichotomy between exploiting the available parallelism but not the local computation (Minibatch Accelerated SGD) and exploiting the local computation but not the parallelism (Single-Machine Accelerated SGD).

Our lower bound applies quite broadly, including to the settings considered by much of the existing work on stochastic first-order optimization in the intermittent communication setting. But, like many lower bounds, we should not interpret this to mean we cannot make progress. Rather, it indicates that we need to expand our model or modify our assumptions in order to develop better methods. In Section 5 we explore several additional assumptions that allow for circumventing our lower bound. These include when the third derivative of the objective is bounded (as in recent work by Yuan and Ma [2020]), when the objective has a certain statistical learning-like structure, or when the algorithm has access to a more powerful oracle.

2 Setting and Notation

We aim to understand the fundamental limits of stochastic first-order algorithms in the intermittent communication setting. Accordingly, we consider a standard smooth, convex problem

$$\min_x F(x)$$

where $F$ is convex, has a minimizer with norm $\|x^*\| \leq B$, and $F$ is $H$-smooth, meaning that its gradient $\nabla F$ is $H$-Lipschitz continuous. We consider algorithms that gain information about the objective via a stochastic gradient oracle $g$ with bounded variance1, which satisfies for all $x$

$$\mathbb{E}_g g(x; z) = \nabla F(x) \quad \text{and} \quad \mathbb{E}_g \|g(x; z) - \nabla F(x)\|^2 \leq \sigma^2$$

Characterizing optimal methods for this well-studied class of smooth, convex optimization objectives requires focusing on a specific family of optimization algorithms. We consider intermittent communication algorithms, which attempt to optimize $F$ using $M$ parallel workers, each of which is allowed $K$ queries to $g$ in each of $R$ rounds of communication. Such intermittent communication algorithms can be formalized using the graph oracle framework of Woodworth et al. [2018] which focuses on the dependence structure between different stochastic gradient computations.

1This assumption can be strong, and does not hold for natural problems like least squares regression [Nguyen et al., 2019]. Nevertheless, this strengthens rather than weakens our lower bound.

Finally, we are considering a “homogeneous” setting, where each of the machines have access to stochastic gradients from the same distribution, in contrast to the more challenging “heterogeneous” setting, where they come from different distributions, which could arise in a machine learning context when each machine uses data from a different source. The heterogeneous setting is interesting, important, and widely studied, but we focus here on the more basic question of min-max rates for homogeneous distributed optimization. We point out that our lower bounds also apply to heterogeneous objectives since homogeneous optimization is a special case of heterogeneous optimization, and there are also some lower bounds specific to the heterogeneous setting [Arjevani and Shamir, 2015] but they do not apply here.

3 The Lower Bound

Our main result is a lower bound on what suboptimality can be guaranteed by any (possibly randomized) intermittent communication algorithm in the worst case:

**Theorem 1.** For any $H, B, \sigma, K, R > 0$ and $M \geq 2$, and any intermittent communication algorithm, there exists a convex, $H$-smooth objective with a minimizer of norm $\|x^*\| \leq B$ in any dimension $d = \Omega(KR + H^2B^2\sigma^{-2}M(KR)^{1.25})$ and a stochastic gradient oracle satisfying (2), such that for a numerical constant $C$, the algorithm will have error at least

$$\mathbb{E} F(\hat{x}) - F^* \geq C \min \left\{ \frac{HB^2}{K^2R^2} + \frac{\sigma B}{\sqrt{KBR}} \frac{HB^2}{R^2\log^2 M} + \frac{\sigma B}{\sqrt{MKR}} \right\}$$

**Proof Sketch.** The first and fourth terms of this lower bound follow directly from previous work [Woodworth et al., 2018]. The term $HB^2/(K^2R^2)$ corresponds to optimizing a function with a deterministic gradient oracle, and the term $\sigma B/\sqrt{MKR}$ is a well-known statistical limit [Nemirovskij and Yudin, 1983]. The distinguishing feature of our lower bound is thus the second and third terms, which depend differently on $K$ than $R$. For quadratic objectives, the min-max complexity really does just depend on the product $KR$, and is given by the sum of the first and fourth terms above [Woodworth et al., 2020]. Consequently, in contrast to all of the lower bounds for sequential, smooth, convex optimization that we are aware of, which can be proven using quadratic hard instances, our lower bound proof requires going beyond quadratics, and we use the following hard instance:

$$F(x) = \psi'(-\zeta)x_1 + \psi(x_N) + \sum_{i=1}^{N-1} \psi(x_{i+1} - x_i)$$

where $\psi: \mathbb{R} \to \mathbb{R}$ is defined

$$\psi(x) := \frac{\sqrt{H} x}{2\beta} \arctan \left( \frac{\sqrt{H} \beta x}{2} \right) - \frac{1}{2\beta^2} \log \left( 1 + \frac{H \beta^2 x^2}{4} \right)$$

