Why random reshuffling beats stochastic gradient descent

As Published: https://doi.org/10.1007/s10107-019-01440-w

Publisher: Springer Berlin Heidelberg

Persistent URL: https://hdl.handle.net/1721.1/132030

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Why random reshuffling beats stochastic gradient descent

Cite this article as: M. Gürbüzbalaban, A. Ozdaglar and P. A. Parrilo, Why random reshuffling beats stochastic gradient descent, Mathematical Programming https://doi.org/10.1007/s10107-019-01440-w

Noname manuscript No.
(will be inserted by the editor)

Why Random Reshuffling Beats Stochastic Gradient Descent

M. Gürbüzbalaban · A. Ozdaglar · P.A. Parrilo.

Received: date / Accepted: date

Abstract We analyze the convergence rate of the random reshuffling (RR) method, which is a randomized first-order incremental algorithm for minimizing a finite sum of convex component functions. RR proceeds in cycles, picking a uniformly random order (permutation) and processing the component functions one at a time according to this order, i.e., at each cycle, each component function is sampled without replacement from the collection. Though RR has been numerically observed to outperform its with-replacement counterpart stochastic gradient descent (SGD), characterization of its convergence rate has been a long standing open question. In this paper, we answer this question by providing various convergence rate results for RR and variants when the sum function is strongly convex. We first focus on quadratic component functions and show that the expected distance of the iterates generated by RR with stepsize $\alpha_k = \Theta(1/k^s)$ for $s \in (0, 1]$ converges to zero at rate $\mathcal{O}(1/k^s)$ (with $s = 1$ requiring adjusting the stepsize to the strong convexity constant). Our main result shows that when the component functions are quadratics or smooth (with a Lipschitz assumption on the Hessian matrices), RR with iterate averaging and a diminishing stepsize $\alpha_k = \Theta(1/k^s)$ for $s \in (1/2, 1)$ converges at rate $\Theta(1/k^{2s})$ with probability one in the suboptimality of the objective value, thus improving upon the $\Omega(1/k)$ rate of SGD. Our analysis draws on the theory of Polyak-Ruppert averaging and relies on decoupling the dependent cycle gradient error into an independent term over cycles and another term dominated by α_k^2. This allows us to apply law of large numbers to an appropriately weighted version of the cycle gradient errors, where the weights depend on the stepsize. We also provide high probability convergence rate estimates that

M. Gürbüzbalaban
Department of Management Science and Information Systems, Rutgers University, Piscataway, NJ, 08854. E-mail: mg1366@rutgers.edu

A. Ozdaglar
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA. E-mail: asuman@mit.edu.

P.A. Parrilo
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA. E-mail: parrilo@mit.edu

shows decay rate of different terms and allows us to propose a modification of RR with convergence rate $\mathcal{O}(\frac{1}{k^2})$.

1 Introduction: First-order incremental methods

We consider the following unconstrained optimization problem where the objective function is the sum of a large number of component functions:

$$\min f(x) := \sum_{i=1}^{m} f_i(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n, \tag{1}$$

with $f_i : \mathbb{R}^n \to \mathbb{R}$. This problem arises in many contexts and applications including regression or more generally parameter estimation problems (where $f_i(x)$ is the loss function representing the error between the output and the prediction of a parametric model) [2, 3, 5, 12], minimization of an expected value of a function (where the expectation is taken over a finite probability distribution or approximated by an m-sample average) [11,38], machine learning [38,41,42], or distributed optimization over networks [27, 28, 33].

One widely studied approach for solving problem (1) is the *deterministic incremental gradient (IG) method* [4–6]. IG method is similar to the standard gradient method with the key difference that at each iteration, the decision vector is updated incrementally by taking sequential steps along the gradient of the component functions f_i in a cyclic order. Hence, we can view each outer iteration k as a cycle of m inner iterations: starting from initial point $x_0^0 \in \mathbb{R}^n$, for each $k \geq 0$, we update the iterate x_i^k as

$$x_i^k := x_{i-1}^k - \alpha_k \nabla f_i(x_{i-1}^k), \qquad i = 1, 2, \ldots, m, \tag{2}$$

where $\alpha_k > 0$ is a stepsize with the convention that $x_0^{k+1} = x_m^k$.

Intuitively, it is clear that slow progress can be obtained if the functions that are processed consecutively have gradients close to zero. Indeed, the performance of IG is known to be pretty sensitive to the order functions are processed [6, Example 2.1.3] where *an order* σ is defined as a permutation of $\{1, 2, \ldots, m\}$. In some special cases when the component functions have a particular symmetry structure, there may be a *favorable order* σ to process the component functions which can lead to better performance than other choices of the order (see e.g. [6, Example 2.1.6]). IG iterations with respect to an order σ are of the form:

$$x_i^k := x_{i-1}^k - \alpha_k \nabla f_{\sigma(i)}(x_{i-1}^k), \qquad i = 1, 2, \ldots, m. \tag{3}$$

However, in general a favorable order is not known in advance, and a common approach is choosing the indices of functions to process as independent and uniformly distributed samples from the set $\{1, 2, \ldots, m\}$. This way no particular order is favored, making the method less vulnerable to particularly bad orders. This approach amounts to at each iteration sampling the function indices *with replacement* from the set $\{1, 2, \ldots, m\}$ and is called the *Stochastic Gradient Descent* (SGD) method, a.k.a. *Robbins-Monro* algorithm [37]. SGD is strongly related to the classical field of stochastic approximation [25]. Recently it has received a lot of attention due to its applicability to large-scale problems and became popular especially in machine learning applications (see e.g. [8, 9, 11, 43]).

An alternative popular approach that works well in practice is following a mixed approach between SGD and IG, sampling the functions randomly but not allowing repetitions, that is sampling the component functions at each iteration *without-replacement*, or equivalently picking a random order at each cycle. Specifically, at each cycle k, we draw a permutation σ_k of $\{1, 2, \dots, m\}$ independently and uniformly at random over the set of all permutations

$$\Gamma = \big\{\sigma \;:\; \sigma \text{ is a permutation of } \{1, 2, \dots, m\}\big\}, \tag{4}$$

and process the functions with this order:

$$x_i^k := x_{i-1}^k - \alpha_k \nabla f_{\sigma_k(i)}(x_{i-1}^k), \qquad i = 1, 2, \dots, m, \tag{5}$$

where $\alpha_k > 0$ is a stepsize. We set $x_0^{k+1} = x_m^k$ as before and refer to $\{x_0^k\}$ as the *outer iterates*. This method is called the *Random Reshuffling* (RR) method [6, Section 2.1] and will be the focus of this paper.

2 Motivation and summary of contributions

Without-replacement sampling schemes are often easier to implement efficiently compared to with-replacement sampling schemes, guarantee that every point in the data set is touched at least once, and often have better practical performance than their with-replacement counterparts [4,6,7,9,18,20,34,35]. For instance, Bottou [7] empirically compares SGD and RR methods and finds that RR converges with a rate close to $\sim 1/k^2$ whereas SGD is much slower achieving its min-max lower bound of $\Omega(1/k)$ for strongly convex objective functions [1,30]. This discrepancy in rate between RR and SGD is not only observed for large m but also for small m (as we illustrate in Example 1), and understanding it theoretically has been a long-standing open problem [4,6,35].

To our knowledge, the only existing theoretical analysis for RR is given by a recent paper of Recht and Ré [34] which focuses on least mean squared optimization and formulates a conjecture that would prove that the expected convergence rate of RR is faster than that of SGD. Given N arbitrary positive-definite matrices of dimension $n \times n$, the conjecture says that products of any K matrices chosen from this set of N matrices satisfy a non-commutative arithmetic-geometric mean inequality for every positive integer N and every $K \leq N$. This conjecture has been proven only in some special cases (for $N = 2$ [34], for $N = 3$ [23] and when N is a multiple of 3 and $K = 3$ [44]). Recht and Ré also analyze a special case of (1) (that arises when $f_i(x) = (a_i^T x - y_i)^2$ where a_i is a column vector that is randomly generated according to a random model and y_i is a scalar) and show that after a fixed amount of iterations, the upper bounds on the expected mean square error using without-replacement sampling is smaller than that of with-replacement sampling with high probability on most models of a_i (probabilities are taken with respect to the random data generation model). Despite these advances, there has been a lack of convergence theory for RR that characterizes its convergence rate and explains its fast performance. Analyzing algorithms based on without-replacement sampling such as RR is more difficult than with-replacement based approaches such as SGD. The reason is that the underlying independence assumption for the with-replacement sampling allows a tractable analysis with classical martingale convergence theory [26,31], whereas without-replacement sampling introduces correlations and dependencies among the sampled gradients and iterates that are harder to analyze [34]. The aim of our paper is to *fill this theoretical gap* for the case when the objective function f in (1) is strongly convex and *develop a novel algorithm that can accelerate the convergence further*. We next summarize our contributions.

We first consider the case when the component functions are quadratics. Building on the recent convergence rate results for the cyclic IG [21, Theorem 3.1], we first present a key result (Theorem 1) that provides an upper bound for the distance from the optimal solution of the iterates generated by an incremental method that processes component functions with an *arbitrary fixed order* and uses a stepsize $\Theta(1/k^s)$ for $s \in (0,1]$. This upper bound decays at rate $\mathcal{O}(1/k^s)$ and depends on the strong convexity constant of the sum function and an order dependent parameter given by a weighted average of Hessian matrices where the weights are given by the sum of the component gradients processed up to that point according to the given order. We use this result to show that the distance to the optimal solution of the iterates generated by RR algorithm with stepsize $\Theta(1/k^s)$, for all $s \in (0,1]$, converges to 0 at rate $\mathcal{O}(1/k^s)$ in expectation (where the expectation is over the random sequence of iterates). Moreover, we show that achieving the rate $\mathcal{O}(1/k)$ involves adapting the stepsize to the strong convexity constant of the sum function.

We then consider the q-suffix averages of the iterates generated by RR for some $q \in (0,1]$ (which is obtained by averaging the last qk iterates at iteration k) and show that with a stepsize $\alpha_k = R/(k+1)^s$ for $s \in (1/2,1)$ and $R > 0$, they converge *almost surely at rate $\mathcal{O}(1/k^s)$ to the optimal solution.* We provide an explicit characterization of the asymptotic rate constant in terms of the averaging parameter q, the stepsize parameters R and s and the Hessian matrices and the gradients of the component functions at the optimal solution (parts (i) and (ii) of Theorem 3). Using strong convexity, this implies an almost sure convergence rate $\Theta(1/k^{2s})$ in the suboptimality of the objective value. Our analysis views RR as a gradient descent method with random gradient errors. Since the permutations arising in each cycle of the RR algorithm are sampled independently, by conditioning on the last iterate from the prior cycle, we eliminate the cross-dependencies of the cumulative gradient error among the cycles in our approach. A key step in our proof is to decouple the cycle gradient error into a $\mathcal{O}(\alpha_k)$ term independent over cycles and another term that scales as $\mathcal{O}(\alpha_k^2)$. This allows us to use strong law of large numbers for a properly weighted average of the cycle error gradient sequence (where the weights depend on the stepsize) and show almost sure convergence of the q-suffix averaged iterates. Another key component of our analysis is to adapt the Polyak-Ruppert averaging techniques developed for SGD [26, 31] to RR.

We also provide a high probability convergence rate estimate for the distance of q-suffix averages to the optimal solution that consists of two terms, with the first term corresponding to a $1/k^s$ decay of a “bias” term (where bias is defined as the expected value of the cycle gradient errors of RR which may be non-zero) and the second term representing a $1/k$ decay for $0 < q < 1$ (and $\log k/k$ decay for $q = 1$); see part (iii) of Theorem 3 . These results are obtained by martingale concentration techniques. We use the characterization of the bias to estimate it with a term that can be computed during the RR iterations. We show that subtracting the estimated bias from the averaged RR iterates accelerates the convergence rate further, leaving only the second error term of $1/k$ decay in the iterates (part (iv) of Theorem 3). Based on this result, we propose a new algorithm which we call the *De-biased Random Reshuffling* (DRR) method that can accelerate the asymptotic convergence rate of RR in the suboptimality of the function values from $\mathcal{O}(1/k^{2s})$ to $\mathcal{O}(1/k^2)$.

Finally, in Theorem 4 we show that our results in Theorem 3 extend to the more general case when component functions are smooth (twice continuously differentiable) under a Lipschitz assumption on the Hessian, which allows us to control the second order term in a Taylor expansion of the gradient.

Outline: The outline of the paper is as follows. In Section 3, we introduce our approach for analyzing RR, present Polyak-Ruppert averaging and give a motivating example. Section 4 focuses on the case when component functions are quadratics. We first present a convergence rate estimate for IG with a fixed arbitrary order. We then focus on RR and study convergence of averaged iterates to the optimal solution. Section 5 extends our results to smooth functions. Section 6 proposes the DRR algorithm that can accelerate RR further. Finally, we conclude with a summary of our work in Section 7. Some of the technical lemmas required in the details of the proofs are deferred to Sections A, B and C of the Appendix.

