OPT2019: 11th Annual Workshop on Optimization for Machine Learning

How Good is SGD with Random Shuffling?

Itay Safran ITAY.SAFRAN@WEIZMANN.AC.IL
Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science, Israel

Abstract

We study the performance of stochastic gradient descent (SGD) on smooth and strongly-convex finite-sum optimization problems. In contrast to the majority of existing theoretical works, which assume that individual functions are sampled with replacement, we focus here on popular but poorly-understood heuristics, which involve going over random permutations of the individual functions. This setting has been investigated in several recent works, but the optimal error rates remains unclear. In this paper, we provide lower bounds on the expected optimization error with these heuristics (using SGD with any constant step size), which elucidate their advantages and disadvantages. In particular, we prove that after k passes over n individual functions, if the functions are re-shuffled after every pass, the best possible optimization error for SGD is at least $\Omega\left(1/(nk)^2 + 1/nk^3\right)$, which partially corresponds to recently derived upper bounds, and we conjecture to be tight. Moreover, if the functions are only shuffled once, then the lower bound increases to $\Omega(1/nk^2)$. Since there are strictly smaller upper bounds for random reshuffling, this proves an inherent performance gap between SGD with single shuffling and repeated shuffling. As a more minor contribution, we also provide a non-asymptotic $\Omega(1/k^2)$ lower bound (independent of n) for cyclic gradient descent, where no random shuffling takes place.

1. Main

We consider variants of stochastic gradient descent (SGD) for solving unconstrained finite-sum problems of the form

$$\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) \;=\; \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})\;, \tag{1}$$

where $\mathcal{X}$ is some Euclidean space $\mathbb{R}^d$ (or more generally some real Hilbert space), F is a strongly convex function, and each individual function f_i is smooth (with Lipschitz gradients) and Lipschitz on a bounded domain. Such problems are extremely common in machine learning applications, which often boil down to minimizing the average loss over n data points with respect to a class of predictors parameterized by a vector $\mathbf{x}$. When n is large, perhaps the most common approach to solve such problems is via stochastic gradient descent, which initializes at some point in $\mathcal{X}$ and involves iterations of the form $\mathbf{x}' := \mathbf{x} - \eta\nabla f_i(\mathbf{x})$ where $i \in \{1,\ldots,n\}$. The majority of existing theoretical works assume that each i is sampled independently across iterations (also known as with-replacement sampling). For example, if it is chosen independently and uniformly at random from $\{1,\ldots,n\}$, then $\mathbb{E}[\nabla f_i(\mathbf{x})|\mathbf{x}] = \nabla F(\mathbf{x})$, so the algorithm can be seen as a noisy version of exact gradient descent on F (with iterations of the form $\mathbf{x}' := \mathbf{x} - \eta\nabla F(\mathbf{x})$), which greatly facilitates its analysis.

However, this straightforward sampling approach suffers from practical drawbacks, such as requiring truly random data access and hence longer runtime. In practice, it is quite common to use *without-replacement* sampling heuristics, which utilize the individual functions in some random or even deterministic order (see for example [1–4, 9, 13, 14]). Moreover, to get sufficiently high accuracy, it is common to perform several passes over the data, where each pass either uses the same order as the previous one, or some new random order. The different algorithm variants we study in this paper are presented as Algorithms 1 to 4 below. We assume that all algorithms take as input the functions $f_1, \ldots, f_n$, a step size parameter $\eta > 0$ (which remains constant throughout the iterations), and an initialization point $\mathbf{x}_0$. The algorithms then perform k passes (which we will also refer to as epochs) over the individual functions, but differ in their sampling strategies:

- Algorithm 1 (SGD with random reshuffling) chooses a new permutation of the functions at the beginning of every epoch, and processes the individual functions in that order.
- Algorithm 2 (SGD with single shuffling) uses the same random permutation for all k epochs.
- Algorithm 3 (Cyclic SGD) performs k passes over the individual functions, each in the same fixed order (which we will assume without loss of generality to be the canonical order $f_1, \ldots, f_n$)

In contrast, Algorithm 4 presents SGD using with-replacement sampling, where each iteration an individual function is chosen uniformly and independently.

