Random Shuffling Beats SGD after Finite Epochs

Jeff HaoChen [1] **Suvrit Sra** [2]

Abstract

A long-standing problem in optimization is proving that RANDOMSHUFFLE, the without-replacement version of SGD, converges faster than (the usual) with-replacement SGD. Building upon (Gürbüzbalaban et al., 2015b), we present the first non-asymptotic results for this problem, proving that after a reasonable number of epochs RANDOMSHUFFLE converges faster than SGD. Specifically, we prove that for strongly convex, second-order smooth functions, the iterates of RANDOMSHUFFLE converge to the optimal solution as $\mathcal{O}(1/T^2 + n^3/T^3)$, where n is the number of components in the objective, and T is number of iterations. This result implies that after $\mathcal{O}(\sqrt{n})$ epochs, RANDOMSHUFFLE is *strictly better* than SGD (which converges as $\mathcal{O}(1/T)$). The key step toward showing this better dependence on T is the introduction of n into the bound; and as our analysis shows, in general a dependence on n is unavoidable without further changes. To understand how RANDOMSHUFFLE works in practice, we further explore two valuable settings: data sparsity and over-parameterization. For sparse data, RANDOMSHUFFLE has the rate $\mathcal{O}\left(1/T^2\right)$, again strictly better than SGD. Under a setting closely related to over-parameterization, RANDOMSHUFFLE is shown to converge faster than SGD after any *arbitrary* number of iterations. Finally, we extend the analysis of RANDOMSHUFFLE to smooth convex and some non-convex functions.

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University [2]Massachusetts Institute of Technology. Correspondence to: Jeff HaoChen <zhc15@mails.tsinghua.edu.cn>, Suvrit Sra <suvrit@mit.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

1. Introduction

We focus on minimization of the finite-sum

$$F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1.1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is smooth and convex, and the sum F is strongly convex. A classical approach to solving (1.1) is stochastic gradient descent (SGD). At each iteration SGD independently samples an index i uniformly from $\{1, \ldots, n\}$, and uses the (stochastic) gradient ∇f_i to compute its update. The stochasticity makes each iteration of SGD cheap, and the uniformly independent sampling of i makes ∇f_i an unbiased estimator of the full gradient ∇F. These properties are central to SGD's effectiveness in large scale machine learning, and underlie much of its theoretical analysis (see e.g., (Rakhlin et al., 2012; Bertsekas, 2011; Bottou et al., 2016; Shalev-Shwartz and Ben-David, 2014)).

However, what is actually used in practice is the *without replacement* version of SGD, henceforth called RANDOMSHUFFLE. At each epoch RANDOMSHUFFLE samples a random permutation of the n functions uniformly independently (some implementations shuffle the data only once at load, rather than at each epoch). Then, it performs SGD-style updates by going through the n functions according to the sampled permutation. By avoiding random sampling at each iteration, RANDOMSHUFFLE can be computationally more practical (Bottou, 2012); and empirically, it is known to converge faster than SGD (Bottou, 2009).

Resolving this discrepancy between theory and practice of SGD has been long an open problem. Recently, this problem has drawn renewed attention, with the goal of better understanding RANDOMSHUFFLE. The key difficulty is that without-replacement produces non-independent samples, which greatly complicates the analysis. Two extreme case results are known: Shamir (2016) shows that RANDOMSHUFFLE is not much worse than SGD, provided the number of epochs is not too large; while Gürbüzbalaban et al. (2015b) show that RANDOMSHUFFLE converges faster than SGD *asymptotically* at the rate $\mathcal{O}(\frac{1}{T^2})$.

But it remains unclear what happens in between, i.e., after a (reasonable) finite number of epochs are run. This regime is the most compelling one to study, since in practice one runs

neither one, nor infinitely many epochs. This background motivates the central question of our paper:

> *Does* RANDOMSHUFFLE *converge faster than* SGD *after a reasonable number of epochs?*

We answer this question positively in this paper; our results are more precisely summarized below.

1.1. Summary of results

We follow the common practice of reporting convergence rates depending on T, the number of calls to the (stochastic / incremental) gradient oracle. For instance, SGD converges at the rate $\mathcal{O}(\frac{1}{T})$ for solving (1.1), ignoring logarithmic terms in the bound (Rakhlin et al., 2012). Our key observation for RANDOMSHUFFLE is that one should include dependence on n into the bound (see Section 3.3). This compromise then leads to a better dependence on T, which further shows how RANDOMSHUFFLE beats SGD after a finite number of epochs. Our main contributions are the following:

- Under moderate assumptions, we establish a convergence rate of $\mathcal{O}(1/T^2 + n^3/T^3)$ for RANDOMSHUFFLE, where n is the number of components in (1.1), and T the total number of iterations (Thm. 1 and 2). From the bounds we can calculate the number of epochs after which RANDOMSHUFFLE is *strictly better* than SGD.
- We prove that a dependence on n is necessary for beating the SGD rate $\mathcal{O}(1/T)$. This tradeoff precludes the possibility of proving a rate of the type $\mathcal{O}(1/T^{1+\delta})$ with some $\delta > 0$ in the general case, and justifies our choice of introducing n into the rate (Thm. 3).
- Assuming sparse data, a setting common in machine learning, we further improve the convergence rate of RANDOMSHUFFLE to $\mathcal{O}(1/T^2)$. This rate is *strictly better* than SGD, indicating RANDOMSHUFFLE's advantage in such cases (Thm. 4).
- We consider a setting where variance vanishes at the global minimum, which is closely related to the notion of over-parameterization in recent literature. We show that RANDOMSHUFFLE converges faster than SGD after *arbitrary* number of iterations. (Thm. 5)

In Section 4, we discuss various aspects of our results in detail, including explicit comparisons to SGD, the role of condition numbers, as well as limitations. In Section 7, we analyze RANDOMSHUFFLE to a class of non-convex functions and convex functions.