Notation: We study the point-wise dominance of stochastic sequences by deterministic sequences and use the following notation. Let $x_k = x_k(\omega)$ be a stochastic real-valued sequence (where ω can be thought as the source of randomness) and y_k be a real-valued deterministic sequence. We write $x_k = \mathcal{O}(y_k) \iff \exists h > 0, \exists k_0 \quad \text{such that} \quad |x_k| \leq h|y_k| \quad \forall k \geq k_0$, for all ω, where h and k_0 are independent of ω (Note that the requirement is that this inequality holds for all ω, not just for almost all ω). When x_k is non-negative for every ω, given another deterministic positive sequence z_k, we also introduce the inequality version of this definition: $x_k \leq y_k + o(z_k) \iff \forall \varepsilon > 0, \exists k_0(\varepsilon) \quad \text{such that} \quad z_k^{-1}|x_k(\omega) - y_k| \leq \varepsilon, \quad \forall k \geq k_0(\varepsilon), \forall \omega$ where k_0 depends on ε but is independent of ω. When x_k is deterministic, these definitions reduce to the standard definitions of $\mathcal{O}(\cdot)$ and $o(\cdot)$ for deterministic sequences. For random x_k, the only difference is that we require the constants to be independent of the choice of ω. For example, if x_k is uniformly distributed over $[0, 10]$, we write $x_k = \mathcal{O}(1)$. Throughout the paper, $\|\cdot\|$ denotes the 2-norm for vectors or matrices, depending on the context. We also define the $\mathcal{O}(\cdot)$ notation beyond scalars for matrix- and vector-valued sequences analogously: Given a sequence of matrix-valued random variables $X_k(\omega) \in \mathbb{R}^{n\times p}$ where ω is the source of randomness, $n, p \geq 1$ are arbitrary fixed integers and a deterministic real-valued sequence y_k, we say $\|X_k(\omega)\| = \mathcal{O}(y_k)$ if and only if there exists $h > 0$ and k_0 such that $\|X_k(\omega)\| \leq h|y_k|$ for all $k \geq k_0$, for all ω, where h and k_0 are independent of ω. Note that when when $n = 1$ or $p = 1$, $\|\cdot\|$ is equivalent to the Euclidean norm.

3 Preliminaries

We consider solving problem (1) with RR method with iterations given in (5). Throughout we assume the following:

Assumption 1 *The sum function $f(x) = \sum_{i=1}^m f_i(x)$ is strongly convex, i.e., there exists a constant $c > 0$ such that the function $f(x) - \frac{c}{2}\|x\|^2$ is convex on $\mathbb{R}^n$.*[1]

Note that this assumption is on the sum function f, it does not require the convexity of the individual component functions f_i. A consequence of this assumption is that there exists a unique optimal solution to (1) which we denote by x^*. Another consequence is that the Hessian at the optimal solution is invertible since

$$H_* := \nabla^2 f(x^*) \succeq cI_n \succ 0, \tag{6}$$

where I_n is the $n \times n$ identity matrix.

To analyze RR, we view it as a gradient method with random gradient errors and rewrite the outer iterations (5) as

$$\frac{x_0^k - x_0^{k+1}}{\alpha_k} = \nabla f(x_0^k) + E_k, \tag{7}$$

where

$$E_k := \sum_{i=1}^m \left(\nabla f_{\sigma_k(i)}(x_{i-1}^k) - \nabla f_{\sigma_k(i)}(x_0^k) \right) \tag{8}$$

is the cumulative gradient errors associated with the cycle k. This approach is similar to the analysis of SGD, where one writes each (inner) iteration as a gradient method with error. The key difference that simplifies the analysis of SGD is the fact that the iteration gradient errors at the current iterate are independent (because of independent identically distributed (i.i.d.) sampling of component function indices) allowing use of martingale central limit theorems to obtain convergence and rate results (see e.g. [13, 17, 25, 31]). In contrast, for RR, not only are the iteration gradient errors dependent (because of sampling a random order at cycle k coupling indices $\sigma_k(i)$ and $\sigma_k(j)$ for $i \neq j$), but also the cycle gradient errors E_{k_1} and E_{k_2} for cycles $k_1 \neq k_2$ are dependent as they both depend on the history of the iterates. [2] Nevertheless, the analysis of RR is facilitated considerably by the fact that each cycle of RR is based on i.i.d. permutations. Therefore, by conditioning to the last iterate from the previous cycle, the analysis of the cumulative gradient errors E_k can be simplified - as noted from the proof of Theorem 2 below.

A key idea in our analysis is to use a recent upper bound for the convergence rate of cyclic incremental gradient method which applies to arbitrary fixed deterministic order (see Theorem 1). This bound implies an almost sure upper bound (in fact one that holds for all sample paths) on the distance of the outer iterates x_0^k generated by RR from the optimal solution x^* (see Section 4.1). Crucially, this result implies an upper bound in expected distance which is asymptotically m times smaller than the almost sure guarantees on the distance of the iterates.

[1] Such functions arise naturally in support vector machines and other regularized learning algorithms or regression problems (see e.g. [32, 36, 38])

[2] There is some literature that analyzes SGD under correlated noise [25, Ch. 6], but the noise needs to have a special structure (such as a mixing property) which does not seem to be applicable to the analysis of RR.

In analyzing RR, we will also consider the *average* of the outer iterate sequence given by $\bar{x}_k := \frac{\sum_{j=0}^{k-1} x_0^j}{k}$. We also consider averaging only the most recent iterates, i.e. at iteration k, averaging the last qk iterates for some constant $q \in (0,1]$:

$$\bar{x}_{q,k} := \frac{\sum_{j=(1-q)k}^{k-1} x_0^j}{qk}, \quad 0 < q \leq 1.$$

The generated sequence is referred to as the *q-suffix average* of the sequence x_0^k. For $q = 1$, we have $\bar{x}_{1,j} = \bar{x}_j$ and it is easy to see that we can compute this quantity based on the recursion

$$\bar{x}_j = (1 - \frac{1}{j})\bar{x}_{j-1} + \frac{1}{j} x_0^{j-1} \quad \text{for} \quad j = 1, 2, \dots, k. \tag{9}$$

Note that this requires storing only a vector of length n. For $0 < q < 1$ fixed, it can be verified after a straightforward computation that the q-suffix average satisfies the identity

$$\bar{x}_{q,k} = \frac{\bar{x}_{1,k} - (1-q)\bar{x}_{1,(1-q)k}}{q} = \frac{\bar{x}_k - (1-q)\bar{x}_{(1-q)k}}{q}.$$

Therefore, based on this identity, one can still use the recursion (9) to compute $\bar{x}_{q,k}$. Alternatively, $\bar{x}_{q,k}$ can be computed from the following recursion

$$\bar{y}_j = (1 - \frac{1}{j})\bar{y}_{j-1} + \frac{1}{j} x_0^{(1-q)k+j-1}, \quad \text{for} \quad j = 1, 2, \dots, qk, \tag{10}$$

with initialization $\bar{y}_0 = 0$ where it can be checked that $\bar{x}_{q,k} = \bar{y}_{qk}$. Hence, q-suffix averages can be computed efficiently in an online manner during the iterations, requiring only a memory of length n which is the dimension of the underlying optimization problem (1).

For SGD, it was shown that q-suffix averaging with $0 < q < 1$ leads to better performance then averaging (which corresponds to the $q = 1$ case by definition), improving the convergence rate in the suboptimality of the function value from $\log k/k$ to $1/k$ [32, 40]. This is in line with our results in Section 4 which show faster rate for the $0 < q < 1$ case. The parameter q can be thought as a measure of how much memory one uses during the averaging process. We define the *q-suffix average* of the stepsize in a similar way:

$$\bar{\alpha}_{q,k} = \frac{\sum_{j=(1-q)k}^{k-1} \alpha_j}{qk}, \quad 0 < q \leq 1. \tag{11}$$

We note that the *q-suffix average* of the stepsize can be computed in an online manner by a similar approach to the computation of $\bar{x}_{q,k}$ described above.

We will obtain our strongest convergence results (in the almost sure sense and with a similar m dependence as the expected guarantees) for averaged iterate sequences with “large step sizes”, a technique known as Polyak-Ruppert averaging, which has been used in achieving optimal rates for SGD in a robust manner as explained next.

3.1 Polyak-Ruppert averaging

SGD has a long history going back to the seminal paper of Robbins and Monro [37]. It has been analyzed under different assumptions extensively in the stochastic approximation literature (see e.g. [25]). For stochastic convex optimization, it has been shown that SGD has a min-max lower bound of $\Omega(1/k)$ [1, 30]. One way of achieving this optimal $1/k$ rate is to use a stepsize $\alpha_k = R/k$ where R is a positive scalar adjusted properly to the strong convexity constant of the objective function [13, 17, 25] but this requires the knowledge or the estimate of an accurate lower bound on the strong convexity constant. If a lower bound is not known or cannot be estimated accurately, the convergence can be potentially slow [29, Section 2.1]. Polyak-Ruppert averaging is a technique that allows to get the optimal $\sim 1/k$ rate in an asymptotically efficient manner without the need to adjust to the strong convexity constant. It relies on using a larger stepsize $\alpha_k = R/k^s$ (with R an arbitrary positive constant and $s \in (1/2, 1)$) that decays slower than $\Theta(1/k)$ but then taking the time average of the iterates to filter out the undesired oscillations arising due to the larger steps [25, 29, 31].[3] We will later show that the same technique allows us to get almost sure guarantees for the averaged iterates without the need to tune the stepsize to the strong convexity constant (see Theorems 3 and 4).

3.2 A motivating example

Before presenting our convergence analysis, we consider a simple example that highlights the difference in convergence mechanisms of SGD and RR and gives intuition on why RR is faster than SGD asymptotically.

Example 1 Consider the component functions

$$f_1(x) = \frac{1}{2}(x-1)^2, \quad f_2(x) = \frac{1}{2}(x+1)^2 + \frac{x^2}{2}, \tag{12}$$

with $f(x) = f_1(x) + f_2(x) = \frac{3}{2}x^2 + 1$ and $x^* = 0$. The outer RR iterates $\{x_0^k\}$ satisfy

$$x_0^{k+1} = x_0^k - \alpha_k \left(\nabla f_{\sigma_k(1)}(x_0^k) + f_{\sigma_k(2)}(x_1^k)\right) = x_0^k - \alpha_k(\nabla f(x_0^k) + E_k), \tag{13}$$

where the cycle gradient errors are given by

$$E_k = \begin{cases} \nabla f_2(x_1^k) - \nabla f_2(x_0^k) & \text{with probability } 1/2, \quad \text{for } \sigma_k = \{1,2\}, \\ \nabla f_1(x_1^k) - \nabla f_1(x_0^k) & \text{with probability } 1/2, \quad \text{for } \sigma_k = \{2,1\}. \end{cases} \tag{14}$$

Plugging in the identities $\nabla f_1(x) = x - 1$, $\nabla f_2(x) = 2x + 1$ obtained from (13) and the inner update formula (5), we obtain

$$E_k = \alpha_k \mu(\sigma_k) - 2\alpha_k x_0^k, \tag{15}$$

[3] IG shows similar properties to SGD in terms of the robustness of the stepsize rules $\alpha_k = R/k^s$. The convergence rate (in k) is only robust to the strong convexity constant of the objective for $s < 1$ but not for $s = 1$ [29, Section 2.1].

where $\mu(\sigma_k) = -\nabla^2 f_{\sigma_k(2)}(x^*)\nabla f_{\sigma_k(1)}(x^*)$ satisfying

$$\mu(\sigma_k) = \begin{cases} +2 & \text{with probability } 1/2, \quad \text{for } \sigma_k = \{1,2\}, \\ -1 & \text{with probability } 1/2, \quad \text{for } \sigma_k = \{2,1\}. \end{cases}$$

In contrast, SGD starting from an initial point y^0 leads to the iterations

$$y^{j+1} = y^j - \alpha_j \nabla f_{i_j}(y^j) = y^j - \frac{\alpha_j}{2}(\nabla f(y^j) + e^j), \tag{16}$$

where i_j is an independent and identically distributed (i.i.d.) random variable with a uniform distribution over the index set $\{1,2\}$ and the gradient error e^j is given by

$$e^j = \begin{cases} -2 - y^j & \text{with probability } 1/2, \quad \text{for } i_j = 1, \\ 2 + y^j & \text{with probability } 1/2, \quad \text{for } i_j = 2. \end{cases} \tag{17}$$

We consider a stepsize of $\alpha_k = \frac{R}{k^s}$ with $s = 0.75$ for both algorithms. Note that for this example, RR is globally convergent to the optimal solution $x^* = 0$ with probability one, therefore $x_0^j \to 0$.[4] By a similar argument, it can be shown that SGD is also convergent to the optimal solution $x^* = 0$ in mean-square, i.e. $\mathbb{E}\|y^j\|^2 \to 0$ (see also e.g. [25]). Then, it follows from (17) and (15) that the cumulative gradient error of SGD for any cycle k (defined as the cumulative sum $\sum_{j=(k-1)m}^{km-1} e_j)$) has zero expectation and $\Theta(1)$ variance whereas the gradient errors in RR are $E_k = \mathcal{O}(\alpha_k)$ with a typically non-zero expectation satisfying $\mathbb{E}(E_k) = \alpha_k(1 - 2x_0^k)$ and an asymptotically smaller variance $\mathcal{O}(\alpha_k^2)$ compared to SGD. In other words, the cycle gradient errors go to zero with probability one for RR whereas the gradient errors in SGD are typically bounded away from zero with a positive probability. Informally, this leads to a more accurate direction of descent for RR and is the main reason behind the faster convergence we demonstrate for RR compared to SGD in our analysis.

We also observe that the cycle gradient error E_k given by (15) consists of the sum of two terms: The first term is $\mathcal{O}(\alpha_k)$ and is independent over the cycles as the permutations σ_k are independent and identically distributed whereas the second term is of smaller (second) order as $x_0^j \to 0$. We will show later in Lemma 4 that such a decomposition can be obtained more generally when component functions are quadratics or they are smooth functions and will be a key step in the proof of Theorem 3.

Figure 1 compares the RR and SGD algorithms with averaging in terms of the histogram of the error (distance of the averaged iterates to the optimal solution x^*). In other words, we compare the approximation errors $\bar{x}_k - x^*$ and $\bar{y}_k - x^*$ where where $\bar{y}_k := \frac{\sum_{j=0}^{mk-1} y^j}{mk}$ is the averaged SGD iterates after k cycles (or equivalently mk inner iterations). For a fair comparison, both algorithms are run with the same parameters using $k = 500$ cycles over 10000 sample paths created for the Example (1) where $s = 0.75$. The left panel in Figure 1 compares the histograms

[4] To see this, note that the RR iterations for this example are given by $x_0^{k+1} = (1 - \frac{3}{2}\alpha_k + 2\alpha_k^2)x_0^k - \alpha_k^2\mu(\sigma_k)$ which implies, after taking norms of both sides and using the fact that $\|\mu(\sigma_k)\| \leq 2$, $\text{dist}_{k+1} \leq (1 - \frac{3}{2}\alpha_k + 2\alpha_k^2)\text{dist}_k + 2\alpha_k^2$. Then, by invoking classical results for the asymptotic behavior of non-negative sequences (see e.g. [6, Appendix A.4.3], we get $\text{dist}_{k+1} \to 0$. Theorem 1 also shows global convergence of RR on this example.