Algorithm 1 SGD with Random Reshuffling

```
$\mathbf{x} := \mathbf{x}_0$
for $t = 1, \ldots, k$ do
    Sample a permutation $\sigma(1), \ldots, \sigma(n)$ of $\{1, \ldots, n\}$ uniformly at random
    for $j = 1, \ldots, n$ do
        $\mathbf{x} := \mathbf{x} - \eta \nabla f_{\sigma(j)}(\mathbf{x})$
    end for
    $\mathbf{x}_t := \mathbf{x}$
end for
```

Algorithm 2 SGD with Single Shuffling

```
$\mathbf{x} := \mathbf{x}_0$
Sample a permutation $\sigma(1), \ldots, \sigma(n)$ of $\{1, \ldots, n\}$ uniformly at random
for $t = 1, \ldots, k$ do
    for $j = 1, \ldots, n$ do
        $\mathbf{x} := \mathbf{x} - \eta \nabla f_{\sigma(j)}(\mathbf{x})$
    end for
    $\mathbf{x}_t := \mathbf{x}$
end for
```

Algorithm 3 Cyclic SGD

```
$\mathbf{x} := \mathbf{x}_0$
for $t = 1, \ldots, k$ do
    for $j = 1, \ldots, n$ do
        $\mathbf{x} := \mathbf{x} - \eta \nabla f_j(\mathbf{x})$
    end for
    $\mathbf{x}_t := \mathbf{x}$
end for
```

Algorithm 4 SGD with Replacement

```
$\mathbf{x} := \mathbf{x}_0$
for $t = 1, \ldots, k$ do
    for $j = 1, \ldots, n$ do
        Sample $i \in \{1, \ldots, n\}$ uniformly
        $\mathbf{x} := \mathbf{x} - \eta \nabla f_i(\mathbf{x})$
    end for
    $\mathbf{x}_t := \mathbf{x}$
end for
```

These without-replacement sampling heuristics are often easier and faster to implement in practice. In addition, when using random permutations, they often exhibit faster error decay than with-replacement SGD. A common intuitive explanation for this phenomenon is that random permutations force the algorithm to touch each individual function exactly once during each epoch, whereas with-replacement makes the algorithm touch each function once only in expectation. However, theoretically analyzing these sampling heuristics has proven to be very challenging, since the individual iterations are no longer statistically independent.

	Random Reshuffling	Single Shuffling	Cyclic	With Replacement
Upper Bound	$1/k^2$ [6] $1/n$ (for $k=1$) [15] $1/(nk)^2 + 1/k^3$ [7] $1/nk^2$ [8]	$1/k^2$ [5] $1/n$ (for $k=1$) [15]	$1/k^2$ [5]	$1/nk$
Lower Bound	$1/n$ (for $k=1$) [7] $\boldsymbol{1/(nk)^2 + 1/nk^3}$	$\boldsymbol{1/nk^2}$	$1/k^2$ ([5], asymptotic) $\boldsymbol{1/k^2}$ (non-asymptotic)	$1/nk$

Table 1: Upper and lower bounds on the expected optimization error $\mathbb{E}[F(\mathbf{x}_k) - \inf_{\mathbf{x}} F(\mathbf{x})]$ for constant-step-size SGD with various sampling strategies, after k passes over n individual functions, in terms of n, k. Boldface letters refer to new results in this paper. We note that the upper bound of [7] additionally requires that the Hessian of each f_i is Lipschitz, and the upper bounds of [7] and [8] require k to be larger than a problem-dependent parameter (depending for example on the condition number). Also, the upper bound of [15] requires functions which are generalized linear functions. Our lower bounds apply under all such assumptions, and for any value of n, k. Finally, we note that the upper bound of [8] is actually not on the optimization error for $\mathbf{x}_k$, but rather on a certain averaging of several iterates – see Remark 3 in the supplementary material for a further discussion.