1.2. Related work

Very Recently, Jain et al. (2019) obtained a convergence rate of $\mathcal{O}(n/T^2)$ for RANDOMSHUFFLE under strongly convex setting without assuming Hessian smoothness. Their result improves our result when the number of epochs is smaller than κn, with κ being the condition number.

Recht and Ré (2012) conjectured a remarkable matrix AM-GM inequality that underlies RANDOMSHUFFLE's superiority over SGD. While limited progress on this inequality has been reported (Israel et al., 2016; Zhang, 2014), the full conjecture is wide open. With the technique of transductive Rademacher complexity, Shamir (2016) shows that SGD is not worse than RANDOMSHUFFLE provided the number of iterations is not too large. Ying et al. (2018) show that for a fixed step size, RANDOMSHUFFLE converges to a distribution closer to optimal than SGD asymptotically.

Most closely related to our work, Gürbüzbalaban et al. (2015b) prove that RANDOMSHUFFLE limits to a $\mathcal{O}\big(\frac{1}{T^2}\big)$ rate for large T. Their analysis is based on an epoch level iteration, which "wraps up" the bias brought by without replacement sampling into the error of a single epoch. However, their results are based on an asymptotic version of Chung's Lemma, and are therefore asymptotic. A precise bound on the number of epochs after which their convergence rates apply is not clear. Moreover, the constants hidden in their bound are unclear and can be potentially large. Building upon a similar approach, while introducing new theory to explicitly control the constants, we establish precise non-asymptotic results. A key challenge therein was to reduce the dependence on n, which we overcome by a few steps of carefully constructed AM-GM inequalities.

When the functions in (1.1) are visited in a deterministic order (e.g., cyclic), the method turns into Incremental Gradient Descent (IGD), which has a long history (Bertsekas, 2011). Kohonen (1974) shows that IGD converges to a limit cycle under constant step size and quadratic functions. Convergence to neighborhood of optimality for more general functions is studied in several works, under the assumption that step size is bounded away from zero (see for instance (Solodov, 1998)). With properly diminishing step size, Nedić and Bertsekas (2001) show that an $\mathcal{O}(1/\sqrt{T})$ convergence rate in terms of distance to optimal can be achieved under strong convexity of the finite-sum. This rate is further improved in (Gürbüzbalaban et al., 2015a) to $\mathcal{O}(1/T)$ under a second order differentiability assumption.

In practice, RANDOMSHUFFLE has been proposed as a standard heuristic (Bottou, 2012). With numerical experiments, Bottou (2009) notices an approximately $\mathcal{O}(1/T^2)$ convergence rate of RANDOMSHUFFLE. Without-replacement sampling also improves data-access efficiency in distributed settings (Feng et al., 2012; Lee et al., 2015). The permutation-sampling idea has also been embedded into more complicated algorithms; see (De and Goldstein, 2016; Defazio et al., 2014; Shamir, 2016) for variance-reduced methods, and (Shalev-Shwartz and Zhang, 2013) for decomposition methods.

Table 1. Comparison of convergence rates of SGD and RANDOMSHUFFLE. All functions considered are strongly convex except for the Polyak-Łojasiewicz condition setting. We omit all the constants from the rate (for details on constants, please see Section 4). Under the sparse setting (sparsity level ρ), we are not aware of specialized results corresponding to SGD.

Algorithm	Quadratic	Lipschitz Hessian	Sparse Data	PL Condition
SGD	$\mathcal{O}(1/T)$	$\mathcal{O}(1/T)$	$\mathcal{O}(1/T)$	$\mathcal{O}(1/T)$
RANDOMSHUFFLE	$\mathcal{O}(1/T^2 + n^3/T^3)$	$\mathcal{O}(1/T^2 + n^3/T^3)$	$\mathcal{O}(1/T^2 + \rho^2 n^3/T^3)$	$\mathcal{O}(1/T^2 + n^3/T^3)$

Finally, we note a related but separate body of work on coordinate descent, where a similar problem has been studied: *when does random permutation over coordinates behave well?* Gürbüzbalaban et al. (2017) give two kinds of quadratic problems where cyclic coordinate descent beats the with-replacement randomized one, which is a stronger result indicating that random permutation also beats the with-replacement method. However, such a deterministic version of the algorithm suffers from poor worst case. Indeed, in (Sun and Ye, 2016) a setting is analyzed where cyclic coordinate descent can be much worse than both with-replacement and random permutation versions. Lee and Wright (2016) further analyze how the random permutation version of coordinate descent avoids the slow convergence of cyclic version. Wright and Lee (2017) propose a more general class of quadratic functions where random permutation outperforms cyclic coordinate descent.

2. Background and problem setup

For problem (1.1), we assume the finite sum function $F(x) : \mathbb{R}^d \to \mathbb{R}$ is strongly convex, i.e.,

$$F(x) \geq F(y) + \langle \nabla F(y), x - y \rangle + \tfrac{\mu}{2} \|x - y\|^2 ,$$

where $x, y \in \mathbb{R}^d$, and $\mu > 0$ is the strong convexity parameter. Furthermore, we assume each component function is L-smooth, so that for $i = 1, \ldots, n$, there exists a constant L such that

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L \|x - y\| . \tag{2.1}$$

Furthermore, we assume that the component functions are second order differentiable with a Lipschitz continuous Hessian. We use $H_i(x)$ to denote the Hessian of function f_i at x. Specifically, for each $i = 1, \ldots, n$, we assume that for all $x, y \in \mathbb{R}^d$, there exists a constant L_H such that

$$\|H_i(x) - H_i(y)\| \leq L_H \|x - y\| . \tag{2.2}$$

The norm is the spectral norm for matrices and ℓ_2 norm for vectors. We denote the unique minimizer of $F(x)$ as x^*, the index set $\{1, \cdots, n\}$ as $[n]$. The complexity bound is represented as $\mathcal{O}(\cdot)$, with all logarithmic terms hidden. All other parameters that might be hidden in the complexity bounds will be clarified in corresponding sections.