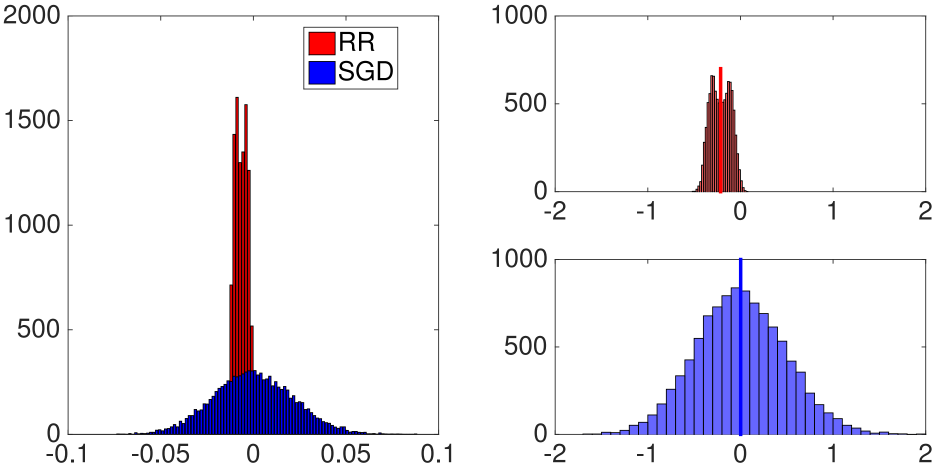


Fig. 1: Left panel: Comparison of the histogram of the approximation error $\bar{x}_k - x^*$ of the averaged iterates for RR and SGD after $k = 500$ cycles over 10000 sample paths created for the Example 1 with $s = 0.75$. Each sample path contains 1000 gradient computations for both RR and SGD. Right, top panel: Histogram of the scaled approximation error $k^s(\bar{x}_k - x^*)$ for RR iterates which is concentrated around the vertical line in red. Right, bottom panel: Histogram of the scaled approximation error $k^{1/2}(\bar{x}_k - x^*)$ for SGD which has the shape of a standard normal distribution. The vertical blue line passing through the origin is the axis of symmetry for this distribution indicating that this distribution is centered.

of $\bar{x}_k - x^*$ and $\bar{y}_k - x^*$ and shows that the approximation error $\bar{x}_k - x^*$ for RR is typically much smaller compared to that of SGD suggesting RR has a faster convergence rate. The top panel on the right illustrates that the scaled approximation error $k^s(\bar{x}_k - x^*)$ is concentrated around its mean (marked by the red line) suggesting $\mathcal{O}(1/k^s)$ convergence rate almost surely for the averaged RR iterates. On the other hand, the bottom panel on the right shows that the distribution of $k^{1/2}(\bar{y}_k - x^*)$ is approximately a standard normal distribution as predicted by the theory [31], illustrating the $\mathcal{O}(1/k^{1/2})$ convergence rate of the averaged SGD iterates to the optimal solution x^* in distribution. In Section 4, we will develop the first convergence theory for RR, establishing the $\mathcal{O}(1/k^s)$ convergence rate we observe in the numerical experiments and show that $k^s(\bar{x}_k - x^*)$ converges almost surely to a point for which we provide an explicit formula.

4 Quadratic component functions

We first consider quadratic component functions which allows an elegant analysis without the need to approximate higher order terms. We will show in Section 5 that the same line of analysis extends to smooth component function under a Lipschitz assumption on the Hessian matrices. Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be a quadratic function of the form

$$f_i(x) = \frac{1}{2} x^T P_i x - q_i^T x + r_i, \quad i = 1, 2, \dots, m, \tag{18}$$

where P_i is a symmetric $n \times n$ matrix, $q_i \in \mathbb{R}^n$ is a column vector and r_i is a scalar. Note that f_i has Lipschitz gradients, i.e.,

$$\|\nabla f_i(y) - \nabla f_i(z)\| \leq L_i \|y - z\|, \quad \forall y, z \in \mathbb{R}^n,$$

where $L_i = \|P_i\|$. It follows from the triangle inequality that f has Lipschitz gradients with Lipschitz constant at most

$$L := \sum_{i=1}^{m} L_i. \tag{19}$$

Moreover, Assumption 1 implies that the Hessian matrix of the sum satisfies $\nabla^2 f(x) = \sum_{i=1}^m \nabla^2 f_i(x) = \sum_{i=1}^m P_i \geq cI_n > 0$. Therefore, the solution x^* to (1) is unique.

4.1 Convergence Rate

Our convergence analysis of RR builds on a recent upper bound for convergence rate of (deterministic) cyclic IG method (see [21, Theorem 3.1]), which applies to any fixed permutation σ of $\{1, 2, \dots, m\}$. This result implies an upper bound (for all sample paths) on the distance to the optimal solution of the iterates generated by RR which is presented next. For our analysis throughout this paper, we introduce the Lyapunov function

$$\text{dist}_k := \|x_0^k - x_*\|, \tag{20}$$

which is the distance of the iterates to the optimal solution. Note that this quantity is deterministic for the IG method with a fixed order σ, whereas it is random for the RR method as the order σ is selected randomly for RR.

Theorem 1 *[21, Theorem 3.1] Let Assumption 1 hold. Let $f_i(x)$ be a quadratic function of the form $f_i(x) = \frac{1}{2} x^T P_i x - q_i^T x + r_i$ where P_i is a symmetric $n \times n$ matrix, $q_i \in \mathbb{R}^n$ is a column vector and r_i is a scalar for $i = 1, 2, \dots, m$. Suppose Assumption 1 holds. Consider the iterates $\{x_0^k\}$ generated by the iterations* (5) *with a fixed order σ and stepsize $\alpha_k = R/(k+1)^s$ where $R > 0$ and $s \in (1/2, 1)$. Then*[5]*,*

$$dist_k \leq \frac{R\|\mu(\sigma)\|}{c} \frac{1}{k^s} + o(\frac{1}{k^s}) \qquad if \quad 1/2 < s < 1, \tag{21}$$

$$dist_k \leq \frac{R^2\|\mu(\sigma)\|}{Rc - 1} \frac{1}{k} + o(\frac{1}{k}) \qquad if \quad s = 1 \text{ and } Rc > 1, \tag{22}$$

[5] The original result in [21, Theorem 3.1] was stated for $\sigma = \{1, 2, \dots, m\}$ but here we translate this result into an arbitrary permutation σ of $\{1, 2, \dots, m\}$ by noting that processing the set of functions $\{f_1, f_2, \dots, f_m\}$ with order σ is equivalent to processing the permuted functions $\{f_{\sigma_1}, f_{\sigma_2}, \dots, f_{\sigma_m}\}$ with order $\{1, 2, \dots, m\}$.

where c is the strong convexity constant of the sum function $f(x)$ and

$$\mu(\sigma) = -\sum_{1\leq i<j\leq m} P_{\sigma(j)} \nabla f_{\sigma(i)}(x^*). \tag{23}$$

This theorem provides an upper bound on the rate with a rate constant $\mu(\sigma)$ that depends on the order σ. Note that the best rate that IG with a fixed order σ can attain in terms of upper bounds is $\mathcal{O}(1/k)$ and requires a stepsize $R/(k+1)$ with $R > 1/c$ (see also [21, Theorem 3.4] for the lower bound of $\Omega(1/k)$ for IG under some conditions). We next provide some upper bounds on $\mu(\sigma)$. We define

$$G_* := \sup_{1\leq i\leq m} \|\nabla f_i(x^*)\|, \tag{24}$$

$$M_\Gamma := \sup_{\sigma\in\Gamma} \|\mu(\sigma)\|. \tag{25}$$

Using $L_i = \|P_i\|$ for each i, it follows from the triangle inequality that

$$\begin{aligned} \|\mu(\sigma)\| \leq M_\Gamma &\leq \sup_{\sigma\in\Gamma} \sum_{1\leq i<j\leq m} L_{\sigma(j)} \|\nabla f_{\sigma(i)}(x^*)\| = \sup_{\sigma\in\Gamma} \sum_{j=2}^{m} L_{\sigma(j)} \sum_{i=1}^{j-1} \|\nabla f_{\sigma(i)}(x^*)\| \\ &\leq \sup_{\sigma\in\Gamma} \sum_{j=2}^{m} L_{\sigma(j)} (j-1) G_* \\ &\leq (m-1) G_* \sup_{\sigma\in\Gamma} \sum_{j=2}^{m} L_{\sigma(j)} \leq L(m-1)G_*, \end{aligned} \tag{26}$$

where we used the definitions of the Lipschitz constant L and the gradient bound G_* from (19) and (24) respectively. By replacing $\mu(\sigma)$ by M_Γ in Theorem 1 one can get an upper bound on the worst-case convergence rate that applies to any choice of fixed order σ. Using a similar argument along the lines of the proof of Theorem 1 on the convergence rate of IG, it is straightforward to show that RR never performs any slower than this worst-case convergence rate which is the subject of the next result.

Corollary 1 *Under the setting of Theorem 1, if σ is sampled uniformly at each cycle instead of being kept fixed, then*

$$dist_k \leq \frac{RM_\Gamma}{c}\frac{1}{k^s} + o(\frac{1}{k^s}) \qquad \text{if} \quad 1/2 < s < 1, \tag{27}$$

$$dist_k \leq \frac{R^2 M_\Gamma}{Rc-1}\frac{1}{k} + o(\frac{1}{k}) \qquad \text{if} \quad s = 1 \text{ and } Rc > 1, \tag{28}$$

with probability one where M_Γ is deterministic and is defined by (25).

Corollary 1 provides a simple worst-case upper bound on the rate, however the rate constant $M_\Gamma = \sup_{\sigma\in\Gamma}\|\mu(\sigma)\|$ is pessimistic and can be thought as a worst-case performance measure that holds for every sample path. One way to get better constants is to consider convergence in expectation, a weaker notion of convergence compared to almost sure convergence. In the next theorem, we show that M_Γ can be improved to a typically much smaller constant $\|\bar{\mu}\|$ where

$$\bar{\mu} := \mathbb{E}\big(\mu(\sigma_1)\big) = \frac{\sum_{\sigma\in\Gamma}\mu(\sigma)}{|\Gamma|} \tag{29}$$

can be thought as a measure of average performance over the choice of random permutations.

Theorem 2 *Let $f_i(x)$ be a quadratic function of the form $f_i(x) = \frac{1}{2}x^T P_i x - q_i^T x + r_i$, where P_i is a symmetric $n\times n$ matrix, $q_i \in \mathbb{R}^n$ is a column vector and r_i is a scalar for $i = 1, 2, \ldots, m$. Suppose Assumption 1 holds. Consider the iterates $\{x_0^k\}$ generated by the RR iterations* (5) *and stepsize $\alpha_k = R/(k+1)^s$ where $R > 0$ and $s \in (0, 1]$. Then,*

$$\mathbb{E}\left(dist_k\right) \le \frac{R\|\bar{\mu}\|}{c}\frac{1}{k^s} + o(\frac{1}{k^s}) \qquad \textit{if} \quad 1/2 < s < 1, \tag{30}$$

$$\mathbb{E}\left(dist_k\right) \le \frac{R^2\|\bar{\mu}\|}{Rc-1}\frac{1}{k} + o(\frac{1}{k}) \qquad \textit{if} \quad s = 1 \textit{ and } Rc > 1, \tag{31}$$

where the expectation is taken over the sequence of iterates, $\bar{\mu}$ is defined by (29).

Remark 1 A consequence of Lemma 3 proved in the Appendix is that

$$\bar{\mu} = \frac{1}{2}\sum_{i=1}^{m} P_i \nabla f_i(x^*), \tag{32}$$

where $\bar{\mu}$ is defined by (29). By the triangle inequality, $\|\bar{\mu}\| \le \sum_{i=1}^m L_i G_* = LG_*$ where G_* is defined by (24). This upper bound is $m-1$ times smaller than the previous upper bound on M_Γ in (26).

It is also natural to ask what would happen to the rate constants and to the rate if one would take stepsize $\alpha_k = \Theta(1/k^s)$ and apply (Polyak-Ruppert) averaging to the RR iterates, especially given the fact that $\mathcal{O}(1/k^s)$ stepsize used in averaging does not require adjustment of the parameter R to the strong convexity level. More generally, one could consider q-suffix averaging. In the next section, we show that for the averaged RR iterates, similar upper bounds in (30) hold not only in expectation but also in probability. Another benefit of averaging is that it allows us to estimate and subtract *the bias term* in the iterations to get a more accurate estimation of the optimal solution as we discuss later in part (*iii*) of Theorem 3 and in Section 6.