In the past few years, some progress has been made in this front, and we summarize the known results on the expected optimization error (or at least what these results imply[1]), as well as our new results, in Table 1. First, we note that for SGD with replacement, classic results imply an optimization error of $\mathcal{O}(1/nk)$ after nk stochastic iterations, and this is known to be tight (see for example [10]). For SGD with random reshuffling, better bounds have been shown in recent years, generally implying that when the number of epochs k is sufficiently large, such sampling schemes are better than with-replacement sampling, with optimization error decaying quadratically rather than linearly in k. However, the optimal dependencies on n, k and other problem-dependent parameters remain unclear (HaoChen and Sra [7] point out that for $k = 1$, one cannot hope to achieve worst-case error smaller than $\mathcal{O}(1/n)$, but for $k > 1$ not much is known). Some other recent theoretical works on SGD with random reshuffling (but under somewhat different settings) include [13, 16]. For cyclic SGD, an $\mathcal{O}(1/k^2)$ upper bound was shown in [5], as well as a matching asymptotic lower bound in terms of k. For SGD with single shuffling, we are actually not aware of a rigorous theoretical analysis. Thus, we only have the upper bound trivially implied by the analysis for cyclic SGD, and for $k = 1$, the upper bound implied by the analysis for random reshuffling (since in that case there is no distinction between single shuffling and random reshuffling). Indeed, for single shuffling, even different epochs are not statistically independent, which makes the analysis particularly challenging.

In this paper, we provide lower bounds on the expected optimization error of SGD with these sampling heuristics, which complement the existing upper bounds and provide further insights on the advantages and disadvantages of each. We focus on constant-step size SGD, as it simplifies our analysis, and existing upper bounds in the literature are derived in the same setting. Before we present our contributions, we first specify the assumptions we use.

We consider finite-sum optimization problems as in Eq. (1), and our lower bound constructions satisfy the following rather specific conditions for some given positive parameters G, λ (recall that for lower bounds, making the assumptions more stringent actually strengthens the result):

Assumption 1 *$F(\mathbf{x})$ is a quadratic finite-sum function of the form $\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$ for some $n > 1$, which is λ-strongly convex. Each f_i is convex and quadratic, has λ-Lipschitz gradients, and moreover, is G-Lipschitz for any $\mathbf{x}$ such that $\|\mathbf{x} - \mathbf{x}^*\| \leq 1$ where $\mathbf{x}^* = \arg\min F(\mathbf{x})$. Also, the algorithm is initialized at some $\mathbf{x}_0$ for which $\|\mathbf{x}_0 - \mathbf{x}^*\| \leq 1$.*

We now turn to present our contributions. For SGD with random reshuffling (Algorithm 1), we prove the following theorem:

Theorem 1 *For any $k \geq 1, n > 1$, and positive G, λ such that $G \geq 6\lambda$, there exists a function F on $\mathbb{R}$ and an initialization point x_0 satisfying Assumption 1, such that for any step size $\eta > 0$,*

$$\mathbb{E}\left[F(x_k) - \inf_x F(x)\right] \;\geq\; c \cdot \min\left\{\lambda\, ,\, \frac{G^2}{\lambda}\left(\frac{1}{(nk)^2} + \frac{1}{nk^3}\right)\right\}\;,$$

where $c > 0$ is a universal constant.