2.1. The algorithms under study: SGD and RANDOMSHUFFLE

For both SGD and RANDOMSHUFFLE, we use γ as the step size, which is predetermined before the algorithms are run. The sequences generated by both methods are denoted as $(x_k)_{k=0}^T$; here x_0 is the initial point and T is the total number of iterations (i.e., number of stochastic gradients used).

SGD runs as follows: at each iteration $1 \leq k \leq T$, it picks an index $s(k)$ independently uniformly from the index set $[n]$, and then performs the update

$$x_k = x_{k-1} - \gamma \nabla f_{s(k)}(x_{k-1}). \tag{SGD}$$

In contrast, RANDOMSHUFFLE runs as follows: for each epoch t, it picks one permutation $\sigma_t(\cdot) : [n] \to [n]$ independently uniformly from the set of all permutations of $[n]$. Then, it sequentially visits each of the component functions of the finite-sum (1.1) and performs the update

$$x_k^t = x_{k-1}^t - \gamma \nabla f_{\sigma_t(k)} \left(x_{k-1}^t \right), \quad \text{(RANDOMSHUFFLE)}$$

for $1 \leq k \leq n$. Here $x_k^t = x_{(t-1)n+k}$ represents the k-th iterate within the t-th epoch. For two consecutive epochs t and $t+1$, one has $x_0^{t+1} = x_n^t$; for the initial point one has $x_0^1 = x_0$. For convenience of analysis, we always assume RANDOMSHUFFLE is run for an integer number of epochs, i.e., $T = ln$ for some $l \in \mathbb{Z}^+$. This is a reasonable assumption given our main interest is when several epochs of RANDOMSHUFFLE are run.

3. Convergence analysis

The goal of this section is to build theoretical analysis for RANDOMSHUFFLE. Specifically, we answer the following question: *when can we show* RANDOMSHUFFLE *to be better than* SGD? We begin by first analyzing quadratic functions in Section 3.1, where the analysis benefits from having a constant Hessian. Subsequently, in Section 3.2, we extend our analysis to the general (smooth) strongly convex setting. A key idea in our analysis is to make the convergence rate bounds sensitive to n, the number of components in the finite-sum (1.1). In Section 3.3, we discuss and justify the necessity of introducing n into our convergence bound.

3.1. RANDOMSHUFFLE for quadratics

We first consider the quadratic instance of (1.1), where

$$f_i(x) = \tfrac{1}{2}x^T A_i x + b_i^T x, \qquad i = 1, \dots, n, \tag{3.1}$$

where $A_i \in \mathbb{R}^{d\times d}$ is positive semi-definite, and $b_i \in \mathbb{R}^d$. We should notice often in analyzing strongly convex problems, the quadratic case presents a good example when tight bounds are achieved.

Quadratic functions have a constant Hessian function $H_i(x) = A_i$, which eases our analysis. In particular, our bound depends on the following constants: (i) strong convexity parameter μ, and component-wise Lipschitz constant L; (ii) diameter bound $\|x - x^*\| \le D$ (i.e., any iterate x remains bounded; can be enforced by explicit projection if needed); and (iii) bounded gradients $\|\nabla f_i(x)\| \le G$ for each f_i ($1 \le i \le n$), and any x satisfying (ii). We omit these constants for clarity, but discuss the condition number further in Section 4.

Our main result for RANDOMSHUFFLE is the following theorem (omitting logarithmic terms):

Theorem 1. *With f_i defined by (3.1), let the condition number of problem (1.1) be $\kappa = L/\mu$. So long as $\frac{T}{\log T} > 6(1+\kappa)n$, with step size $\gamma = \frac{4\log T}{T\mu}$,* RANDOMSHUFFLE *achieves convergence rate:*

$$\mathbb{E}[\|x_T - x^*\|^2] \le \mathcal{O}\Big(\frac{1}{T^2} + \frac{n^3}{T^3}\Big).$$

We prove this theorem based on the same idea as (Gürbüzbalaban et al., 2015b), i.e., by establishing an epoch-based recursion inequality. However, we no longer think of n as a constant, but rather state it *explicitly* in the bound and try to optimize the dependence on it. The main challenge comes with controlling the dependence on n, which is reduced to what appears in Theorem 1 as several steps of a carefully designed AM-GM inequality. (See equation (A.8) and (A.9) in supplementary material.)

We provide a proof sketch for Theorem 1 in Section 8, deferring the involved technical details to the appendix.

In terms of sample complexity, Theorem 1 implies the following corollary:

Corollary 1. *With f_i defined by (3.1), the sample complexity for* RANDOMSHUFFLE *to achieve $\mathbb{E}[\|x_T - x^*\|^2]$ is no more than $\mathcal{O}\big(\epsilon^{-\frac{1}{2}} + n\epsilon^{-\frac{1}{3}}\big)$.*

3.2. RANDOMSHUFFLE for strongly convex problems

Next, we consider the more general case where each component function f_i is convex and the sum $F(x) = \frac{1}{n}\sum_i f_i(x)$ is strongly convex. Surprisingly[1], one can easily adapt the methodology of the proof for Theorem 1 in this setting. To this end, our analysis requires one further assumption that each component function is twice differentiable, and its Hessian satisfies the Lipschitz condition (2.2) with constant L_H. Under these assumptions, we obtain the following result:

Theorem 2. *Define constant*

$$C = \max\left\{\frac{32}{\mu^2}(L_H L D + 3L_H G), 12(1+\frac{L}{\mu})\right\}.$$

So long as $\frac{T}{\log T} > Cn$, with step size $\eta = \frac{8\log T}{T\mu}$, RANDOMSHUFFLE *achieves convergence rate:*

$$\mathbb{E}[\|x_T - x^*\|^2] \le \mathcal{O}\Big(\frac{1}{T^2} + \frac{n^3}{T^3}\Big).$$

Except for extra dependence on L_H and a mildly different step size, this rate is essentially the same as that in quadratic case. The proof for the result can be found in the supplement. Due to the similar formulation, most of the consequences noted in Section 3.1 also hold in this general setting.