4.2 Convergence rate with averaging

The following theorem characterizes the rate of convergence of the averages of iterates generated by RR. Part (*i*) and (*ii*) of this theorem show that q-suffix averages of the RR iterates converge at rate $1/k^s$ to the optimal solution almost surely with a stepsize $\Theta(1/k^s)$ for $s \in (1/2, 1)$. By gradient Lipschitzness, this translates into a rate of $\Theta(1/k^{2s})$ for the suboptimality of the objective value. The result is based on decoupling the cycle gradient errors E_k into a $\Theta(\alpha_k)$ term independent over the cycles and another $\mathcal{O}(\alpha_k^2)$ term that becomes negligible in the limit. Part (iii) is a high-probability convergence rate estimate for the approximation error $\bar{x}_{q,k} - x^*$. The approximation error consists of two terms, the first term $b_{q,k}$ which we call the "bias" term is deterministic and decays like $\sim 1/k^s$. It comes from the expected value of the independent part of the gradient cycle errors which may be different than zero. The second part is on the order of $1/k$ for $0 < q < 1$ (and $\log k/k$ when $q = 1$) and it is based on the Azuma-Hoeffding inequality for martingale concentration. Finally, part (iv) is on estimating the bias term $b_{q,k}$ with another quantity $\hat{b}_{q,k}$. It shows that by subtracting the estimated bias from the averaged iterates, we can approximate the optimal solution x^* up to an $\mathcal{O}(1/k)$ error in distances or equivalently up to an $\mathcal{O}(1/k^2)$ error in the suboptimality of the objective value. In Section 6, this result will be fundamental for Algorithm 1 that accelerates the convergence of RR from $\Theta(1/k^{2s})$ to $\mathcal{O}(1/k^2)$ with high probability in the suboptimality of the objective value.

Theorem 3 *Let $f_i(x)$ be a quadratic function of the form*

$$f_i(x) = \frac{1}{2} x^T P_i x - q_i^T x + r_i,$$

where P_i is a symmetric $n \times n$ matrix, $q_i \in \mathbb{R}^n$ is a column vector and r_i is a scalar for $i = 1, 2, \dots, m$. Consider the q-suffix averages $\bar{x}_{q,k}$ of the RR iterates generated by the iterations (5) *with stepsize $\alpha_k = \frac{R}{(k+1)^s}$ where $R > 0$ and $s \in (\frac{1}{2}, 1)$. Suppose that Assumption 1 holds. Then the following statements are true:*

(i) *For any $0 < q \leq 1$, the q-suffix averaged stepsize $\bar{\alpha}_{q,k}$ defined in* (11) *satisfies*

$$\bar{\alpha}_{q,k} = \frac{a_q(s)}{k^s} + \mathcal{O}(\frac{1}{k}) \quad \textit{where} \quad a_q(s) = \frac{1 - (1-q)^{1-s}}{q(1-s)} R. \tag{33}$$

(ii) *For any $0 < q \leq 1$, we have*

$$\lim_{k \to \infty} \frac{\bar{x}_{q,k} - x^*}{\bar{\alpha}_{q,k}} = -H_*^{-1} \bar{\mu} \quad a.s., \tag{34}$$

where $\bar{\mu}$ is given by (32), *i.e., the normalized error $(\bar{x}_{q,k} - x^*)/\bar{\alpha}_{q,k}$ converges to the constant vector $-H_*^{-1}\bar{\mu}$ almost surely where $H_* = \sum_{i=1}^m P_i$ is the Hessian matrix at the optimal solution and $\bar{\mu}$ is given by* (32). *Then, from part* (i),

$$\lim_{k \to \infty} k^s (\bar{x}_{q,k} - x^*) = -a_q(s) H_*^{-1} \bar{\mu} \quad a.s. \tag{35}$$

Hence, the q-suffix averaged iterates $\bar{x}_{q,k}$ converge to the optimal solution x^* with rate $1/k^s$ almost surely.

(iii) *We have*

$$\bar{x}_{q,k} - x^* = b_{q,k} + \frac{1}{k} e_{q,k} + + \begin{cases} \mathcal{O}\big(\frac{\log k}{k}\big) & \textit{if} \quad q = 1 \\ \mathcal{O}\big(\frac{1}{k}\big) & \textit{if} \quad 0 < q < 1, \end{cases}$$

where $\|e_{q,k}\| \leq B\sqrt{\log(1/\delta)}$ with probability $1 - \delta$ for a deterministic constant $B = \mathcal{O}(1)$,

$$b_{q,k} = -\bar{\alpha}_{q,k} H_*^{-1} \bar{\mu} \tag{36}$$

is deterministic, $\bar{\mu}$ is given by (32) *and $\bar{\alpha}_{q,k}$ is the averaged stepsize defined in* (11). *The constants hidden by $\mathcal{O}(\cdot)$ depend only on G_*, L, m, R, c, q and s.*

(*iv*) *Let*

$$\hat{b}_{q,k} = -\bar{\alpha}_{q,k} \left[\sum_{i=1}^{m} P_{\sigma_k(i)} \right]^{-1} \sum_{i=1}^{m} P_{\sigma_k(i)} \nabla f_{\sigma_k(i)}(x_{i-1}^k)/2, \tag{37}$$

where $\bar{\alpha}_{q,k}$ *is the averaged stepsize defined in* (11)*. Then,* $\hat{b}_{q,k} = b_{q,k} + \mathcal{O}(\alpha_k^2)$*. It follows from part* (*ii*) *that*

$$(\bar{x}_{q,k} - \hat{b}_{q,k}) - x^* = \frac{1}{k} e_{q,k} + \begin{cases} \mathcal{O}\left(\frac{\log k}{k}\right) & \text{if} \quad q = 1 \\ \mathcal{O}\left(\frac{1}{k}\right) & \text{if} \quad 0 < q < 1. \end{cases}$$

where $\|e_{q,k}\| \leq B\sqrt{\log(1/\delta)}$ *with probability* $1 - \delta$ *for a constant* $B = \mathcal{O}(1)$*.*

Proof (*i*) As the stepsize sequence is monotonically decreasing, we have the bounds

$$\int_{(1-q)k}^{k} \frac{R}{(x+2)^s} dx \leq \sum_{j=(1-q)k}^{k} \alpha_j = \sum_{j=(1-q)k}^{k} \frac{R}{(k+1)^s} \leq R + \int_{(1-q)k}^{k-1} \frac{R}{(x+1)^s} dx.$$

Dividing each term by qk, after a straightforward integration we obtain

$$\bar{\alpha}_{q,k} = \frac{k^{1-s} - \big((1-q)k + 1\big)^{1-s} + \mathcal{O}(1)}{(1-s)qk} R = \frac{a_q(s)}{k^s} + \mathcal{O}\left(\frac{1}{k}\right),$$

which completes the proof.

(*ii*) Taking the q-suffix averages of both sides of (7), we obtain

$$I_{q,k} := \frac{\sum_{j=(1-q)k}^{k-1} (x_0^j - x_0^{j+1}) \alpha_j^{-1}}{qk} = \frac{\sum_{j=(1-q)k}^{k-1} \nabla f(x_0^j) + E_j}{qk}. \tag{38}$$

As f is a quadratic, the first order Taylor series for the gradient of f is exact:

$$\nabla f(x_0^j) = H_*(x_0^j - x^*). \tag{39}$$

Therefore, (38) becomes $I_{q,k} = \frac{\sum_{j=(1-q)k}^{k-1} H_*(x_0^j - x^*) + E_j}{qk}$ which is equivalent to

$$I_{q,k} = H_*(\bar{x}_{q,k} - x^*) + \frac{\sum_{j=(1-q)k}^{k-1} E_j}{qk} = H_*(\bar{x}_{q,k} - x^*) + \bar{\alpha}_{q,k} Y_{q,k}, \tag{40}$$

where $Y_{q,k}$ is defined as

$$Y_{q,k} := \frac{1}{\bar{\alpha}_{q,k}} \frac{\sum_{j=(1-q)k}^{k-1} E_j}{qk} = \frac{\sum_{j=(1-q)k}^{k-1} E_j}{\sum_{j=(1-q)k}^{k-1} \alpha_j} \tag{41}$$

and can be interpreted as the (q-suffix) averaged gradient error sequence E_j normalized by the (q-suffix) averaged stepsize sequence α_j. Since H_* is invertible by the strong convexity of f (see (6)), we can rewrite (40) as

$$\begin{aligned} \bar{x}_{q,k} - x^* &= -H_*^{-1} \bar{\alpha}_{q,k} Y_{q,k} + H_*^{-1} I_{q,k} \\ &= -H_*^{-1} \bar{\alpha}_{q,k} Y_{q,k} + \begin{cases} \mathcal{O}\left(\frac{1}{k}\right) & \text{if} \quad 0 < q < 1 \\ \mathcal{O}\left(\frac{\log k}{k}\right) & \text{if} \quad q = 1, \end{cases} \end{aligned} \tag{42}$$

where we used the inequality $\|H_*^{-1}\| \leq 1/c$ implied by (6) and Lemma 2 from the appendix to provide an upper bound for the second term in the first equality. Note that, as a consequence of Lemma 2, $\mathcal{O}(\cdot)$ notation above hides a constant that depends only on the parameters G_*, L, c, m, R, s, q and also dist_0 when $q = 1$. Then, dividing both sides of (42) by $\bar{\alpha}_{q,k}$, taking limits as k goes to infinity, using part (i) on the asymptotic behavior of $\bar{\alpha}_{q,k}$ and the fact that $Y_{q,k} \to \bar{\mu}$ a.s. from Lemma 4, we obtain the claimed result.

(iii) By parts (i) and (iii) of Lemma 4 from the appendix that relates the gradient error sequence E_j to a sequence of i.i.d. variables $\mu(\sigma_j)$, for $0 < q \leq 1$,

$$Y_{q,k} = \frac{\sum_{j=(1-q)k}^{k-1} E_j}{\sum_{j=(1-q)k}^{k-1} \alpha_j} = \frac{\sum_{i=(1-q)k}^{k-1} \alpha_j \mu(\sigma_j) + \mathcal{O}(\alpha_j^2)}{\sum_{j=(1-q)k}^{k-1} \alpha_j}. \tag{43}$$

We first give a proof for $q = 1$, the proof for the remaining $q \in (0, 1)$ case will be similar. Assume $q = 1$. Plugging $q = 1$ and (43) into (42), we obtain

$$\begin{aligned}
\bar{x}_{1,k} - x^* &= \mathcal{O}(\frac{\log k}{k}) - H_*^{-1} \bar{\alpha}_{1,k} Y_{1,k} \\
&= \mathcal{O}(\frac{\log k}{k}) - H_*^{-1} \left(\frac{\sum_{j=0}^{k-1} \alpha_j \big(\mu(\sigma_j) - \bar{\mu}\big)}{k} + \frac{\sum_{j=0}^{k-1} \alpha_j \bar{\mu} + \mathcal{O}(\alpha_j^2)}{k} \right) \\
&= b_{1,k} + \mathcal{O}(\frac{\log k}{k}) - H_*^{-1} \frac{\sum_{j=0}^{k-1} \alpha_j (\mu(\sigma_j) - \bar{\mu})}{k} - H_*^{-1} \sum_{j=0}^{k-1} \frac{\mathcal{O}(\alpha_j^2)}{k} \\
&= b_{1,k} + \mathcal{O}(\frac{\log k}{k}) - H_*^{-1} \frac{\sum_{j=0}^{k-1} \alpha_j (\mu(\sigma_j) - \bar{\mu})}{k},
\end{aligned} \tag{44}$$

where $b_{1,k}$ is defined by (36) and we used in the last step the fact that for $s > 1/2$

$$\sum_{j=0}^{\infty} \alpha_j^2 = \sum_{j=1}^{\infty} \frac{R^2}{j^{2s}} = R^2 \zeta(2s) < \infty, \tag{45}$$

where $\zeta(\cdot)$ is the Riemann-Zeta function. We now study the asymptotic behavior of the last summation term in (44) by introducing the process $S_{1,k} = \sum_{j=0}^{k-1} Z_j$, where $Z_j := \alpha_j (\mu(\sigma_j) - \bar{\mu})$ and $k \geq 0$ with the convention that $S_{1,0} = 0$. Equipped with this definition, (44) becomes

$$\bar{x}_{1,k} - x^* = b_{1,k} + \mathcal{O}(\frac{\log k}{k}) + e_{1,k}, \quad e_{1,k} := -H_*^{-1} \frac{S_{1,k}}{k}. \tag{46}$$

The random variables Z_j are independent, centered and have an identical distribution up to the scaling factor α_j. Therefore, $S_{1,k}$ is a sum of centered random variables satisfying:

$$\|S_{1,k} - S_{1,k-1}\| = \big\|\alpha_{k-1}\big(\mu(\sigma_{k-1}) - \bar{\mu}\big)\big\| \leq \gamma_{k-1} := \alpha_{k-1} L m G_*, \tag{47}$$

where we used (75) in the last inequality (see also Lemma 3). Then, by the Azuma-Hoeffding inequality, for every $t > 0$,

$$\mathbb{P}\left(\|\frac{S_{1,k}}{k}\| > \frac{t}{k} \right) \leq 2 \exp \left(- \frac{t^2}{2 \sum_{j=0}^{k-1} \gamma_j^2} \right) = 2 \exp\big(- \frac{t^2}{\beta} \big),$$

where $\beta = 2\sum_{j=0}^{\infty} \gamma_j^2 < \infty$ as α_j is square-summable (see (45)). Note that β depends only on G_*, L, m and the stepsize parameters R and s. It is easy to see that selecting $t \geq t_\delta = \sqrt{\beta \log(2/\delta)}$ makes the right-hand side $\leq \delta$. Therefore for any $\delta > 0$, with probability at least $1-\delta$,

$$\|\frac{S_{1,k}}{k}\| \leq \frac{\sqrt{\beta \log(2/\delta)}}{k}, \tag{48}$$

which if inserted into the expression (46) completes the proof for the $q = 1$ case. For $0 < q < 1$ case, the same line of reasoning applies except that we replace $b_{1,k}$ with $b_{q,k}$ and we can improve the $\mathcal{O}(\log k/k)$ term in the expression (46) to $\mathcal{O}(1/k)$, this is justified by (42). Then, this leads to

$$\bar{x}_{q,k} - x^* = b_{q,k} + \mathcal{O}(\frac{1}{k}) + e_{q,k}, \quad e_{q,k} := -H_*^{-1}\frac{S_{q,k}}{qk}, \tag{49}$$

where $S_{q,k} := \sum_{j=(1-q)k}^{k-1} Z_j = S_{1,k} - S_{1,(1-q)k}$ is the q-suffix cumulative sum (cumulative sum of the last qk terms) of the sequence Z_k. Then using (48), with probability at least $1-\delta$,

$$\|\frac{S_{q,k}}{k}\| \leq \|\frac{S_{1,k}}{k}\| + \|\frac{S_{1,(1-q)k}}{k}\| \leq \frac{2t_\delta}{k}. \tag{50}$$

Plugging this high probability bound into (49), we conclude.

