Proximal and Federated Random Reshuffling

Konstantin Mishchenko [1] Ahmed Khaled [2] Peter Richtárik [3]

Abstract

Random Reshuffling (RR), also known as Stochastic Gradient Descent (SGD) without replacement, is a popular and theoretically grounded method for finite-sum minimization. We propose two new algorithms: Proximal and Federated Random Reshuffling (ProxRR and FedRR). The first algorithm, ProxRR, solves composite finite-sum minimization problems in which the objective is the sum of a (potentially non-smooth) convex regularizer and an average of n smooth objectives. ProxRR evaluates the proximal operator once per epoch only. When the proximal operator is expensive to compute, this small difference makes ProxRR up to n times faster than algorithms that evaluate the proximal operator in every iteration, such as proximal (stochastic) gradient descent. We give examples of practical optimization tasks where the proximal operator is difficult to compute and ProxRR has a clear advantage. One such task is federated or distributed optimization, where the evaluation of the proximal operator corresponds to communication across the network. We obtain our second algorithm, FedRR, as a special case of ProxRR applied to federated optimization, and prove it has a smaller communication footprint than either distributed gradient descent or Local SGD. Our theory covers both constant and decreasing stepsizes, and allows for importance resampling schemes that can improve conditioning, which may be of independent interest. Our theory covers both convex and nonconvex regimes. Finally, we corroborate our results with experiments on real data sets.

1. Introduction

Modern theory and practice of training supervised machine learning models is based on the paradigm of regularized empirical risk minimization (ERM) (Shalev-Shwartz & Ben-David, 2014). While the ultimate goal of supervised learning is to train models that generalize well to unseen data, in practice only a finite data set is available during training. Settling for a model merely minimizing the average loss on this training set—the empirical risk—is insufficient, as this often leads to over-fitting and poor generalization performance in practice. Due to this reason, empirical risk is virtually always amended with a suitably chosen regularizer whose role is to encode prior knowledge about the learning task at hand, thus biasing the training algorithm towards better performing models.

The regularization framework is quite general and perhaps surprisingly it also allows us to consider methods for federated learning (FL)—a paradigm in which we aim at training model for a number of clients that do not want to reveal their data (Konečný et al., 2016; McMahan et al., 2017; Kairouz et al., 2019). The training in FL usually happens on devices with only a small number of model updates being shared with a global host. To this end, Federated Averaging algorithm has emerged that performs Local SGD updates on the clients' devices and periodically aggregates their average. Its analysis usually requires special techniques and deliberately constructed sequences hindering the research in this direction. We shall see, however, that the convergence of our FedRR follows from merely applying our algorithm for regularized problems to a carefully chosen reformulation.

Formally, regularized ERM problems are optimization problems of the form

$$\min_{x\in\mathbb{R}^d} \left[P(x) := \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(x) + \psi(x)\right], \qquad (1)$$

where $f_i\colon \mathbb{R}^d \to \mathbb{R}$ is the loss of model parameterized by vector $x \in \mathbb{R}^d$ on the i-th training data point, and $\psi\colon \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ is a regularizer. Let $[n] := \{1, 2, \dots, n\}$. We shall make the following assumption throughout the paper without explicitly mentioning it:

Assumption 1. The functions f_i are L_i-smooth, and the regularizer ψ is proper, closed and convex. Let $L_{\max} := \max_{i\in[n]} L_i$.

[1] CNRS, DI ENS, Inria [2] Princeton University [3] KAUST. Correspondence to: Konstantin Mishchenko <konsta.mish@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

In some results we will additionally assume that either the individual functions f_i, or their average $f := \frac{1}{n}\sum_i f_i$, or the regularizer ψ are μ-strongly convex. Whenever we need such additional assumptions, we will make this explicitly clear. While all these concepts are standard, we review them briefly in Appendix A.