3.3. Understanding the dependence on n

Since the motivation of building our convergence rate analysis is to show that RANDOMSHUFFLE behaves better than SGD, we would definitely hope that our convergence bounds have a better dependence on T compared to the $\mathcal{O}(\frac{1}{T})$ bound for SGD. In an ideal situation, one may hope for a rate of the form $\mathcal{O}(\frac{1}{T^{1+\delta}})$ with some $\delta > 0$. One intuitive criticism toward this goal is evident: if we allow $T < n$, then by setting $n > T^2$, RANDOMSHUFFLE is essentially same as SGD by the birthday paradox. Therefore, a $\mathcal{O}\left(\frac{1}{T^{1+\delta}}\right)$ bound is unlikely to hold.

However, this argument is not rigorous when we require a positive number of epochs to be run (at least one round through all the data). To this end, we provide the following result indicating the impossibility of obtaining $\mathcal{O}(\frac{1}{T^{1+\delta}})$ even when $T \ge n$ is required.

Theorem 3. *Given the information of μ, L, G. Under the assumption of constant step sizes, no step size choice for* RANDOMSHUFFLE *leads to a convergence rate $o\left(\frac{1}{T}\right)$ for any $T \ge n$, if we do not allow n to appear in the bound.*

The key idea to prove Theorem 3 is by constructing a special instance of problem (1.1). In particular, the following quadratic instance of (1.1) lays the foundation of our proof:[2]

$$f_i(x) = \begin{cases} \frac{1}{2}(x-b)'A(x-b) & i \text{ odd}, \\ \frac{1}{2}(x+b)'A(x+b) & i \text{ even}. \end{cases} \tag{3.2}$$

[1] Intuitively, the change of Hessian over the domain can raise challenges. However, our convergence rate here is quite similar to quadratic case, with only mild dependence on Hessian Lipschitz constant.

[2] The same setting is also used for the comparison of limiting cycles of randomized/deterministic incremental gradient descent methods, see for instance Example 1.5.6 of (Bertsekas, 1999).

Here $(\cdot)'$ denotes the transpose of a vector, $A \in \mathbb{R}^{d\times d}$ is some positive definite matrix, and $b \in \mathbb{R}^d$ is some vector. Running RANDOMSHUFFLE on (3.2) leads to a closed-formed expression of RANDOMSHUFFLE's error. Then by setting $T = n$ (i.e., only running RANDOMSHUFFLE for one epoch) and assuming a convergence rate of $o\left(\frac{1}{T}\right)$, we deduce a contradiction by properly setting A and b. The detailed proof can be found in our supplementary document. We directly have the following corollary:

Corollary 2. *Given the information of μ, L, G, under the assumption $T \geq n$ and constant step size, there is no step size choice that leads to a convergence rate $\mathcal{O}\left(\frac{1}{T^{1+\delta}}\right)$ for any $\delta > 0$.*

This result indicates that in order to achieve a better dependence on T using constant step sizes, the bound should either: (i) depend on n; (ii) make some stronger assumptions on T being large enough (at least exclude $T = n$); or (iii) leverage a more versatile step size schedule, which could potentially be hard to design and analyze.

Although Theorem 3 shows that one may not hope (assuming constant step sizes) for a better dependence on T for RANDOMSHUFFLE without an extra n dependence, whether the current dependence on n is optimal still requires further discussion. In the special case $n = T$, numerical evidence has shown that RANDOMSHUFFLE behaves at least as well as SGD. However, our bound fails to even show RANDOMSHUFFLE converges in this setting. Therefore, it is reasonable to conjecture that a better dependence on n exists. In the following section, we improve the dependence on n under a specific setting. But whether a better dependence on n can be achieved in general remains open.

4. Discussion of results

We discuss below our results in more detail, including their implications, strengths, and limitations.

Comparison with SGD: It is well-known that under strong convexity SGD converges with a rate of $\mathcal{O}\left(\frac{1}{T}\right)$ (Rakhlin et al., 2012). A direct comparison indicates the following fact: RANDOMSHUFFLE is provably better than SGD after $\mathcal{O}(\sqrt{n})$ epochs. This is an acceptable amount of epochs for even some of the largest data sets in current machine learning literature. To our knowledge, this is the *first* result rigorously showing that RANDOMSHUFFLE behaves better than SGD within a reasonable number of epochs. To some extent, this result confirms the belief and observation that RANDOMSHUFFLE is the "correct" choice in real life, at least when the number of epochs is comparable with $\sqrt{n}$.

Deterministic variant: When the algorithm is run in a deterministic fashion, i.e., the functions f_i are visited in a fixed order, better convergence rate than SGD can also be achieved as T becomes large. For instance, a result in (Gürbüzbalaban et al., 2015a) translates into a $\mathcal{O}\left(\frac{n^2}{T^2}\right)$ bound for the deterministic case. This directly implies the same bound for RANDOMSHUFFLE, since random permutation always has the weaker worst case. But according to this bound, at least n epochs are required for RANDOMSHUFFLE to achieve an error smaller than SGD, which is not a realistic number of epochs in most applications.