![Figure 1: The function $\psi(x)$](image)
and where $\beta$, $\zeta$, and $N$ are hyperparameters that are chosen depending on $H, B, \sigma, M, K, R$ so that $F$ satisfies the necessary conditions. This construction closely resembles the classic lower bound for deterministic first-order optimization of Nesterov [2004], which corresponds to $\psi(x) = x^2$. To describe our stochastic gradient oracle, we will use $\text{prog}_n(x) := \max \{ j : |x_j| > \alpha \}$, which denotes the highest index of a coordinate of $x$ that is significantly non-zero. We also define $F^-$ to be equal to the objective with the $\text{prog}_n(x)^{th}$ term removed:

$$F^-(x) = \psi'(-\zeta)x_1 + \psi(x_N) + \sum_{i \notin \text{prog}_n(x)} \psi(x_{i+1} - x_i)$$

The stochastic gradient oracle for $F$ that we use then resembles

$$g(x) = \begin{cases} \nabla F^-(x) & \text{w.p. } 1 - p \\ \nabla F(x) + \frac{1 - p}{p} (\nabla F(x) - \nabla F^-(x)) & \text{w.p. } p \end{cases}$$

This stochastic gradient oracle is similar to the one used by Arjevani et al. [2019] to prove lower bounds for non-convex optimization, and its key property is that $F[\text{prog}_n(g(x))] \leq \text{prog}_n(x) = 1 - p$. Therefore, each oracle access only reveals information about the next coordinate of the gradient the algorithm with probability $p$, and therefore the algorithm is essentially only able to make progress with probability $p$. The rest of the proof revolves around bounding the total progress of the algorithm and showing that $\text{prog}_n(x) \leq N$, then $x$ has high suboptimality.

Since each machine makes $KR$ sequential queries and only makes progress with probability $p$, the total progress scales like $KR \cdot p$. By taking $p$ smaller, we decrease the amount of progress made by the algorithm, and therefore increase the lower bound. Indeed, when $p \approx 1/K$, the algorithm only increases its progress by about $\log M$ per round, which gives rise to the key $(HB^2)/(R^2 \log^2 M)$ term in the lower bound. However, we are constrained in how small we can take $p$ since our stochastic gradient oracle has variance

$$\sup_x \mathbb{E}||g(x) - \nabla F(x)||^2 \approx \frac{1}{p} \sup_x \psi'(x)^2$$

This is where our choice of $\psi$ comes in. Specifically, we chose the function $\psi$ to be convex and smooth so that $F$ is, but we also made it Lipschitz:

$$|\psi'(x)| = \left| \frac{\sqrt{H}}{2\beta} \arctan \left( \frac{\sqrt{H} \beta x}{2} \right) \right| \leq \frac{\pi \sqrt{H}}{4 \beta}$$

Notably, this Lipschitz bound on $\psi$, which implies a bound on $||\nabla F(x)||_\infty$, is the key non-quadric property that allows for our lower bound. Since $\psi'$ is bounded, we are able to choose $p \approx H \sigma^{-2} \beta^{-2}$ without violating the variance constraint on the stochastic gradient oracle. Carefully balancing $\beta$ completes the argument.

Another important aspect of our lower bound is that it applies to arbitrary randomized algorithms, rather than more restricted families of algorithms like “zero-respecting” methods [Carmen et al., 2017]. We therefore prove our Theorem using techniques similar to Woodworth and Srebro [2016], Carmon et al. [2017], Arjevani et al. [2019], and others, who introduce a random rotation matrix, $U$; construct a hard instance like $F(U^T x)$; and argue that any algorithm behaves almost as if it were zero-respecting. For complete details of the proof plus an extension of the lower bound to strongly convex objectives, we refer readers to the full version (see Appendices A-C in [Woodworth et al., 2021]).