Proximal SGD. When the number n of training data points is huge, as is increasingly common in practice, the most efficient algorithms for solving (1) are stochastic first-order methods, such as stochastic gradient descent (SGD) (Bordes et al., 2009), in one or another of its many variants proposed in the last decade (Shang et al., 2018; Pham et al., 2020). These method almost invariably rely on alternating stochastic gradient steps with the evaluation of the proximal operator

$$\mathrm{prox}_{\gamma\psi}(x) := \mathrm{argmin}_{z\in\mathbb{R}^d}\left\{\gamma\psi(z) + \tfrac{1}{2}\|z-x\|^2\right\}.$$

The simplest of these has the form

$$x_{k+1}^{\mathrm{SGD}} = \mathrm{prox}_{\gamma_k\psi}(x_k^{\mathrm{SGD}} - \gamma_k \nabla f_{i_k}(x_k^{\mathrm{SGD}})), \qquad (2)$$

where i_k is an index from $\{1, 2, \ldots, n\}$ chosen uniformly at random, and $\gamma_k > 0$ is a properly chosen learning rate. Our understanding of (2) is quite mature; see (Gorbunov et al., 2020) for a general treatment which considers methods of this form in conjunction with more advanced stochastic gradient estimators in place of ∇f_{i_k}.

Applications such as training sparse linear models (Tibshirani, 1996), nonnegative matrix factorization (Lee & Seung, 1999), image deblurring (Rudin et al., 1992; Vogel, 2002), and training with group selection (Yuan & Lin, 2006) all rely on the use of hand-crafted regularizes. For many of them, the proximal operator can be evaluated efficiently, and SGD is near or at the top of the list of efficient training algorithms.

Random reshuffling. A particularly successful variant of SGD is based on the idea of random shuffling (permutation) of the training data followed by n iterations of the form (2), with the index i_k following the pre-selected permutation (Bottou, 2012). This process is repeated several times, each time using a new freshly sampled random permutation of the data, and the resulting method is known under the name *Random Reshuffling (RR)*. When the same permutation is used throughout, the technique is known under the name *Shuffle-Once (SO)*.

One of the main advantages of this approach is rooted in its intrinsic ability to avoid cache misses when reading the data from memory, which enables a significantly faster implementation. Furthermore, RR is often observed to converge in fewer iterations than SGD in practice. This can intuitively be ascribed to the fact that while due to its sampling-with-replacement approach SGD can miss to learn from some data points in any given epoch, RR will learn from each data point in each epoch. Understanding the random reshuffling trick, and why it works, has been a non-trivial open problem for the past decade (Bottou, 2009; Recht & Ré, 2012; Gürbüzbalaban et al., 2019; Haochen & Sra, 2019), and has inspired significant ongoing research effort (Shamir, 2016; Haochen & Sra, 2019; Nagaraj et al., 2019; Mishchenko et al., 2020; Ahn et al., 2020).

2. Contributions

Our goal in this paper is twofold: we develop RR in new settings (namely, for proximal and federated learning), and also address some of the shortcomings of existing theory, in particular in the dependence on the condition number as well as step-size scheduling. The difficulty of analyzing RR has been the main obstacle in the development of even some of the most seemingly benign extensions of the method. Indeed, while these extensions are well understood in combination with its much simpler-to-analyze cousin SGD, *to the best of our knowledge, there exists no theoretical analysis of proximal, parallel, and importance sampling variants of RR with both constant and decreasing stepsizes, and in most cases it is not even clear how should such methods be constructed.* In this section we outline the key contributions of our work, and also offer a few intuitive explanations motivating some of the development.

2.1. RR in new problem settings

- **New algorithm: ProxRR.** Despite rich literature on Proximal SGD (Gorbunov et al., 2020), it is not obvious how one should extend RR to solve problem (1) when a regularizer ψ is present. Indeed, the standard practice for SGD is to apply the proximal operator after each stochastic step (Duchi & Singer, 2009), i.e., in analogy with (2). On the other hand, RR is motivated by the fact that a data pass better approximates the full gradient step (Bertsekas, 2011). The following example shows that if we apply the proximal operator after each step of RR, **we would no longer approximate the full gradient after an epoch**:

Example 1. Let $n = 2$, $\psi(x) = \frac{1}{2}\|x\|^2$, $f_1(x) = \langle c_1, x\rangle$, $f_2(x) = \langle c_2, x\rangle$ with some $c_1, c_2 \in \mathbb{R}^d$, $c_1 \neq c_2$. Let $x_0 \in \mathbb{R}^d$, $\gamma > 0$ and define $x_1 = x_0 - \gamma\nabla f_1(x_0)$, $x_2 = x_1 - \gamma\nabla f_2(x_1)$. Then, we have $\mathrm{prox}_{2\gamma\psi}(x_2) = \mathrm{prox}_{2\gamma\psi}(x_0 - 2\gamma\nabla f(x_0))$. However, if $\tilde{x}_1 = \mathrm{prox}_{\gamma\psi}(x_0 - \gamma\nabla f_1(x_0))$ and $\tilde{x}_2 = \mathrm{prox}_{\gamma\psi}(x_1 - \gamma\nabla f_2(\tilde{x}_1))$, then $\tilde{x}_2 \neq \mathrm{prox}_{2\gamma\psi}(x_0 - 2\gamma\nabla f(x_0))$.

Motivated by this observation, we propose ProxRR (Algorithm 1), in which the proximal operator is applied at the end of each epoch of RR, i.e., after each pass through all randomly reshuffled data. A notable property of Algorithm 1 is that *only a single proximal operator evaluation is needed*

Algorithm 1 Proximal Random Reshuffling (ProxRR) and Shuffle-Once (ProxSO)

```
1: Input: Stepsizes $\gamma_t > 0$, initial vector $x_0 \in \mathbb{R}^d$, number of epochs $T$
2: Sample a permutation $\pi = (\pi_{0u}, \pi_1, \dots, \pi_{n-1})$ of $[n]$ (Do step 1 only for ProxSO)
3: for epochs $t = 0, 1, \dots, T-1$ do
4:     Sample a permutation $\pi = (\pi_0, \pi_1, \dots, \pi_{n-1})$ of $[n]$ (Do step 3 only for ProxRR)
5:     $x_t^0 = x_t$
6:     for $i = 0, 1, \dots, n-1$ do
7:         $x_t^{i+1} = x_t^i - \gamma_t \nabla f_{\pi_i}(x_t^i)$
8:     end for
9:     $x_{t+1} = \mathrm{prox}_{\gamma_t n \psi}(x_t^n)$
10: end for
```

during each data pass. This is in sharp contrast with the way Proximal SGD works, and offers significant advantages in regimes where the evaluation of the proximal mapping is expensive (e.g., comparable to the evaluation of n gradients $\nabla f_1, \dots, \nabla f_n$).

• **Convergence of ProxRR (for strongly convex functions or regularizer).** We establish several convergence results for ProxRR, of which we highlight two here. Both offer a linear convergence rate with a fixed stepsize to a neighborhood of the solution. In both we reply on Assumption 1. Firstly, in the case when in addition, each f_i is μ-strongly convex, we prove the rate (see Theorem 2)

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] \le (1-\gamma\mu)^{nT} \|x_0 - x_*\|^2 + \frac{2\gamma^2\sigma_{\mathrm{rad}}^2}{\mu},$$

where $\gamma_t = \gamma \le {}^1\!/_{L_{\max}}$ is the stepsize, and σ_{rad}^2 is a *shuffling radius* constant (for precise definition, see (4)). In Theorem 1 we bound the shuffling radius in terms of $\|\nabla f(x_*)\|^2$, n, $L_{\max}$ and the more common quantity $\sigma_*^2 := \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x_*) - \nabla f(x_*)\|^2$. We give a similar rate of convergence if ψ is also strongly-convex.