Comparison with GD: Another interesting viewpoint is by comparing RANDOMSHUFFLE with Gradient Descent (GD). One of the limitations of our result is that we do not show a regime where RANDOMSHUFFLE can be better than GD. By computing the average for each epoch and running exact GD on (1.1), one can get a convergence rate of the form $\mathcal{O}(e^{-\frac{T}{n}})$. This fact shows that our convergence rate for RANDOMSHUFFLE is worse than GD. This comes naturally from the epoch based recursion (8.1) in our proof methodology, since for one epoch the sum of the gradients is only shown to be no worse than a full gradient. It is true that GD should behave better in long-term as the dependence on n is negligible, and comparing with GD is not the major goal for this paper. However, being worse than GD even when T is relatively small indicates that the dependence on n probably can still be improved. It may be worth investigating whether RANDOMSHUFFLE can be better than both SGD and GD in some regime. However, different techniques may be required.

Epochs required: It is also a limitation that our bound only holds after a certain number of epochs. Moreover, this number of epochs is dependent on κ (e.g., $\mathcal{O}(\kappa)$ epochs for the quadratic case). This limits the interest of our result to cases when the problem is not too ill-conditioned. Otherwise, such a number of epochs will be unrealistic by itself. We are currently not certain whether similar bounds can be proved when allowing T to assume smaller values, or even after only one epoch.

Dependence on κ: It should be noticed that κ can be large sometimes. Therefore, it may be informative to view our result in a κ-dependent form. In particular, we still assume D, L, L_H are constant, but no longer μ. We use the bound $G \leq \max_i \|\nabla f_i(x^*)\| + DL$ and assume $\max_i \|\nabla f_i(x^*)\|$ is constant. Since $\kappa = \frac{L}{\mu}$, we now have $\kappa = \Theta(\frac{1}{\mu})$. Our results translate into κ-dependent sample complexity $\mathcal{O}(\kappa n + \kappa^2\epsilon^{-\frac{1}{2}} + n\kappa^{\frac{4}{3}}\epsilon^{-\frac{1}{3}} + n\kappa^{\frac{3}{2}}\epsilon^{-\frac{1}{4}})$ for quadratic problems, and $\mathcal{O}(\kappa^2 n + \kappa^2\epsilon^{-\frac{1}{2}} + n\kappa^{\frac{4}{3}}\epsilon^{-\frac{1}{3}} + n\kappa^{\frac{3}{2}}\epsilon^{-\frac{1}{4}})$ for strongly convex ones.

At first sight, the dependence on κ in the convergence rate may seem relatively high. However, it is important to notice that our sample complexity's dependence on κ is actually *better* than what is known for SGD. A $\mathcal{O}(\frac{4G^2}{T\mu^2})$ convergence bound for SGD has long been known (Rakhlin et al., 2012), which translates into a $\mathcal{O}(\frac{\kappa^2}{\epsilon})$, κ-dependent sample complexity in our notation. Although better κ

dependence has been shown for $F(x_T) - F(x^*) < \epsilon$ (see e.g., (Hazan and Kale, 2014)), the κ^2 dependence is still the best for $\mathbb{E}[\|x_T - x^*\|^2] < \epsilon$ as far as we know (e.g., Nguyen et al. (2018)). Furthermore, according to (Nemirovskii et al., 1983), the lower bound to achieve $F(x_T) - F(x^*) < \epsilon$ for strongly convex F using stochastic gradients is $\Omega(\frac{\kappa}{\epsilon})$. Translating this into the sample complexity to achieve $\mathbb{E}[\|x_T - x^*\|^2] < \epsilon$ is likely to introduce another κ into the bound. Therefore, it is reasonable to believe that $\mathcal{O}(\frac{\kappa^2}{\epsilon})$ is the best sample complexity one can get for SGD (which is worse than RANDOMSHUFFLE), to achieve $\mathbb{E}[\|x_T - x^*\|^2] < \epsilon$.

Assumptions on bounded iterates / gradients: Although the bounded iterates / gradients assumptions may appear to be strong, they are actually quite reasonable: unlike SGD, RANDOMSHUFFLE with proper step size decreases the loss function after one epoch no matter what the random permutation is. The bounded iterates / gradients assumptions can be therefore guaranteed by showing that RANDOMSHUFFLE is indeed a "descent" algorithm (after ensuring that the initial sublevel set is bounded).

5. Sparse functions

In the literature on large-scale machine learning, sparsity is a common feature of data. When the data are sparse, each training data point has only a few non-zero features. Under such a setting, each iteration of SGD only modifies a few dimensions of the decision variables. Some commonly occurring sparse problems include large-scale logistic regression, matrix completion, and graph cuts.

Sparse data provides a prospective setting under which RANDOMSHUFFLE might be powerful. Intuitively, when data are sparse, with-replacement sampling used by SGD is likely to miss some decision variables, while RANDOMSHUFFLE is guaranteed to update all possible decision variables in one epoch. In this section, we show some theoretical results justifying such intuition.

Formally, a sparse finite-sum problem assumes the form

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_{e_i}),$$

where e_i $(1 \le i \le n)$ denotes a small subset of $\{1, \dots, d\}$ and x_{e_i} denotes the entries of the vector x indexed by e_i. Define the set $E := \{e_i : 1 \le i \le n\}$. By representing each subset $e_i \subseteq E$ with a node, and considering edges (e_i, e_j) for all $e_i \cap e_j \neq \emptyset$, we get a graph with n nodes. Following the notation in (Recht et al., 2011), we consider the *sparsity factor* of the graph:

$$\rho := \frac{\max_{1\le i\le n} |\{e_j \in E : e_i \cap e_j \neq \emptyset\}|}{n}. \tag{5.1}$$

One obvious fact is $\frac{1}{n} \le \rho \le 1$. The statistic (5.1) indicates how likely is it that two subsets of indices intersect, which reflects the sparsity of the problem. For a problem with strong sparsity, we may anticipate a relatively small value for ρ. We summarize our result with the following theorem:

Theorem 4. *Define constant*

$$C = \max\left\{\frac{32}{\mu^2}(L_H LD + 3L_H G), 12(1 + \frac{L}{\mu})\right\}.$$

So long as $\frac{T}{\log T} > Cn$, *with step size* $\eta = \frac{8\log T}{T\mu}$, RANDOMSHUFFLE *achieves convergence rate:*

$$\mathbb{E}[\|x_T - x^*\|^2] \le \mathcal{O}\Big(\frac{1}{T^2} + \frac{\rho^2 n^3}{T^3}\Big).$$

Compared with Theorem 2, the bound in Theorem 4 depends on the parameter ρ, so we can exploit sparsity to obtain a faster convergence rate. The key to proving Theorem 4 lies in constructing a tighter bound for the error term in the main recursion (see §8) by including a discount due to sparsity.