### 4 An Optimal Algorithm

The lower bound in Theorem 1 is matched (up to log factors) by the combination of two simple distributed variants of the accelerated SGD algorithm, AC-SA, of Lan [2012]. In the sequential setting, AC-SA algorithm maintains two iterates $y_t$ and $x_t$ which it updates according to

$$y_{t+1} = y_t - \gamma_t \nabla_y \left( \beta_t^{-1} y_t + (1 - \beta_t^{-1}) x_t \right)$$

$$x_{t+1} = \beta_t^{-1} y_{t+1} + (1 - \beta_t^{-1}) x_t$$

for stepsize parameters $\gamma_t$ and $\beta_t$. In the smooth, convex setting, this algorithm converges like [Lan, 2012]

$$\mathbb{E}[F(x_T) - F^*] \leq c \cdot \left( \frac{HB^2}{T^2} + \frac{\sigma B}{\sqrt{T}} \right)$$

The optimal algorithm for the intermittent communication setting combines the following two variants of AC-SA.

The first algorithm, which we refer to as **Minibatch Accelerated SGD**, implements $R$ iterations of AC-SA using minibatch gradients of size $MK$ [Cotter et al., 2011]. Specifically, the method maintains two iterates $y_T$ and $x_T$ which are shared across all the machines. During each round of communication, each machine uses $g$ to compute $K$ independent estimates of $\nabla F(\beta_t^{-1} y_T + (1 - \beta_t^{-1}) x_T)$; the machines then communicate their minibatches, averaging them together into a larger minibatch of size $MK$, and then they update $y_T$ and $x_T$. Minibatching reduces the variance of updates by a factor of $MK$, so (3) implies convergence like

$$\mathbb{E}[F(x_{KR}) - F^*] \leq c \cdot \left( \frac{HB^2}{K^2 R^2} + \frac{\sigma B}{\sqrt{MKR}} \right)$$

The second algorithm, which we call **Single-Machine Accelerated SGD**, “parallelizes” AC-SA differently, specifically by simply ignoring $M - 1$ of the available machines and doing $T = KR$ steps of AC-SA on the remaining one, therefore converging like

$$\mathbb{E}[F(x_{KR}) - F^*] \leq c \cdot \left( \frac{HB^2}{K^2 R^2} + \frac{\sigma B}{\sqrt{KR}} \right)$$

From here, we see that Theorem 1 is equal (up to log factors) to the minimum of (4) and (5), so one of these methods is always optimal:

**Corollary 1.** For any $H, B, \sigma, K, R, M > 0$, the algorithm which returns the output of Minibatch Accelerated SGD when $K \leq \sigma^2 R^3/(H^2 B^2)$ and the output of Single-Machine Accelerated SGD when $K > \sigma^2 R^3/(H^2 B^2)$ is optimal up to a factor of $O(\log^2 M)$.  

5361

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)  
Sister Conferences Best Papers Track
So, in light of Theorem 1 and Corollary 1, we see that intermittent communication algorithms are offered the following dilemma: they may either attain the optimal statistical rate \( \sigma B / \sqrt{MKR} \) but suffer an optimization rate \( HB^2/(K^2 \log^2 M) \) that does not benefit from \( K \) at all, or they may attain the optimal optimization rate of \( HB^2/(K^2 R^2) \) but suffer a statistical rate \( \sigma B / \sqrt{MKR} \) as if only single machine were available. In this sense, there is a very real dichotomy between exploiting parallelism and leveraging local computation.

5 Breaking the Lower Bound

Perhaps the most important use of a lower bound is in understanding how to break it. Instead of viewing the lower bound as telling us to give up any hope of improving over the naive optimal method in Section 4, we should view it as informing us about possible means of making progress.

One way to break our lower bound is by introducing additional assumptions that are not satisfied by the hard instance. These assumptions could then be used to establish when and how some alternate method improves over the “optimal” method in Section 4. Several methods, which operate within the intermittent communication framework of Section 2, have been shown to be better than the “optimal algorithm” in practice for specific instances. However, attempts to demonstrate the benefit of these methods theoretically have so far failed, and we now understand why. In order to understand such benefits, we must introduce additional assumptions, and ask not “is this alternate method better” but rather “under what conditions is this alternate method better?” Below we suggest possible additional assumptions, including ones that have appeared in recent analysis and also other plausible assumptions one could rely on.

Another way to break the lower bound is by considering algorithms that go beyond the stochastic oracle framework of Section 2, utilizing more powerful oracles that nevertheless could be equally easy to implement. Understanding the lower bound can inform us of what type of such extensions might be useful, thus guiding development of novel types of optimization algorithms.