Both mentioned rates show exponential (linear in logarithmic scale) convergence to a neighborhood whose size is proportional to $\gamma^2\sigma_{\mathrm{rad}}^2$. Since we can choose γ to be arbitrarily small or periodically decrease it, this implies that the iterates converge to x_* in the limit. Moreover, we show in Section 3 that when $\gamma = \mathcal{O}(\frac{1}{T})$ the error is $\mathcal{O}(\frac{1}{T^2})$, which is superior to the $\mathcal{O}(\frac{1}{T})$ error of SGD. All of our results in the convex case apply to the Shuffle-Once algorithm as well.

• **Convergence of ProxRR for nonconvex optimization.** In the nonconvex regime, and under suitable assumptions, we establish (see Theorems 5 and 3) an $\mathcal{O}(\frac{1}{\gamma T})$ rate up to a neighborhood of size $\mathcal{O}(\gamma^2)$. For a certain stepsize it yields an $\mathcal{O}(\frac{1}{\varepsilon^3})$ convergence rate.

• **Application to Federated Learning.** In Section 5 we describe an application of our results to federated learning (Konečný et al., 2016; McMahan et al., 2017; Kairouz et al., 2019). In this way we obtain the FedRR method, which is similar to Local SGD, except the local solver is a single pass of RR over the local data. Empirically, FedRR can be vastly superior to Local SGD (see Figure 2). Remarkably, we also show that the rate of FedRR *beats the best known lower bound for Local SGD* due to (Woodworth et al., 2020) (we needed to adapt it from the original online to the finite-sum setting we consider in this paper) for large enough n. See Appendix G for more details.

2.2. Improving vanilla RR

Besides the above results, we describe two extensions that improve upon the rates of vanilla Random Reshuffling (i.e. with no prox) and which are of independent interest.

• **Extension 1: Importance resampling for Proximal RR.**

All existing rates of convergence of RR in the strongly-convex regime exhibit a dependence on $\max_i L_i/\mu$ (e.g. (Mishchenko et al., 2020; Ahn et al., 2020) and others), where L_i is the smoothness constant of f_i. We observe that this is highly suboptimal compared to the $\bar{L}/\mu$ rate (for $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$) that SGD and variance-reduced methods can achieve: indeed, the difference between these two condition numbers can be of order n (Gower et al., 2019). This is a serious problem as the difference between the rate of convergence of RR and SGD is tightly related to the condition number (Safran & Shamir, 2021). In other words: **existing results on RR can be suboptimal by up to a factor of n compared to the best known rates for SGD**.

To handle this, we reformulate problem (1) into a similar problem with a larger number of summands. In particular, for each $i \in [n]$ we include n_i copies of the function $\frac{1}{n_i} f_i$, and then take average of all $N = \sum_i n_i$ functions constructed this way. The value of n_i depends on the "importance" of f_i, described below. We then apply ProxRR to this reformulation. If f_i is L_i-smooth for all $i \in [n]$ and we let $\bar{L} := \frac{1}{n}\sum_i L_i$, then we choose $n_i = \lceil {}^{L_i}\!/_{\bar{L}} \rceil$. It is easy to show that $N \le 2n$, and hence our reformulation leads to at most a doubling of the number of functions forming the finite sum. However, the overall complexity of RR/ProxRR applied to this reformulation will depend on $\bar{L}$ instead of $\max_i L_i$ (see Theorem 9), which can improve the convergence rate by up to a factor of n. For details of the construction and our complexity results, Appendix I.

• **Extension 2: Decreasing stepsizes.**

The convergence of RR is not always exact and depends on the parameters of the objective. Similarly, if the shuffling radius σ_{rad}^2 is positive, and we wish to find an ε-approximate solution, the optimal choice of a fixed stepsize for ProxRR will depend on ε. This deficiency can be fixed by using decreasing stepsizes in both vanilla RR (Ahn et al., 2020) and in SGD (Stich, 2019). However, the rate given by (Ahn et al., 2020) does not recover linear convergence in the absence of noise (i.e. $\sigma_* = 0$). We propose a stepsize schedule that allows us to both recover linear convergence in the absence of noise and recover the optimal $\mathcal{O}((nT^2)^{-1})$ rate of convergence of RR in the presence of noise. Our proposed stepsize schedule is thus noise-adaptive without requiring any knowledge of the magnitude of the noise σ_*. For details, see Appendix J.