(iv) By Lemma 1, we have $\max\limits_{1\leq i<m} \|x_{i-1}^k - x^*\| = \mathcal{O}(\alpha_k)$. Therefore,

$$\|\nabla f_{\sigma_k(i)}(x_{i-1}^k) - \nabla f_{\sigma_k(i)}(x^*)\| = \mathcal{O}(\alpha_k), \tag{51}$$

for any $i = 1, 2, \ldots, m$. As a consequence,

$$\begin{aligned}\hat{b}_{q,k} &= -\bar{\alpha}_{q,k} H_*^{-1} \sum_{i=1}^{m} P_{\sigma_k(i)}\Big(\nabla f_{\sigma_k(i)}(x^*) + \mathcal{O}(\alpha_k)\Big) \\ &= -\bar{\alpha}_{q,k} H_*^{-1} \sum_{j=1}^{m} P_j \nabla f_j(x^*) + \mathcal{O}(\alpha_k^2) = b_{q,k} + \mathcal{O}(\alpha_k^2),\end{aligned}$$

where in the second equality we use the fact that $\bar{\alpha}_{q,k} = \mathcal{O}(1/k^s) = \mathcal{O}(\alpha_k)$ implied by part (i).

5 Extension to smooth component functions

Extending our results to more general smooth functions requires obtaining similar bounds for the cycle gradient errors which depend on the gradients and Hessian matrices of the component functions along the inner iterates. In order to be able to control the change of gradients and Hessian matrices along the iterates, we introduce the following assumption which has also been used to analyze SGD [26].

Assumption 2 *The functions f_i are convex on $\mathbb{R}^n$ and have Lipschitz continuous second derivatives, i.e. there exists a constant U_i such that*

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq U_i \|x - y\|, \quad \forall x, \forall y \in \mathbb{R}^n,$$

for $i = 1, 2, \ldots, m$.

Under this assumption, by the triangle inequality, $\nabla^2 f(\cdot)$ is also Lipschitz with constant $U := \sum_{i=1}^m U_i$. When the component functions are quadratics, we have the special case with $U = U_i = 0$. We will now see how this assumption makes it possible to control the change of gradients of the component functions. Smooth functions f with Lipschitz Hessians are quadratic-like in the sense that the first-order Taylor approximation to the gradient of f is almost affine (with a quadratic term controlled by the parameter U) satisfying

$$\nabla f(x) = \nabla f(x^*) + H_*(x - x^*) + \eta, \quad \|\eta\| \leq \frac{U}{2}\|x - x^*\|^2, \quad \forall x, \tag{52}$$

(see e.g. [19, Section 1.3]) The analysis of Theorem 3 (and Lemma 4 it builds upon) considers the $U = 0$ case (see e.g. (39) and (51)) applying a first-order Taylor approximation to the gradient of the component functions at $x = x_0^k$ where $\|x - x^*\| = \|x_0^k - x^*\| = \mathcal{O}(\alpha_k)$ by Lemma 1. Therefore, when $U \neq 0$, an extra correction term $\eta = \mathcal{O}(\alpha_k^2)$ needs to be added to the analysis. However, we show in the next theorem that this correction term does not cause a slow down in the convergence rate (in terms of dependency in k) compared to the quadratic case because the q-suffix averages of this $\mathcal{O}(\alpha_k^2)$ correction term decays like $\mathcal{O}(1/k)$.[6]

We will also need one more technical assumption that appeared in a number of papers in the literature for analyzing incremental methods to rule out the case that the iterates diverge to infinity. In particular, this assumption is also made in [21, Assumption 3.4] for generalizing Theorem 1 on the rate of deterministic IG from quadratic functions to general smooth functions.

Assumption 3 *Iterates $\{x_j^k\}_{j,k}$ generated are uniformly bounded, i.e. there exists a non-empty closed Euclidean ball $\mathcal{X} \subset \mathbb{R}^n$ that contains all the iterates a.s.*[7]

Equipped with these two assumptions, all the results of Theorem 3 extend naturally with minor modifications. In particular, P_i (which is a constant Hessian matrix in the setting of Theorem 3) needs to be replaced by $\nabla^2 f_i(x^*)$ or $\nabla^2 f_i(x_{i-1}^k)$ depending on the context.

Theorem 4 *Consider the RR iterations given by* (5) *with stepsize $\alpha_k = \frac{R}{(k+1)^s}$ where $R > 0$ and $s \in (\frac{1}{2}, 1)$. Suppose that Assumptions 1, 2 and 3 hold. Then the following statements are true:*

(i) For any $0 < q \leq 1$, $\lim_{k\to\infty} k^s(\bar{x}_{q,k} - x^*) = -a_q(s)H_*^{-1}\bar{v}$ a.s. where $H_* = \nabla^2 f(x^*)$ is the Hessian matrix at the optimal solution, $a_q(s)$ is defined by (33) *and*

$$\bar{v} := \frac{1}{2}\sum_{i=1}^m \nabla^2 f_i(x^*)\nabla f_i(x^*). \tag{53}$$

(ii) We have

$$\bar{x}_{q,k} - x^* = r_{q,k} + \frac{1}{k}\hat{e}_{q,k} + \begin{cases} \mathcal{O}\big(\frac{\log k}{k}\big) & \text{if} \quad q = 1, \\ \mathcal{O}\big(\frac{1}{k}\big) & \text{if} \quad 0 < q < 1, \end{cases}$$

[6] This is due to the fact that the sequence α_k^2 is summable when $s > 1/2$.

[7] Note that if this assumption holds and if f_i is three-times continuously differentiable on the compact set $\mathcal{X}$, then the third-order derivatives are bounded and Assumption 2 holds.

where $\|\hat{e}_{q,k}\| \le \hat{B}\sqrt{\log(1/\delta)}$ *with probability* $1-\delta$ *for a deterministic constant* $\hat{B} = \mathcal{O}(1)$ *and*

$$r_{q,k} = -\bar{\alpha}_{q,k} H_*^{-1} \bar{v} \tag{54}$$

is deterministic. The constants hidden by $\mathcal{O}(\cdot)$ *depend only on* G_*, L, m, R, c, q, s *and* U.

(iii) *Let*

$$\hat{r}_{q,k} = -\bar{\alpha}_{q,k}\left[\sum_{i=1}^{m} \nabla^2 f_{\sigma_k(i)}(x_{i-1}^k)\right]^{-1} \sum_{i=1}^{m} \nabla^2 f_{\sigma_k(i)}(x_{i-1}^k) \nabla f_{\sigma_k(i)}(x_{i-1}^k)/2.$$

Then, $\hat{r}_{q,k} = r_{q,k} + \mathcal{O}(\alpha_k^2)$. *It follows from part* (ii) *that*

$$(\bar{x}_{q,k} - \hat{r}_{q,k}) - x^* = \frac{1}{k}\hat{e}_{q,k} + \begin{cases} \mathcal{O}\big(\frac{\log k}{k}\big) & \text{if} \quad q = 1, \\ \mathcal{O}\big(\frac{1}{k}\big) & \text{if} \quad 0 < q < 1. \end{cases}$$

where $\|\hat{e}_{q,k}\| \le \hat{B}\sqrt{\log(1/\delta)}$ *with probability* $1-\delta$ *for a deterministic constant* $\hat{B} = \mathcal{O}(1)$.

Proof The proof techniques of Theorem 3 applies directly except that the Taylor approximation for the gradients of the component functions will have an extra term compared to the proof of Theorem 3 (see also (52)). Also, instead of Lemmas 2 and 4 that apply to only quadratic functions, their extensions Lemmas 6 and 7 given in the appendix are used in the proof. For the sake of completeness, besides these changes, we also give an overview of the main modifications required for each part of the proof:

(i) The expression (39) for the gradient should be modified to include an extra error term η_j of the form

$$\nabla f(x_0^j) = H_*(x_0^j - x^*) + \eta_j, \quad \|\eta_j\| \le \frac{U}{2}\|x_0^j - x^*\|^2. \tag{55}$$

By Lemma 5, $\sum_j \eta_j \le \frac{U}{2}\|\|x_0^j - x^*\|^2 = \mathcal{O}(\alpha_j^2)$ therefore the sequence η_j is summable and if averaged decays like $\mathcal{O}(1/k)$ without degrading the convergence rate except possibly the constants hidden by $\mathcal{O}(\cdot)$.

(ii) The same proof applies by invoking Lemma 7 in lieu of Lemma 4.

(iii) Instead of Lemma 1, we use Lemma 5. The expression (51) on the difference of gradients needs to be adjusted as

$$\|\nabla f_{\sigma_k(i)}(x_{i-1}^k) - \nabla f_{\sigma_k(i)}(x^*) - \nabla^2 f_{\sigma_k(i)}(x^*)(x_{i-1}^k - x^*)\| \le \frac{U}{2}\|x_i^k - x^*\|^2. \tag{56}$$

The right-hand side is still $\mathcal{O}(\alpha_k^2)$ by an application of Lemma 5 therefore the rest of the proof applies.

6 An RR algorithm with bias removal

Part (*iii*) of Theorem 4 (see also part (*iii*) of Theorem 3) shows that if the estimate of the bias term $\hat{r}_{q,k}$ given by (54) is subtracted from the q-suffix averaged RS iterates, then the distance to the optimal solution of the q-suffix averaged iterates becomes on the order of $\mathcal{O}(1/k)$ for $0 < q < 1$ and on the order of $\mathcal{O}(\log k/k)$ for $q = 1$ with high probability. By strong convexity, this translates into a rate of $\tilde{\mathcal{O}}(1/k^2)$ in the suboptimality of the objective values (where $\tilde{\mathcal{O}}$ ignores the logarithmic terms in k appearing when $q = 1$.). We call this "subtraction operation", *bias removal*. Algorithm 1 describes the De-biased Random Reshuffling (DRR) method with bias removal. In a practical implementation, the number of cycles can be fixed in advance to a certain number K, and the estimation of the bias can be done only once at the last (K-th) cycle (see Step (*ii*) of Algorithm 1) and then can be subtracted from the averaged iterates.

```
Algorithm 1 De-biased Random Reshuffling (DRR)
Input: Initial point $x_0^0 \in \mathbb{R}^n$, number of cycles $K \in \mathbb{N}$, suffix averaging parameter $q \in (0, 1]$,
stepsize parameters $R > 0$ and $s \in (1/2, 1)$.
Initialization: $\bar{x}_{1,0} = 0 \in \mathbb{R}^n$, $\hat{v}_0 = 0 \in \mathbb{R}^n$, $\bar{\alpha}_{1,0} = 0 \in \mathbb{R}$, $\hat{H}_0 = 0 \in \mathbb{R}^{n \times n}$.
 1. For each cycle $k = 0, 1, 2, \ldots, K - 1$:
   (a) Inner iteration.
        (i) Pick a permutation $\sigma_k$ of $\{1, \ldots, m\}$ uniformly at random.
       (ii) For $i = 1, 2, \ldots, m$:
                Compute $x_i^k$ by: $x_i^k = x_{i-1}^k - \alpha_k \nabla f_{\sigma_k(i)}(x_{i-1}^k)$,   $\alpha_k = \frac{R}{(k+1)^s}$.
            // Precompute for the bias estimation only for the last cycle
            If $k = K - 1$, compute $\hat{v}_i$ and $\hat{H}_i$ by :
            $\hat{v}_i = \hat{v}_{i-1} + \nabla^2 f_{\sigma_k(i)}(x_{i-1}^k) \nabla f_{\sigma_k(i)}(x_{i-1}^k)/2$,   $\hat{H}_i = \hat{H}_{i-1} + \nabla^2 f_{\sigma_k(i)}(x_{i-1}^k)$
      (iii) Set outer iterate: $x_0^{k+1} = x_m^k$.
   (b) Update the simple average of the iterates and the stepsize:
       $\bar{x}_{1,k+1} = \frac{k}{k+1} \bar{x}_{1,k} + \frac{1}{k+1} x_0^k$,   $\bar{\alpha}_{1,k+1} = \frac{k}{k+1} \bar{\alpha}_{1,k} + \frac{1}{k+1} \alpha_k$
 2. If $q \in (0, 1)$, compute $q$-suffix averages from the simple averages:
       $\bar{x}_{q,K} = \frac{\bar{x}_{1,K} - q \bar{x}_{1,(1-q)K}}{1 - q}$,   $\bar{\alpha}_{q,K} = \frac{\bar{\alpha}_{1,K} - q \bar{\alpha}_{1,(1-q)K}}{1 - q}$.
 3. Estimate the bias by the formula (37) : $\hat{b}_{q,K} = -\bar{\alpha}_{q,K} \hat{H}_m^{-1} \hat{v}_m$ in the last cycle.
Output: $\bar{x}_{q,K} - \hat{b}_{q,K}$.
```

The bias removal of the DRR algorithm requires an $n \times n$ matrix inversion which requires $\approx n^3$ arithmetic operations (if there is more structure on the Hessian of f_i such as low-rankness or sparsity this could be improved to $\approx n^2$), but accelerates the convergence with high-probability. For small or moderate n, this could be done efficiently and incrementally processing the functions one at a time; however for large n this may be impractical or infeasible limiting the applicability of this method. Nevertheless, the expensive matrix inversion step does not need to be done at every cycle, it suffices to do it only once at the end of the last cycle. Figure 2 compares the performance of SGD, RR and DRR methods in terms of the histogram of the distance to the optimal solution (left panel) and suboptimality of the objective function (right panel) on a randomly generated quadratic example with a dense Hessian matrix with parameters $m = 50$, $n = 20$.[8] For a fair comparison, we run all the algorithms with the same amount of CPU time.[9] In particular, in Figure 2 we run DRR for 0.5 seconds including the bias correction step, and run RR and SGD for the same amount of time. For both SGD and RR, we use a decaying stepsize $\alpha_k = R/k^s$ with $s = 0.75$ and report the *averaged* iterates. We tune R to the dataset similar to the standard implementations of SGD methods [10] and set it to $R = \frac{1}{3}10^{-3}$. All the algorithms are initialized to zero. The panel on the top and left-hand side of Figure 2 shows the *histograms* of the distance to minimizer of the last iterate for RR, SGD and De-biased RR methods over 500 sample paths where the *y-axis* denotes the distance to minimizer of the last iterate and the *x-axis* denotes the number of occurences over 500 sample paths. We observe from this histogram that SGD is performing the worst. On the bottom, left panel, we remove SGD from the picture and compare only RR and DRR in terms of the histograms of the distance to minimizer. The red line and the blue lines illustrate the mean values obtained from the histograms of DRR and RR methods respectively. We see that for the DRR, the histogram is shifted slightly to the left; i.e. DRR has smaller error on average. On the right-hand side of Figure 2, we plot the histograms of the suboptimality for RR, DRR and SGD methods instead where the *y-axis* denotes the suboptimality in function values of the last iterate and the *x-axis* denotes the number of occurences. We see a similar behavior. We observe that SGD is consistently performing the worst, whereas DRR has better suboptimality on average (averaging over sample paths). Figure 3 repeats the experiment with a longer time budget of 5 seconds otherwise keeping all the experimental setup the same including the stepsize, averaging parameter, initialization and the objective function. We see a clearer separation between the histograms of the RR method and the DRR method. We see qualitatively similar results when we run the algorithms for different amount of times and for different values of the stepsize decay parameter s. These results show that the asymptotic performance would get better if one removes the bias term and typically we need more cycles for the bias correction term to be effective. The results also illustrate the results of Theorem 3 and 4 on the biasedness of the RR iterations in the sense that asymptotically an improvement can be obtained by subtracting the bias.