Relying on a Bounded Third Derivative. As we have mentioned, the min-max rate is much better in the special case of quadratic objectives of the form \( Q(x) = \frac{1}{2} x^\top A x + b^\top x \) for p.s.d. \( A \), e.g. least squares problems, in which case Accelerated Local SGD guarantees [Woodworth et al., 2020]:

\[
\mathbb{E} Q(\hat{x}) - Q^* \leq c \cdot \left( \frac{HB^2}{K^2 R^2} + \frac{\sigma B}{\sqrt{MKR}} \right).
\]

Since improvement over the lower bound is possible when the objective is exactly quadratic, it stands to reason that similar improvement should be possible for close to quadratic objectives. Indeed, Yuan and Ma [2020] show that for smooth, convex objectives with \( \alpha \)-Lipschitz Hessian, another accelerated variant of Local SGD guarantees

\[
\mathbb{E} F(\hat{x}) - F^* \leq \tilde{O} \left( \frac{HB^2}{KR^2} + \frac{\sigma B}{\sqrt{MKR}} + \left( \frac{H\sigma^2 B^4}{MKR^3} \right)^{1/3} + \left( \frac{\alpha \sigma^2 B^5}{R^4 K} \right)^{1/3} \right)
\]

This can improve over the lower bound in Theorem 1 in certain parameter regimes when \( \alpha \) is small enough.

Statistical Learning: Assumptions on Components.

Stochastic optimization commonly arises in the context of statistical learning, where the goal is to minimize the expected loss with respect to a model’s parameters. In this case, the objective can be written \( F(x) = E_{z \sim D} f(x; z) \), where \( z \sim D \) represents data drawn i.i.d. from an unknown distribution, and the “components” \( f(x; z) \) represent the loss of the model parametrized by \( x \) on the example \( z \).

For Theorem 1, we only place restrictions on the \( F \) itself, and on the first and second moments of \( q \). But in the statistical learning setting, it is natural to assume that \( q(x) = \nabla f(x; z) \) for an i.i.d. \( z \sim D \) and that \( f(\cdot; z) \) itself has some structure, like smoothness or convexity for each \( z \) individually. This is a non-trivial restriction on the stochastic gradient oracle, and it is very possible that it could be leveraged to design and analyze a method that converges faster than the lower bound in Theorem 1 would allow. In particular, the stochastic gradient oracle used to prove Theorem 1 cannot be written as the gradient of a random smooth function, so it is unclear what would happen in this case.

Higher Order and Other Stronger Oracles. Another avenue for improved algorithms in the intermittent communication setting is to use stronger stochastic oracles. For instance, a stochastic second-order oracle that estimates \( \nabla^2 F(x) \) [Hendrikx et al., 2021] or a stochastic Hessian-vector product oracle that estimates \( \nabla^2 F(x)v \) given a vector \( v \), which can typically be computed as efficiently as stochastic gradients. In the statistical learning setting, some recent work also considers a stochastic prox oracle which returns \( \arg \min_y f(y; z) + \frac{1}{2} \| x - y \|^2 \) [Wang et al., 2017; Chadha et al., 2021].

As an example, stochastic Hessian-vector products, in conjunction with a stochastic gradient oracle can be used to compute approximate Newton updates \( x_{t+1} = x_t - \eta_t \nabla^2 F(x_t)^{-1} \nabla F(x_t) \), which can be rewritten as

\[
x_{t+1} = x_t + \eta_t \arg \min_y \left\{ \frac{1}{2} y^\top \nabla^2 F(x_t) y + \nabla F(x_t)^\top y \right\}
\]

That is, each update can be viewed as the solution to a quadratic optimization problem, and its stochastic gradients can be computed using stochastic Hessian-vector and gradient access to \( F \). The DiSCO algorithm [Zhang and Xiao, 2015] uses distributed preconditioned conjugate gradient descent to find an approximate Newton step. Alternatively, as previously discussed, the quadratic subproblem could be solved to high accuracy in a single round of communication using Accelerated Local SGD, and this approach [Bullins et al., 2021] may converge faster than Theorem 1 would allow for first-order methods.

Acknowledgements

BW was supported in this work by a Google PhD Research Fellowship. This work was also partially supported by NSF-CCF/BSF award 1718970/2016741. NS is also supported by a Google Faculty Research Award.
References