Corollary 3. *When* $\rho = \mathcal{O}\left(\frac{1}{n}\right)$ *and constant* C *defined as in Theorem 4, so long as* $\frac{T}{\log T} > Cn$, *for step size* $\eta = \frac{8\log T}{T\mu}$, RANDOMSHUFFLE *achieves convergence rate*

$$\mathbb{E}[\|x_T - x^*\|^2] \le \mathcal{O}\Big(\frac{1}{T^2}\Big).$$

As shown in the above corollary, with sparsity factor $\rho = \mathcal{O}\left(\frac{1}{n}\right)$, the proven convergence rate of RANDOMSHUFFLE is strictly better than the $\mathcal{O}\left(\frac{1}{T}\right)$ rate of SGD. This result follows the following intuition: when each dimension is only touched by several functions, letting the algorithm to visit every function would avoid missing certain dimensions. For larger ρ, similar speedup can be observed. In fact, so long as we have $\rho = o(n^{-\frac{1}{2}})$, the proven bound is better off than SGD. Such a result confirms the usage of RANDOMSHUFFLE under sparse setting.

6. An example where RANDOMSHUFFLE always converges faster

In this section, we study a specialized class of convex problems where RANDOMSHUFFLE always converges faster than SGD, i.e., after any *arbitrary* number of iterations.

We build our example with the vanishing variance property: $\nabla f_i(x^*) = 0$ for the optimal point x^*. Moulines and Bach (2011) show that when $F(x)$ is strongly convex, SGD converges linearly in this setting. Notably, this setting is closely related to the notion of over-parameterization in recent machine learning methods such as deep neural networks, as this relationship has been addressed in (Ma et al., 2017). One special case of this setting is when the variance of stochastic gradients can be bounded by either norm of full gradient

or the suboptimality of the loss. Many previous works are built on these assumptions, such as (Tseng, 1998; Solodov, 1998; Schmidt and Roux, 2013; Vaswani et al., 2018).

We consider below a nontrivial subclass of strongly convex problems coined *valid problems*. Given the constraint defined by the strong convexity and Lipschitz-continuity constants, we show that RANDOMSHUFFLE achieves better *worst case* convergence rate among all valid problems.

Given n pairs of positive numbers $(\mu_1, L_1), \cdots, (\mu_n, L_n)$ such that $0 \leq \mu_i \leq L_i$, a dimension d and a point $x^* \in \mathbb{R}^d$, we say a d dimensional finite-sum function $F(x) = \sum_{i=1}^n f_i(x)$ is a *valid problem* if: each component $f_i(x)$ is μ_i-strongly convex, the gradient of component $f_i(x)$ is L_i-Lipschitz continuous, and there is some x^* minimizing all components at the same time (which is equivalent to vanishing componentwise gradient). A set of valid problems $\mathcal{P}$ is characterized by n, $(\mu_1, L_1), \cdots, (\mu_n, L_n)$, d, and an upper bound on initial distance: $R \geq \|x_0 - x^*\|$.

For a problem $P \in \mathcal{P}$, let random variable X_{RS} be the result of running RANDOMSHUFFLE from initial point x_0 for T iterations with step size γ on problem P. Similarly, let X_{SGD} be the result of running SGD from initial point x_0 for T iterations with step size γ on problem P.

For a fixed set $\mathcal{P}$, we can compare the worst case convergence of RANDOMSHUFFLE and SGD within this set:

Theorem 5. *Given constants $(\mu_1, L_1), \cdots, (\mu_n, L_n)$ such that $0 \leq \mu_i \leq L_i$, a dimension d, a point $x^* \in \mathbb{R}^d$ and an upper bound of initial distance $\|x_0 - x^*\|_2 \leq R$. Let $\mathcal{P}$ be the set of valid problems. For step size $\gamma \leq \min_i\{\frac{2}{L_i+\mu_i}\}$ and any $T \geq 1$, there is*

$$\max_{P\in\mathcal{P}} \mathbb{E}\left[\|X_{RS} - x^*\|^2\right] \leq \max_{P\in\mathcal{P}} \mathbb{E}\left[\|X_{SGD} - x^*\|^2\right].$$

This theorem indicates that RANDOMSHUFFLE has a better worst-case convergence rate than SGD after an *arbitrary* number of iterations under this noted setting. [3]

7. Extensions

In this section, we provide some further extensions.

7.1. RANDOMSHUFFLE for nonconvex optimization

The first extension that we discuss is to nonconvex finite sum problems. In particular, we study RANDOMSHUFFLE applied to functions satisfying the *Polyak-Łojasiewicz* (PL) condition (also known as gradient dominated functions):

$$\tfrac{1}{2}\|\nabla F(x)\|^2 \geq \mu(F(x) - F^*), \quad \forall x.$$

Here $\mu > 0$ is some real number, F^* is the minimal function value of $F(\cdot)$. Strongly convexity is a special situation of this condition with μ being the strongly convex parameter. One important implication of this condition is that every stationary point is a global minimum. However function F can be non-convex under such setting. Also, it doesn't imply a unique minimum of the function.