Next, we compare RR and DRR methods to another method SAGA [14] which is a de-biased method that improves on the theory behind variance reduction methods such as SAG [39] and SVRG [24]. SAGA method can achieve linear convergence in expected suboptimality when the objective is strongly convex. For

[8] The quadratic functions $f_i(x)$ have the form $f_i(x) = x^T A_i x + q_i^T x + r_i$. The matrices A_i are chosen randomly satisfying $A_i = \frac{1}{n} R_i R_i^T + \lambda I$ where I is the $n \times n$ identity matrix, R is a random matrix with each entry uniform on the interval $[-50, 50]$ and λ is a regularization parameter to make the problem strongly convex. We set $\lambda = 5$. The vectors q_i are random, each component is uniformly distributed on the interval $[-50, 50]$ and c_i is uniform on the interval $[-1, 1]$.

[9] We note that all experiments were performed on a Macbook Pro with an 3.1 GHz Intel Core i7 processor and 16GB of RAM, using Matlab R2017a running on the operating system Mac OS Sierra v10.12.5.

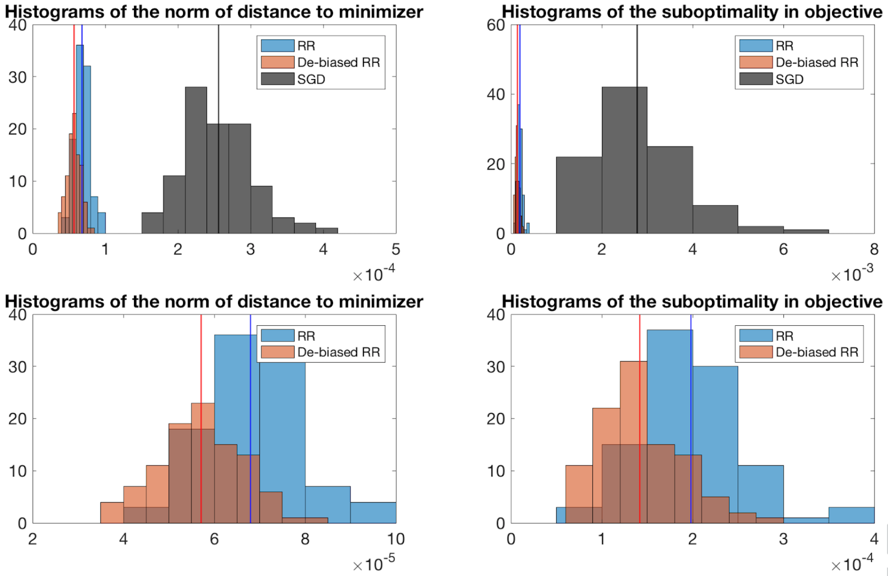


Fig. 2: Comparison of RR, Debiased-RR (DRR) and SGD when component functions are random quadratics with $m = 50$, $n = 20$ and with simulation time 0.5 seconds over 500 sample paths. Top, left: Histograms of dist_k for RR, DRR and SGD. Bottom, left: Histograms of dist_k for RR and DRR only (without SGD). Top, right: Histograms of the suboptimality in objective value for RR, DRR and SGD. Bottom, right: Histograms of the suboptimality in objective value for RR and DRR only (without SGD).

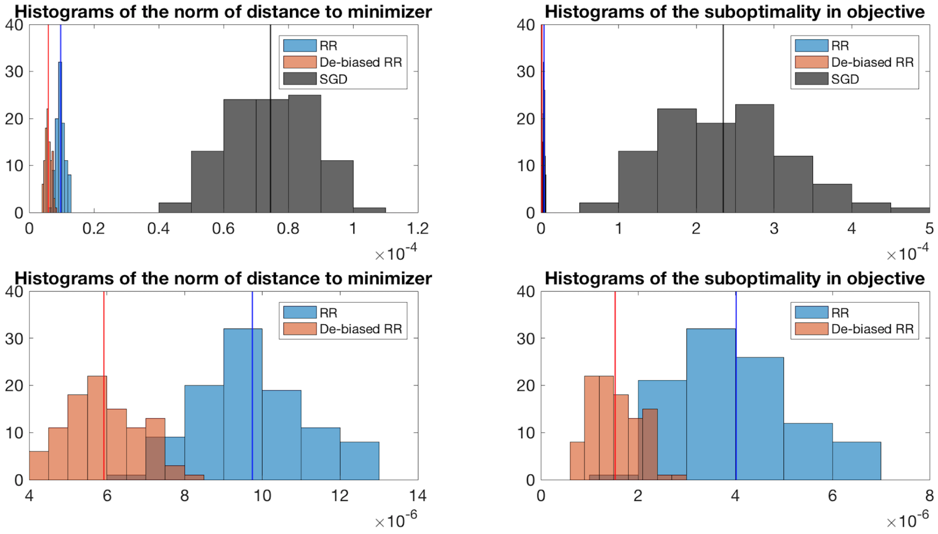
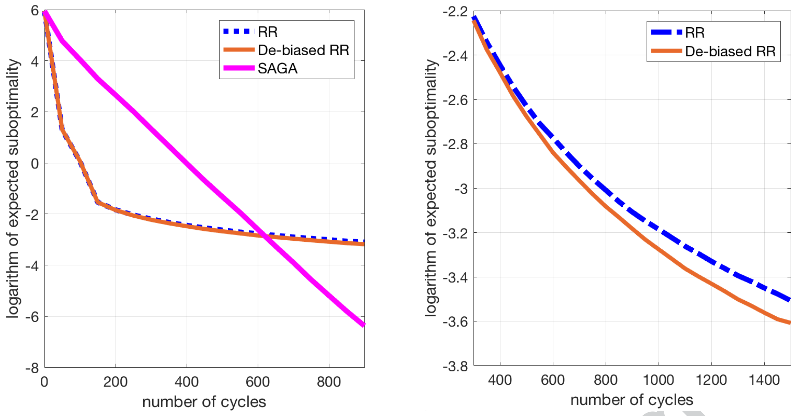


Fig. 3: Comparison of RR, De-biased-RR (DRR) and SGD. The simulation framework and parameters are the same as those in Fig. 2 except that the simulation time is 5 seconds instead for each path.

many structured problems such as logistic regression and linear regression, SAGA can be implemented efficiently requiring $O(n)$ memory [14, 22], however in general, it requires $\mathcal{O}(mn)$ memory to solve the problem (1) (as it stores historical gradients of all the component functions), which is impractical when m is very large [14, 22]. This is in constrast with SGD and RR which requires only $O(n)$ memory to operate, therefore can scale better to large m in general to solve the problem (1). In the left panel of Figure 4, we compare the expected suboptimality



(a) Comparison of RR, DRR and SAGA (b) Comparison of RR and DRR

Fig. 4: Comparison of RR, De-biased RR (DRR) and SAGA methods in terms of performance. The y-axis is the expected suboptimality in function value after k cycles, and x-axis is the number of cycles k. (a): Comparison of RR and DRR only. (b): Comparison of RR, DRR and SAGA methods.

for the DRR, RR and SAGA methods over 500 sample paths with the same experimental setup for RR and DRR methods including the objective, the stepsize choice and the averaging with parameter $s = 0.75$. We first run RR and SAGA methods and plot expected optimality $\mathbb{E}f(x_0^k) - f(x^*)$ versus the number of cycles k. Both RR and SAGA methods have access to the same number of stochastic gradient evaluations for every k which make them directly comparable as the gradient computations determine the running time. We then run the DRR method (including the de-biasing step at the last cycle), giving it the same amount of running time with the other methods for a fair comparison. For the SAGA algorithm we use the recommended stepsize from [14] for strongly convex objectives. The y-axis of Figure 4 is the expected suboptimality in a logarithmic scale whereas the x-axis is the number of cycles. We see on this example that RR and De-biased RR has a fast progress in the beginning compared to SAGA but when the number of cycles grows, SAGA eventually outperforms RR and DRR. If the accuracy desired is not too high (say if being $\varepsilon = 10^{-2}$ of the optimum value $f(x^*)$ is good enough, in this example $f(x^*) \approx -3.9872$), RR and DRR can be good choices; however for higher accuracy requirements (say $\varepsilon = 10^{-4}$ or smaller), SAGA is a better choice. These numerical findings are consistent with the fact that SGD-like algorithms have the fastest progress in the beginning when the iterates are far away from the optimum, and they slow down later with a sublinear convergence rate due to the decaying stepsize needed for guaranteeing convergence [6]. On the right panel of Figure 4, we compare RR and DRR only (removing the SAGA algorithm) to focus on the differences between them; otherwise keeping the experimental setup exactly the same as before. We start seeing a consistent improvement in performance with the DRR method after $k = 300$ cycles and the amount of improvement increases when the number of cycles increases. This is expected as our results regarding the explicit computation of the bias has an asymptotic nature (see part (*iii*) of Theorem 3) and our bias estimation gets more accurate as the number of iterations grows.

7 Conclusion

We analyzed the random reshuffling (RR) method for minimizing a finite sum of convex component functions. When the objective function is strongly convex and the component functions are smooth, averaged RR iterates converge at rate $\sim 1/k^s$ to the optimal solution almost surely (which translates into a rate of $1/k^{2s}$ in the suboptimality of the objective value) for a diminishing stepsize $\alpha_k = \Theta(1/k^s)$ with $s \in (1/2, 1)$. This is faster than SGD's $\Omega(\frac{1}{k})$ rate. Viewing RR as a gradient descent method with random gradient errors, this result builds on first showing that gradient errors E_k satisfying $E_k = \mathcal{O}(\alpha_k)$ and then relating the gradient error sequence to an i.i.d sequence to which martingale theory is applicable. Note that the gradient errors in SGD are larger with a $\mathcal{O}(1)$ variance, which leads to a less accurate gradient descent direction. Beyond RR and SGD comparison, these results also give insight into the fast convergence properties of without-replacement sampling strategies compared to with-replacement sampling strategies.

After characterizing the convergence rate of RR, we look into second-order terms in the asymptotic expansion of the averaged RR iterates and obtain high probability bounds. We use these bounds to develop a new method that can accelerate the convergence rate of RR to $\mathcal{O}(\frac{1}{k^2})$ with high probability. Finally, we show that the $\mathcal{O}(\frac{1}{k^2})$ rate can also be achieved in expectation (which is a weaker notion of convergence with respect to convergence with high probability) for the $s = 1$ case by adjusting the stepsize to the strong convexity constant of the objective properly.