For nk large, this implies a lower bound of $\Omega(1/(nk)^2 + 1/nk^3)$. We conjecture that it is tight, as it seems to combine the "best" behaviors of existing upper bounds: It behaves as $1/n$ for a small constant number k of passes (which is optimal as discussed above), interpolating to $\mathcal{O}(1/(nk)^2)$ when k is large enough, and

[1] For example, some of these papers focus on bounding $\mathbb{E}[\|\mathbf{x}_k - \mathbf{x}^*\|^2]$ where $\mathbf{x}^*$ is the minimum of $F(\cdot)$, rather than the expected optimization error $\mathbb{E}[F(\mathbf{x}_k) - F(\mathbf{x}^*)]$. However, for strongly convex and smooth functions, $\|\mathbf{x}_k - \mathbf{x}^*\|^2$ and $F(\mathbf{x}_k) - F(\mathbf{x}^*)$ are the same up to the strong convexity and smoothness parameters, see for example [11].

contains a term decaying cubically with k. Moreover, the lower bound holds under more general conditions than the upper bounds: For example, it holds for any n, k, and even if the function under consideration is quadratic and on $\mathbb{R}$.

For SGD with a single shuffling (Algorithm 2), we prove the following theorem:

Theorem 2 *For any $k \geq 1, n > 1$, and positive G, λ such that $G \geq 6\lambda$, there exists a function F on $\mathbb{R}$ and an initialization point x_0 satisfying Assumption 1, such that for any step size $\eta > 0$,*

$$\mathbb{E}\left[F(x_k) - \inf_x F(x)\right] \;\geq\; c \cdot \min\left\{\lambda\;,\;\frac{G^2}{\lambda n k^2}\right\}\;,$$

where $c > 0$ is a universal constant.

For nk large this implies a lower bound of $\Omega(1/nk^2)$. Although we are not aware of an upper bound to compare to, this lower bound already proves an inherent performance gap compared to random reshuffling: Indeed, in the latter case there is an upper bound of $\mathcal{O}(1/(nk)^2 + 1/k^3)$, which is smaller than the $\Omega(1/nk^2)$ lower bound for single shuffling when k is sufficiently large. This implies that the added computational effort of repeatedly reshuffling the functions can provably pay off in terms of the convergence rate.

For cyclic SGD (Algorithm 3), we prove the following theorem:

Theorem 3 *For any $k \geq 1, n > 1$, and positive G, λ such that $G \geq 6\lambda$, there exists a function F on $\mathbb{R}$ and an initialization point x_0 satisfying Assumption 1, such that if we run cyclic GD for k epochs with any step size $\eta > 0$, then*

$$F(x_k) - \inf_x F(x) \;\geq\; c \cdot \min\left\{\lambda\;,\;\frac{G^2}{\lambda k^2}\right\}$$

where $c > 0$ is a universal constant.

For large k, this provides an $\Omega(1/k^2)$ lower bound. We note that a similar bound (at least asymptotically and for a certain n) is already implied by [5, Theorem 3.4]. Our contribution here is to present a more explicit and non-asymptotic lower bound which holds for any k and n. All proofs of the above theorems appear in the supplementary material.

Acknowledgments

This research is supported in part by NSF/BSF grant 71286701.

References

[1] Dimitri P Bertsekas and Athena Scientific. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

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

[3] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*. Springer, 2012.

[4] 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.

[5] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo Parrilo. Convergence rate of incremental gradient and Newton methods. *arXiv preprint arXiv:1510.08562*, 2015.

[6] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo Parrilo. Why random reshuffling beats stochastic gradient descent. *arXiv preprint arXiv:1510.08560*, 2015.

[7] Jeffery Z HaoChen and Suvrit Sra. Random shuffling beats sgd after finite epochs. *arXiv preprint arXiv:1806.10077*, 2018.

[8] Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Sgd without replacement: Sharper rates for general smooth convex functions. *arXiv preprint arXiv:1903.01463*, 2019.

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

[10] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[11] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[12] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1571–1578. Omnipress, 2012.

[13] B. Recht and C. Ré. Beneath the valley of the noncommutative arithmetic-geometric mean inequality: conjectures, case-studies, and consequences. In *COLT*, 2012.

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

[15] Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in neural information processing systems*, pages 46–54, 2016.

[16] Bicheng Ying, Kun Yuan, Stefan Vlaski, and Ali H Sayed. Stochastic learning under random reshuffling with constant step-sizes. *IEEE Transactions on Signal Processing*, 67(2):474–489, 2018.