This setting was proposed and analyzed in (Polyak, 1963), where a linear convergence rate for GD was shown. Later, many other optimization methods have been proven efficient under this condition (see (Nesterov and Polyak, 2006) for second order methods and (Reddi et al., 2016) for variance reduced gradient methods). Notably, SGD converges at the rate $\mathcal{O}(1/T)$ under this setting (see appendix for a proof).

Assume each component function f_i being L Lipschitz continuous, and the average function $F(x)$ satisfying the Polyak-Łojasiewicz condition with some constant μ. We have the following extension of our previous result:

Theorem 6. *Under the Polyak-Łojasiewicz condition, define condition number $\kappa = L/\mu$. So long as $\frac{T}{\log T} > 16\kappa^2 n$, with step size $\eta = \frac{2\log T}{T\mu}$, RANDOMSHUFFLE achieves convergence rate:*

$$\mathbb{E}[\|x_T - x^*\|^2] \leq \mathcal{O}\Big(\tfrac{1}{T^2} + \tfrac{n^3}{T^3}\Big).$$

7.2. RANDOMSHUFFLE for convex problems

An important extension of RANDOMSHUFFLE is to the general (smooth) convex case without assuming strong convexity. There are no previous results on the convergence rate of RANDOMSHUFFLE in this setting that show it to be faster than SGD. The only result we are aware of is by Shamir (2016), who shows RANDOMSHUFFLE is not worse than SGD in the (smooth) convex setting. We extend our results to the general convex case, and show a convergence rate that is possibly faster than SGD in certain parameter regimes, albeit only up to constant terms.

We take the viewpoint of gradients with errors, and denote the difference between component gradient and full gradient as the error:

$$\nabla F(x) - \nabla f_i(x) = e_i(x).$$

Different assumptions bounding the error term $e_i(x)$ have been studied in optimization literature. We assume that there is a constant δ that bound the norm of the gradient error:

$$\|e_i(x)\| \leq \delta, \quad \forall x.$$

[3] The inequality is true even if every component is only convex but not necessarily strongly convex, i.e., $\mu_i = 0$ for all i. However, there is no meaning of comparing worst case convergence in terms of distance to a specific x^* in this case, since there can potentially exist more than one global minimizer.

Here i is any index and x is any point in domain. Obviously, $\delta \leq 2G$, with G being the gradient norm bound as before.[4]

Theorem 7. *Assume $\Delta = \mathbb{E}_{i\neq j} H_i(x^*)\nabla f_j(x^*)$ with i, j uniformly drawn from $[n]$, x^* is an arbitrary minimizer of F. Assume $\bar{x} = \frac{n}{T}\sum_{i=1}^{T/n} x_0^i$ being the average of epoch ending points of* RANDOMSHUFFLE*. Then with proper step size, we have the bound*

$$F(\bar{x}) - F(x^*) \leq \frac{2D\sqrt{nD\left(\|\Delta\| + L_H LD^2 + 2L_H DG\right)}}{\sqrt{T}} + \mathcal{O}\left(\left(\tfrac{n}{T}\right)^{2/3}\delta^{1/3} + \left(\tfrac{n}{T}\right)^{3/4}\right).$$

Compared to the known convergence rate of $\mathcal{O}(DG/\sqrt{T})$ for SGD, we can see that RANDOMSHUFFLE and SGD share the same asymptotic rate of $\mathcal{O}(1/\sqrt{T})$. However, in certain parameter space, the constant in front of $1/\sqrt{T}$ for RANDOMSHUFFLE can be smaller than that of SGD.

8. Proof sketch of Theorem 1

In this section we provide a proof sketch for Theorem 1. The central idea is to establish an inequality

$$\mathbb{E}\left[\left\|x_0^{t+1} - x^*\right\|^2\right] \leq (1-n\gamma\alpha_1)\left\|x_0^t - x^*\right\|^2 + n\gamma^3\alpha_2 + n^4\gamma^4\alpha_3, \tag{8.1}$$

where x_0^t and x_0^{t+1} are the beginning and final points of the t-th epoch, respectively. Constant α_1 captures the speed of convergence for the linear convergence part, while α_2 and α_3 together bound the error introduced by randomness.

We start from the following equality for one epoch of RANDOMSHUFFLE:

$$\left\|x_0^{t+1} - x^*\right\|^2 = \left\|x_0^t - x^*\right\|^2 \underbrace{- 2\gamma\langle x_0^t - x^*, n\nabla F(x_0^t)\rangle}_{A_1^t} \underbrace{- 2\gamma\langle x_0^t - x^*, R^t\rangle}_{A_2^t} + \underbrace{2\gamma^2\left\|n\nabla F(x_0^t)\right\|^2}_{A_3^t} + \underbrace{2\gamma^2\left\|R^t\right\|^2}_{A_4^t},$$

where random variable

$$R^t = \sum\nolimits_{i=1}^n \nabla f_{\sigma_t(i)}\left(x_{i-1}^t\right) - \sum\nolimits_{i=1}^n \nabla f_{\sigma_t(i)}(x_0^t)$$

denotes the gradient error of RANDOMSHUFFLE for epoch t dependent on functions permutation $\sigma_t(\cdot)$. The right hand side of the equality can be thought as two parts: terms that behave like full gradient descent (A_1^t and A_3^t) and terms that capture the effects of random sampling (A_2^t and A_4^t). The main body of our analysis involves bounding each of these terms separately.

[4] Another common assumption is when the variance of the gradient (i.e., $\mathbb{E}[\|e_i(x)\|^2]$) is bounded. We made the more rigorous assumption here for ease of a simpler analysis. However, there is at most an extra $\sqrt{n}$ term difference between these two assumptions due to the finite sum structure.