A Proof of Theorem 2

Proof By substituting the gradients of the component functions $\nabla f_i(x) = P_i x - q_i$ into the RR iterations given by (5), we obtain the recursion

$$x_0^{k+1} = \prod_{i=1}^{m}(I_n - \alpha_k P_{\sigma_k(i)})x_0^k + \alpha_k \sum_{i=1}^{m}\prod_{j=i+1}^{m}(I_n - \alpha_k P_{\sigma_k(j)})q_{\sigma_k(i)} \tag{57}$$

$$= \Big(I_n - \alpha_k P + \mathcal{O}(\alpha_k^3)\Big)x_0^k + \alpha_k \sum_{i=1}^{m} q_i - \alpha_k^2 \hat{\mu}_{\sigma_k} + \mathcal{O}(\alpha_k^3), \tag{58}$$

where $P := \sum_{i=1}^m P_i$ and

$$\hat{\mu}_{\sigma_k} := -\sum_{1\le i<j\le m} P_{\sigma_k(j)} \nabla f_{\sigma_k(i)}(x_0^k). \tag{59}$$

Since the component functions are quadratics, the optimal solution can be computed explicitly and is given by $x^* = P^{-1}\sum_{i=1}^m q_i$. Then, it follows after a straightforward computation that (58) is equivalent to

$$x_0^{k+1} - x^* = \big(I - \alpha_k P + \mathcal{O}(\alpha_k^3)\big)(x_0^k - x^*) - \alpha_k^2 \hat{\mu}_{\sigma_k} + \mathcal{O}(\alpha_k^3). \tag{60}$$

We also have

$$\begin{aligned} \|\mu_{\sigma_k} - \hat{\mu}_{\sigma_k}\| &\leq \sum_{1\leq i<j\leq m} \|P_{\sigma_k(j)}\| \|\nabla f_{\sigma_k(i)}(x_0^k) - \nabla f_{\sigma_k(i)}(x^*)\| \\ &\leq \sum_{1\leq i<j\leq m} L_{\sigma_k(j)} L_{\sigma_k(i)} \mathrm{dist}_k = \mathcal{O}(\mathrm{dist}_k), \end{aligned}$$

where μ_{σ_k} is defined by (23) with $\sigma = \sigma_k$. Plugging this into (60),

$$x_0^{k+1} - x^* \;=\; \big(I - \alpha_k P + \mathcal{O}(\alpha_k^2) + \mathcal{O}(\alpha_k^3)\big)(x_0^k - x^*) - \alpha_k^2 \mu_{\sigma_k} + \mathcal{O}(\alpha_k^3 + \alpha_k^2 \mathrm{dist}_k).$$

Taking norm squares of both sides, taking conditional expectations and using the fact that μ_{σ_k} is bounded (see (26)), we obtain

$$\begin{aligned} \mathbb{E}_{\sigma_k}\left(\mathrm{dist}_{k+1}^2 \,\middle|\, x_0^k\right) \;=\;& (x_0^k - x^*)^T\big(I - 2\alpha_k P + \mathcal{O}(\alpha_k^2)\big)(x_0^k - x^*) + 2\alpha_k^2 \langle x_0^k - x^*, -\bar{\mu}\rangle \\ &+ \mathcal{O}(\alpha_k^3 \mathrm{dist}_k + \alpha_k^2 \mathrm{dist}_k^2 + \alpha_k^4), \end{aligned} \tag{61}$$

where $\mathbb{E}_{\sigma_k}$ denotes the expectation with respect to the random permutation σ_k and

$$\bar{\mu} = \mathbb{E}_{\sigma_k}\left(\mu_{\sigma_k}\right) = \mathbb{E}_{\sigma_1}\left(\mu_{\sigma_1}\right).$$

It follows from Cauchy-Schwartz that for any $\beta > 0$,

$$\alpha_k^2 \left\| \langle x_0^k - x^*, -\bar{\mu}\rangle \right\| \leq \alpha_k^2 \mathrm{dist}_k \|\bar{\mu}\| = \left(\sqrt{\beta}\alpha_k^{1/2}\mathrm{dist}_k\right) \frac{\alpha_k^{3/2}\|\bar{\mu}\|}{\sqrt{\beta}} \leq \frac{\beta\alpha_k \mathrm{dist}_k^2}{2} + \frac{\alpha_k^3\|\bar{\mu}\|^2}{2\beta},$$

and also

$$\alpha_k^3 \mathrm{dist}_k = \alpha_k^2\left(\alpha_k \mathrm{dist}_k\right) \leq \frac{\alpha_k^4}{2} + \frac{\alpha_k^2 \mathrm{dist}_k^2}{2}.$$

Plugging these bounds back into (61), using the lower bound (6) on the Hessian $H_* = P$ and invoking the tower property of the expectations:

$$\mathbb{E}\left(\mathrm{dist}_{k+1}^2\right) \;=\; \big(1 - \alpha_k(2c - \beta) + \mathcal{O}(\alpha_k^2)\big)\mathbb{E}\left(\mathrm{dist}_k^2\right) + \alpha_k^3\frac{\|\bar{\mu}\|^2}{\beta} + \mathcal{O}(\alpha_k^4).$$

Plugging in $\alpha_k = R/k^s$, it follows from Chung's lemma [16, Lemma 4.2] that,

$$\mathbb{E}\left(\mathrm{dist}_{k+1}^2\right) \leq \begin{cases} \frac{R^2\|\bar{\mu}\|^2}{\beta(2c-\beta)}\frac{1}{k^{2s}} + o\left(\frac{1}{k^{2s}}\right) & \text{if} \quad 0 < s < 1 \text{ and } 2c - \beta > 0, \\ \frac{R^3\|\bar{\mu}\|^2}{\beta(R(2c-\beta)-2)}\frac{1}{k^2} + o(\frac{1}{k^2}) & \text{if} \quad s = 1 \text{ and } R(2c-\beta) - 2 > 0. \end{cases} \tag{62}$$

Next we choose β to get the best upper bound above. This is done by choosing $\beta = c$ for $0 < s < 1$ and choosing $\beta = (Rc - 1)/R$ for $s = 1$ which yields

$$\mathbb{E}\left(\mathrm{dist}_{k+1}^2\right) \leq \begin{cases} \frac{R^2\|\bar{\mu}\|^2}{c^2}\frac{1}{k^{2s}} + o\left(\frac{1}{k^{2s}}\right) & \text{if} \quad 0 < s < 1, \\ \frac{R^4\|\bar{\mu}\|^2}{(Rc-1)^2}\frac{1}{k^2} + o(\frac{1}{k^2}) & \text{if} \quad s = 1 \text{ and } Rc - 1 > 0. \end{cases} \tag{63}$$

By Jensen's inequality, we have $\mathbb{E}(\mathrm{dist}_k) \leq \left(\mathbb{E}\left(\mathrm{dist}_{k+1}^2\right)\right)^{1/2}$. Therefore, by taking square roots of both sides above in (62) we conclude.

B Technical lemmas for the proof of Theorem 3

The first lemma is on characterizing what is the worst-case distance of the all the inner iterates of RR to the optimal solution x^*. This quantity we want to upper bound is a random variable, but the upper bounds we obtain are deterministic holding for every sample path. This lemma is based on Corollary 1 and uses the fact that the distance between the inner iterates are on the order of the stepsize.

Lemma 1 *Under the conditions of Theorem 3 we have* $\max\limits_{0\le i<m} \|x_i^k - x^*\| = \mathcal{O}(\frac{1}{k^s})$ *where* $\mathcal{O}(\cdot)$ *hides a constant that depends only on* G_*, L, m, c *and* R.

Proof By Corollary 1,

$$\|x_0^k - x^*\| = \mathcal{O}(\frac{1}{k^s}), \tag{64}$$

where $\mathcal{O}(\cdot)$ hides a constant that depends only on G_*, L, m, R and c. We have also for any $0 \le i < m$ and $k \ge 0$,

$$\begin{aligned}\|x_i^k - x^*\| &\le \|x_0^k - x^*\| + \|x_i^k - x_0^k\| = \|x_0^k - x^*\| + i\alpha_k \max_{\ell=1,\dots,i} \|\nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k)\| \\ &\le \|x_0^k - x^*\| + (m-1)\frac{R}{(k+1)^s}\big(G_* + \max_{\ell=1,\dots,i}\|\nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k) - \nabla f_{\sigma_k(\ell)}(x^*)\|\big) \\ &\le \|x_0^k - x^*\| + (m-1)\frac{R}{(k+1)^s}\big(G_* + L\max_{\ell=1,\dots,i}\|x_{\ell-1}^k - x^*\|\big),\end{aligned}$$

where we used the L-Lipschitzness of the gradient of f where L is given by 19. Using (64) and applying this inequality inductively for $i = 0, 1, 2, \dots, m-1$ we conclude.

The second lemma is on characterizing how fast on average the outer iterates move (if normalized by the stepsize) after a cycle of the RR algorithm. This is clearly related to the magnitude of the gradients seen by the iterates and is fundamental for establishing the convergence rate of the averaged RR iterates in Theorem 3.

Lemma 2 *Under the conditions of Theorem 3, consider the sequence*

$$I_{q,k} = \frac{\sum_{j=(1-q)k}^{k-1}(x_0^j - x_0^{j+1})\alpha_j^{-1}}{qk}, \quad 0 < q \le 1. \tag{65}$$

Then,

$$I_{q,k} = \begin{cases}\mathcal{O}\big(\frac{\log k}{k}\big) & \textit{if} \quad q = 1, \\ \mathcal{O}\big(\frac{1}{k}\big) & \textit{if} \quad 0 < q < 1.\end{cases}$$

In the former case, $\mathcal{O}(\cdot)$ *hides a constant that depends only on* G_*, L, m, c, R, s, q *and* $dist_0$. *In the latter case, the same dependency on the constants occurs except that the dependency on* $dist_0$ *can be removed.*

Proof It follows from integration by parts that for any $\ell < k$,

$$-\sum_{j=\ell}^{k-1}(x_0^j - x_0^{j+1})\alpha_j^{-1} = \alpha_k^{-1}(x_0^k - x^*) - \alpha_\ell^{-1}(x_0^\ell - x^*) - \sum_{j=\ell}^{k-1}(x_0^{j+1} - x^*)(\alpha_{j+1}^{-1} - \alpha_j^{-1}). \tag{66}$$

Next, we investigate the asymptotic behavior of the terms on the right-hand side. A consequence of Corollary 1 and the inequality 25 is that

$$\alpha_k^{-1}\|x_0^k - x^*\| = \frac{(k+1)^s}{R}\|x_0^k - x^*\| \le \frac{LmG_*}{c} + o(1) = \mathcal{O}(1), \tag{67}$$

and therefore

$$\begin{aligned}|\alpha_{k+1}^{-1} - \alpha_k^{-1}|\|x_0^k - x^*\| &= \frac{(k+2)^s - (k+1)^s}{(k+1)^s}\alpha_k^{-1}\|x_0^k - x^*\| = \Big(\big(1 + \frac{1}{k+1}\big)^s - 1\Big)\alpha_k^{-1}\|x_0^k - x^*\| \\ &\le \frac{s}{k+1}\alpha_k^{-1}\|x_0^k - x^*\| \le \frac{sLmG_*}{c}\frac{1}{k+1} + o(\frac{1}{k+1}) = \mathcal{O}(\frac{1}{k+1}),\end{aligned}$$

where $\mathcal{O}(\cdot)$ hides a constant that depends only on L, G_*, c, m and s. Then, setting $\ell = (1-q)k$ in (66), it follows that

$$\begin{aligned}
\Big\|\sum_{j=\ell}^{k-1}(x_0^j - x_0^{j+1})\alpha_j^{-1}\Big\| &\le \|\alpha_k^{-1}(x_0^k - x^*)\| + \|\alpha_{(1-q)k}^{-1}(x_0^{(1-q)k} - x^*)\| \qquad (68)\\
&\quad + \sum_{j=(1-q)k}^{k-1} \|x_0^{j+1} - x^*\||\alpha_{j+1}^{-1} - \alpha_j^{-1}|.\\
&= \mathcal{O}(1) + \|\alpha_{(1-q)k}^{-1}(x_0^{(1-q)k} - x^*)\| + \mathcal{O}\Big(\sum_{j=(1-q)k}^{k-1}\frac{1}{j+1}\Big). \qquad (69)
\end{aligned}$$

We also have

$$\|\alpha_{(1-q)k}^{-1}(x_0^{(1-q)k} - x^*)\| = \begin{cases}\alpha_0^{-1}\mathrm{dist}_0 & \text{if} \quad q = 1,\\ \mathcal{O}(1) & \text{if} \quad 0 < q < 1,\end{cases} \qquad (70)$$

where the second part follows from (67) with similar constants for the $\mathcal{O}(\cdot)$ term. As the sequence $\frac{1}{j+1}$ is monotonically decreasing, for any $k > 0$ we have the bounds

$$\sum_{j=(1-q)k}^{k-1}\frac{1}{j+1} \le \frac{1}{(1-q)k+1} + \int_{(1-q)k}^{k-1}\frac{1}{x+1}dx \le \begin{cases}1 + \log k & \text{if} \quad q = 1,\\ 1 + \log(\frac{1}{1-q}) & \text{if} \quad 0 < q < 1.\end{cases} \qquad (71)$$

Note that when $q = 1$ this bound grows with k logarithmically whereas for $q < 1$ it does not grow with k. Then, combining (69), (70) and (71) we obtain

$$\|I_{q,k}\| \le \frac{\|\sum_{j=\ell}^{k-1}(x_0^j - x_0^{j+1})\alpha_j^{-1}\|}{qk} = \begin{cases}\mathcal{O}(\frac{\log k}{k}) & \text{if} \quad q = 1,\\ \mathcal{O}(\frac{1}{k}) & \text{if} \quad 0 < q < 1,\end{cases}$$

as desired which completes the proof.

Lemma 3 *Let σ be a random permutation of $\{1, 2, \ldots, m\}$ sampled uniformly over the set of all permutations Γ defined by* (4) *and $\mu(\sigma)$ be the vector defined by* (23) *that depends on σ. Then,*

$$\bar{\mu} = \mathbb{E}_\sigma\big(\mu(\sigma)\big) = \frac{1}{2}\sum_{i=1}^m P_i\nabla f_i(x^*), \qquad (72)$$

where $\mathbb{E}_\sigma$ denotes the expectation with respect to the random permutation σ and $\bar{\mu}$ is defined by (29).

Proof For any $i \neq \ell$, the joint distribution of $(\sigma(i), \sigma(\ell))$ is uniform over the set of all (ordered) pairs from $\{1, 2, \ldots, m\}$. Therefore, for any $i \neq \ell$,

$$\begin{aligned}
\mathbb{E}_\sigma\big[P_{\sigma(i)}\nabla f_{\sigma(\ell)}(x^*)\big] &= \sum_{i=1}^m \sum_{i\neq j, j=1}^m \frac{P_i\nabla f_j(x^*)}{m(m-1)}\\
&= \frac{\sum_{i=1}^m P_i\sum_{j=1}^m \nabla f_j(x^*) - \sum_{j=1}^m P_j\nabla f_j(x^*)}{m(m-1)} = -\frac{\sum_{j=1}^m P_j\nabla f_j(x^*)}{m(m-1)},
\end{aligned}$$

where we used the fact that $\nabla f(x^*) = \sum_{j=1}^m \nabla f_j(x^*) = 0$ by the first order optimality condition. Then, by taking the expectation of (74), we obtain

$$\mathbb{E}_\sigma(\mu(\sigma)) = -\sum_{i=1}^m\sum_{\ell=0}^{i-1}\mathbb{E}\big[P_{\sigma(i)}\nabla f_{\sigma(\ell)}(x^*)\big] = \sum_{i=1}^m\sum_{\ell=0}^{i-1}\frac{\sum_{j=1}^m P_j\nabla f_j(x^*)}{m(m-1)} = \frac{\sum_{j=1}^m P_j\nabla f_j(x^*)}{2},$$

which completes the proof.