A key challenge toward building (8.1) is to bound $\mathbb{E}[A_2^t]$, where the expectation is over $\sigma_t(\cdot)$. It is not easy to directly bound this term with $\gamma^3 C$ for some constant C. Instead, by introducing second-order information and several steps of carefully designed AM-GM inequality, we obtain the following bound for A_2^t:

Lemma 1. *Over the randomness of the permutation, we have the inequality:*

$$\mathbb{E}[A_2^t] \leq \frac{1}{2}\gamma\mu(n-1)\left\|x_0^t - x^*\right\|^2 + \gamma^2 n^2\left\|\nabla F\left(x_0^t\right)\right\|^2 + \gamma^3\mu^{-1}n^2(n-1)\|\Delta\|^2 + 2\mu^{-1}\gamma^5 L^4 G^2 n^5.$$

where $\Delta = \mathbb{E}_{i\neq j} H_i(x^*)\nabla f_j(x^*)$ with i, j uniformly drawn from $[n]$, and x^* is the minimizer of sum function. Furthermore, we have $\|\Delta\| \leq \frac{1}{n-1}LG$.

We bound A_1^t with standard inequality (Nesterov, 2013)

$$A_1^t \geq \frac{2n\gamma}{L+\mu}\left\|\nabla F(x_0^t)\right\|^2 + 2n\gamma\frac{L\mu}{L+\mu}\left\|x_0^t - x^*\right\|^2, \tag{8.2}$$

where the first term is further used to bound A_3^t and A_2^t, and the second term leads to the optimization gain $\alpha_1 > 0$. We absorb A_4^t into α_3 term in (8.1), which finishes the build-up of the recursion.

Finally, expanding the recursion (8.1) and substituting a proper step-size leads to a bound of the form $\mathcal{O}\left(\frac{1}{T^2} + \frac{n^3}{T^3}\right)$. The complete proof can be found in the supplement.

9. Conclusions

A long-standing problem in the theory of stochastic gradient descent (SGD) is to prove that RANDOMSHUFFLE converges faster than the usual with-replacement SGD. In this paper, we provide the first non-asymptotic convergence rate analysis for RANDOMSHUFFLE. We show in particular that after $\mathcal{O}(\sqrt{n})$ epochs, RANDOMSHUFFLE behaves strictly better than SGD under strong convexity and second-order differentiability. The underlying introduction of dependence on n into the bound plays an important role toward a better dependence on T. We further improve the dependence on n for sparse data settings, showing RANDOMSHUFFLE's advantage in such situations.

Acknowledgments

SS acknowledges support from the NSF-CAREER award (id 1846088).

References

D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

D. P. Bertsekas. Incremental gradient, subgradient, and

proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

L. Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. In *Proceedings of the symposium on learning and data science, Paris*, 2009.

L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *arXiv:1606.04838*, 2016.

S. De and T. Goldstein. Efficient distributed sgd with variance reduction. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 111–120, 2016.

A. Defazio, J. Domke, et al. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning*, pages 1125–1133, 2014.

X. Feng, A. Kumar, B. Recht, and C. Ré. Towards a unified architecture for in-rdbms analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 325–336. ACM, 2012.

M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo. Convergence rate of incremental gradient and newton methods. *arXiv preprint arXiv:1510.08562*, 2015a.

M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo. Why random reshuffling beats stochastic gradient descent. *arXiv preprint arXiv:1510.08560*, 2015b.

M. Gürbüzbalaban, A. E. Ozdaglar, P. A. Parrilo, and N. D. Vanli. When cyclic coordinate descent outperforms randomized coordinate descent. In *NIPS*, 2017.

E. Hazan and S. Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

A. Israel, F. Krahmer, and R. Ward. An arithmetic–geometric mean inequality for products of three matrices. *Linear Algebra and its Applications*, 488:1–12, 2016.

P. Jain, D. Nagaraj, and P. Netrapalli. Sgd without replacement: Sharper rates for general smooth convex functions. *arXiv preprint arXiv:1903.01463*, 2019.

T. Kohonen. An adaptive associative memory principle. *IEEE Transactions on Computers*, 100(4):444–445, 1974.

C.-P. Lee and S. J. Wright. Random permutations fix a worst case for cyclic coordinate descent. *arXiv preprint arXiv:1607.08320*, 2016.

J. D. Lee, Q. Lin, T. Ma, and T. Yang. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *arXiv preprint arXiv:1507.07595*, 2015.

S. Ma, R. Bassily, and M. Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. *arXiv preprint arXiv:1712.06559*, 2017.

E. Moulines and F. R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

A. Nedić and D. Bertsekas. Convergence rate of incremental subgradient algorithms. In *Stochastic optimization: algorithms and applications*, pages 223–264. Springer, 2001.

A. Nemirovskii, D. B. Yudin, and E. R. Dawson. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Y. Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

L. M. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč. Sgd and hogwild! convergence without the bounded gradients assumption. *arXiv preprint arXiv:1802.03801*, 2018.

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

A. Rakhlin, O. Shamir, K. Sridharan, et al. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*. Citeseer, 2012.

B. Recht and C. Ré. Beneath the valley of the noncommutative arithmetic-geometric mean inequality: conjectures, case-studies, and consequences. *arXiv preprint arXiv:1202.4184*, 2012.

B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

M. Schmidt and N. L. Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

O. Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pages 46–54, 2016.

M. V. Solodov. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11(1):23–35, 1998.

R. Sun and Y. Ye. Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. *arXiv preprint arXiv:1604.07130*, 2016.

P. Tseng. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

S. Vaswani, F. Bach, and M. Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

S. J. Wright and C.-P. Lee. Analyzing random permutations for cyclic coordinate descent. *arXiv preprint arXiv:1706.00908*, 2017.

B. Ying, K. Yuan, S. Vlaski, and A. H. Sayed. Stochastic learning under random reshuffling. *arXiv preprint arXiv:1803.07964*, 2018.

T. Zhang. A note on the non-commutative arithmetic-geometric mean inequality. *arXiv:1411.5058*, 2014.