Lemma 4 *Under the conditions of Theorem 3, the following statements are true:*

(i) *We have*

$$E_k = \alpha_k \mu(\sigma_k) + \mathcal{O}(\alpha_k^2), \quad k \geq 0, \tag{73}$$

where E_k *is the gradient error defined by* (8), $\mathcal{O}(\cdot)$ *hides a constant that depends only on* G_*, L, m, R *and* c *and*

$$\mu(\sigma_k) = -\sum_{i=1}^{m} P_{\sigma_k(i)} \sum_{\ell=1}^{i-1} \nabla f_{\sigma_k(\ell)}(x^*) \tag{74}$$

is a sequence of i.i.d. variables where the function $\mu(\cdot)$ *is defined by* (23).

(ii) *For any* $0 < q \leq 1$, $\lim_{k\to\infty} Y_{q,k} = \bar{\mu}$ *a.s. where* $Y_{q,k} = \frac{\sum_{i=(1-q)k}^{k-1} E_j}{\sum_{j=(1-q)k}^{k-1} \alpha_j}$.

(iii) *It holds that*

$$\|\mu(\sigma_k)\| \leq LmG_*. \tag{75}$$

Proof (i) As component functions are quadratics, (8) becomes

$$E_k = \sum_{i=1}^{m} P_{\sigma_k(i)}(x_{i-1}^k - x_0^k) = -\sum_{i=1}^{m} P_{\sigma_k(i)} \alpha_k \sum_{\ell=1}^{i-1} \nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k),$$

where we can substitute

$$\nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k) = \nabla f_{\sigma_k(\ell)}(x^*) + P_{\sigma_k(\ell)}(x_{\ell-1}^k - x^*). \tag{76}$$

Then an application of Lemma 1 proves directly the desired result.

(ii) We introduce the normalized gradient error sequence $Y_j = E_j/\alpha_j$. By part (i), $Y_j = \mu(\sigma_j) + \mathcal{O}(\alpha_j)$ where $\mu(\sigma_j)$ is a sequence of i.i.d. variables. By the strong law of large numbers, we have

$$\lim_{k\to\infty} \frac{\sum_{j=0}^{k-1} \mu(\sigma_j)}{k} = \mathbb{E}\mu(\sigma_j) = \bar{\mu} \quad \text{a.s.}, \tag{77}$$

where the last equality is by the definition of $\bar{\mu}$. Therefore,

$$\lim_{k\to\infty} \frac{\sum_{j=0}^{k-1} Y_j}{k} = \lim_{k\to\infty} \left(\frac{\sum_{j=0}^{k-1} \mu(\sigma_j)}{k} + \frac{\sum_{j=0}^{k-1} \mathcal{O}(\alpha_j)}{k} \right) = \bar{\mu} \quad \text{a.s.},$$

where we used the fact that the second term is negligible as $\sum_{j=0}^{k-1} \alpha_j / k = \mathcal{O}(k^{-s}) \to 0$. As the average of the sequence Y_j converges almost surely, one can show that this implies almost sure convergence of a weighted average of the sequence Y_j as well as long as weights satisfy certain conditions as $k \to \infty$. In particular, as the sequence $\{\alpha_j\}$ is monotonically decreasing and is non-summable, by [15, Theorem 1],

$$\lim_{k\to\infty} Y_{1,k} = \lim_{k\to\infty} \frac{\sum_{j=0}^{k-1} \alpha_j Y_j}{\sum_{j=0}^{k-1} \alpha_j} = \lim_{k\to\infty} \frac{\sum_{j=0}^{k-1} E_j}{\sum_{j=0}^{k-1} \alpha_j} = \bar{\mu} \quad \text{a.s.} \tag{78}$$

This completes the proof for $q = 1$. For $0 < q < 1$, by the definition of $Y_{q,k}$, we can write $Y_{1,k} = (1 - w_k) Y_{q,k} + w_k Y_{1,(1-q)k}$ where the non-negative weights w_k satisfy

$$w_k = \frac{\sum_{j=0}^{(1-q)k-1} \alpha_j}{\sum_{j=0}^{k-1} \alpha_j} \to_{k\to\infty} (1-q)^{1-s} < 1.$$

As both $Y_{1,k}$ and $Y_{1,(1-q)k}$ go to $\bar{\mu}$ a.s. by (78), it follows that

$$\lim_{k\to\infty} Y_{q,k} = \lim_{k\to\infty} \frac{Y_{1,k} - w_k Y_{1,(1-q)k}}{1 - w_k} = \bar{\mu} \quad \text{a.s}$$

as well for any $0 < q < 1$. This completes the proof.

(iii) This is a direct consequence of the triangle inequality applied to the definition (74) with $L_i = \|P_i\|$ and $L = \sum_{i=1}^{m} L_i$.

C Techical Lemmas for the proof of Theorem 4

We first state a result which follows from adapting existing results from the literature to our setting. It extends Corollary 1 from quadratics to smooth functions.

Corollary 2 *Under the setting of Theorem 4, we have*

$$dist_k \leq \frac{RM}{c}\frac{1}{k^s} + o(\frac{1}{k^s}),$$

where the right-hand side is a deterministic sequence, $M := LmG_*$ and G_* is defined by (24).

Proof The result [21, Theorem 3.2] on the asymptotic convergence of incremental gradient implies that all the iterates converge to the optimum, i.e. $x_i^k \to x^*$ for every i fixed as k goes to infinity. Let $\mathcal{X}_\varepsilon$ be the closed ε-ball around the optimum, i.e. $\mathcal{X}_\varepsilon := \{x \in \mathbb{R}^n \ : \ \|x - x^*\| \leq \varepsilon\}$. Clearly, the iterates will be contained in this ball when k is large enough, i.e. for every $\varepsilon > 0$ there exists k_0 (that may depend on ε) such that $x_i^k \in \mathcal{X}$ for any $k \geq k_0$ and for all $i = 1, 2, \ldots, m$. By [21, Theorem 3.2], we have also

$$\limsup_{k\to\infty} k^s \text{dist}_k \leq \frac{RM_\varepsilon}{c}, \tag{79}$$

where $M_\varepsilon := LmG_\varepsilon$ and $G_\varepsilon := \max_{1\leq i\leq m} \sup_{x\in\mathcal{X}_\varepsilon} \|\nabla f_i(x)\|$ is the largest norm of the gradients of the component functions on the compact set $\mathcal{X}_\varepsilon$. If we let ε go to zero, we can replace G_ε with $G_* = \max_{1\leq i\leq m} \|\nabla f_i(x^*)\|$ and M_ε with M in (79). This completes the proof.

Building on this corollary, we obtain the following results.

Lemma 5 *Under the conditions of Theorem 4, all the conclusions of Lemma 1 remain valid.*

Proof The proof of Lemma 1 applies identically except that instead of Corollary 1 we use its extension Corollary 2.

Lemma 6 *Under the conditions of Theorem 4, all the conclusions of Lemma 2 remain valid.*

Proof The proof of Lemma 2 applies identically with the only difference that the bound on $\text{dist}_k = \|x_0^k - x^*\|$ is obtained from Corollary 2 instead of Corollary 1.

Lemma 7 *Under the conditions of Theorem 4, the following statements are true:*

(*i*) *We have*

$$E_k = \alpha_k v(\sigma_k) + \mathcal{O}(\alpha_k^2), \quad k \geq 0, \tag{80}$$

where $\mathcal{O}(\cdot)$ hides a constant that depends only on G_*, L, m, R, c and U and

$$v(\sigma_k) = -\sum_{i=0}^{m-1} \nabla^2 f_{\sigma_k(i)}(x^*) \sum_{\ell=0}^{i-1} \nabla f_{\sigma_k(\ell)}(x^*).$$

(*ii*) *It holds that*

$$\|v(\sigma_k)\| \leq LmG_*, \tag{81}$$

where

$$\bar{v} := \mathbb{E}v(\sigma_k) = \sum_{i=1}^{m} \nabla^2 f_i(x^*)\nabla f_i(x^*)/2. \tag{82}$$

(*iii*) *For any $0 < q \leq 1$, $\lim_{k\to\infty} Y_{q,k} = \bar{v}$ with probability one where*

$$Y_{q,k} = \frac{\sum_{i=(1-q)k}^{k-1} E_j}{\sum_{j=(1-q)k}^{k-1} \alpha_j}. \tag{83}$$

Proof For part (i), first we express E_k using the Taylor expansion and the Hessian Lipschitzness as

$$\begin{aligned} E_k &= \sum_{i=1}^{m}\Big(\nabla^2 f_{\sigma_k(i)}(x_0^k)\Big)(x_{i-1}^k - x_0^k) + \mathcal{O}(U\|x_{i-1}^k - x_0^k\|^2) \\ &= -\sum_{i=1}^{m}\Big(\nabla^2 f_{\sigma_k(i)}(x_0^k)\Big)(x_{i-1}^k - x_0^k) + \mathcal{O}\Big(\alpha_k^2 U\Big\|\sum_{\ell=1}^{i-1}\nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k)\Big\|\Big). \end{aligned}$$

By Lemma 5, we have $\|x_\ell^k - x^*\| = \mathcal{O}(\alpha^k)$ with probability one. Then, by the gradient and Hessian Lipschitzness we can substitute above

$$\nabla f_{\sigma_k(\ell)}(x_{\ell-1}^k) = \nabla f_{\sigma_k(\ell)}(x^*) + \mathcal{O}(\alpha^k), \quad \nabla^2 f_{\sigma_k(\ell)}(x_{\ell-1}^k) = \nabla^2 f_{\sigma_k(\ell)}(x^*) + \mathcal{O}(\alpha^k),$$

which implies directly Equation (80). The rest of the proof for parts (ii) and (iii) is similar to the proof of Lemma 4 and is omitted.

References

1. A. Agarwal, P.L. Bartlett, P. Ravikumar, and M.J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, May 2012.
2. D. Bertsekas. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization*, 6(3):807–822, 1996.
3. D. Bertsekas. A hybrid incremental gradient method for least squares. *SIAM Journal on Optimization*, 7:913–926, 1997.
4. D. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, United States, 1999.
5. D. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. *Optimization for Machine Learning*, 2010:1–38, 2011.
6. D. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, Belmont, MA, United States, 2015.
7. L. Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. In *Proceedings of the symposium on learning and data science, Paris*, 2009.
8. L. Bottou. Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier and G. Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186. Physica-Verlag HD, 2010.
9. L. Bottou. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
10. L. Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, GenevièveB. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer Berlin Heidelberg, 2012.
11. L. Bottou and Y. Le Cun. On-line learning for very large data sets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151, 2005.
12. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
13. K. L. Chung. On a stochastic approximation method. *Annals of Mathematical Statistics*, 25(3):463–483, September 1954.
14. Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.
15. N. Etemadi. Convergence of weighted averages of random variables revisited. *Proceedings of the American Mathematical Society*, 134(9):2739–2744, 2006.
16. V. Fabian. Stochastic approximation of minima with improved asymptotic speed. *The Annals of Mathematical Statistics*, 38(1):191–200, 1967.
17. V. Fabian. On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, pages 1327–1332, 1968.

18. 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.
19. N.I.M. Gould and S. Leyffer. An introduction to algorithms for nonlinear optimization. In J.F. Blowey, A.W. Craig, and T. Shardlow, editors, *Frontiers in Numerical Analysis*, Universitext, pages 109–197. Springer Berlin Heidelberg, 2003.
20. Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv e-prints*, page arXiv:1706.02677, Jun 2017.
21. M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo. Convergence rate of incremental gradient and Newton methods. *arXiv preprint arXiv:1510.08562*, October 2015.
22. Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečnỳ, and Scott Sallinen. Stopwasting my gradients: Practical svrg. In *Advances in Neural Information Processing Systems*, pages 2251–2259, 2015.
23. 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.
24. Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.
25. H. J. Kushner and G. Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.
26. E. Moulines and F. R. Bach. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. *Advances in Neural Information Processing*, pages 451–459, 2011.
27. A. Nedić and A. Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, pages 4711–4716, 2007.
28. A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
29. A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4):1574–1609, January 2009.
30. A. Nemirovskii, D.B. Yudin, and E.R. Dawson. Problem complexity and method efficiency in optimization. 1983.
31. B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.
32. A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 449–456, 2012.
33. S.S. Ram, A. Nedic, and V.V. Veeravalli. Stochastic incremental gradient descent for estimation in sensor networks. In *Signals, Systems and Computers, ACSSC 2007.*, pages 582–586, 2007.
34. B. Recht and C. Ré. Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. *JMLR Workshop and Conference Proceedings*, 23:11.1–11.24, 2012.
35. B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.
36. B. Recht, C. Ré, S. Wright, and Feng N. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 693–701. Curran Associates, Inc., 2011.
37. H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
38. N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.
39. Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems*, pages 2663–2671, 2012.

40. O. Shamir. Open Problem: Is Averaging Needed for Strongly Convex Stochastic Gradient Descent? *COLT*, 2012.
41. J. Sohl-Dickstein, B. Poole, and S. Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In T. Jebara and E. P. Xing, editors, *ICML*, pages 604–612. JMLR Workshop and Conference Proceedings, 2014.
42. E.R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, P. Xinghao, J. Gonzalez, M.J. Franklin, M.I Jordan, and T. Kraska. MLI: An API for distributed machine learning. In *IEEE 13th International Conference on Data Mining (ICDM)*, pages 1187–1192, 2013.
43. T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML, pages 116–, New York, NY, USA, 2004. ACM.
44. T. Zhang. A note on the non-commutative arithmetic-geometric mean inequality. *arXiv preprint arXiv:1411.5058*, November 2014.