3. Theory for strongly convex objectives

3.1. Preliminaries

In the strongly-convex setting, we build upon the notions of *shuffling variance* introduced by Mishchenko et al. (2020) for analyzing RR. Given a stepsize $\gamma > 0$ (held constant during each epoch) and a permutation π of $\{1, 2, \ldots, n\}$, we introduce the points $x_*^1, x_*^2, \ldots, x_*^n$ defined by

$$x_*^i := x_* - \gamma \sum_{j=0}^{i-1} \nabla f_{\pi_j}(x_*), \quad i = 1, \ldots, n. \tag{3}$$

The intuition behind this definition is fairly simple: if we performed i steps starting at x_*, we would end up close to x_*^i. To quantify the closeness, we define the *shuffling radius* and then show how to upper bound it.

Definition 1 (Shuffling radius)**.** Given a stepsize $\gamma > 0$ and a random permutation π of $\{1, 2, \ldots, n\}$ used in Algorithm 1, define $x_*^i = x_*^i(\gamma, \pi)$ as in (3). Then, the shuffling radius is defined by

$$\sigma_{\mathrm{rad}}^2(\gamma) := \max_{i=0,\ldots,n-1} \left[\frac{1}{\gamma^2} \mathbb{E}_\pi \left[D_{f_{\pi_i}}(x_*^i, x_*) \right] \right], \tag{4}$$

where $D_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle$ is the Bregman divergence associated with f evaluated at x, y and where the expectation is taken with respect to the randomness in the permutation π. If there are multiple stepsizes $\gamma_1, \gamma_2, \ldots$ used in Algorithm 1, we take the maximum of all of them as the shuffling radius, i.e., $\sigma_{\mathrm{rad}}^2 := \max_{t \geq 1} \sigma_{\mathrm{rad}}^2(\gamma_t)$.

Theorem 1 (Bounding the shuffling radius)**.** For any stepsize $\gamma > 0$ and any random permutation π of $\{1, 2, \ldots, n\}$ we have $\sigma_{\mathrm{rad}}^2 \leq \frac{L_{\max}}{2} n \left(n \|\nabla f(x_*)\|^2 + \frac{1}{2}\sigma_*^2 \right)$, where x_* is a solution of Problem (1) and σ_*^2 is the population variance at the optimum

$$\sigma_*^2 := \frac{1}{n} \sum_{i=1}^n \|\nabla f_i(x_*) - \nabla f(x_*)\|^2. \tag{5}$$

All proofs are relegated to the supplementary material. In order to better understand the bound given by Theorem 1, observe that if there is no proximal operator (i.e., $\psi = 0$) then $\nabla f(x_*) = 0$ and we get that $\sigma_{\mathrm{rad}}^2 \leq \frac{L_{\max} n \sigma_*^2}{4}$. This recovers the existing upper bound on the shuffling variance of Mishchenko et al. (2020) for vanilla RR. On the other hand, if $\nabla f(x_*) \neq 0$ then we get an additive term of size proportional to the squared norm of $\nabla f(x_*)$.

3.2. Convergence guarantees

Our first theorem establishes a convergence rate for Algorithm 1 applied with a constant stepsize to Problem (1) when each objective f_i is strongly convex. This assumption is commonly satisfied in machine learning applications where each f_i represents a regularized loss on some data points, as in ℓ_2 regularized linear regression and ℓ_2 regularized logistic regression.

Theorem 2. Let Assumption 1 be satisfied. Further, assume that each f_i is μ-strongly convex. If Algorithm 1 is run with constant stepsize $\gamma_t = \gamma \leq 1/L_{\max}$, then its iterates satisfy

$$\mathbb{E}\left[\|x_T - x_*\|^2 \right] \leq (1 - \gamma\mu)^{nT} \|x_0 - x_*\|^2 + \frac{2\gamma^2 \sigma_{\mathrm{rad}}^2}{\mu}.$$

We can convert the guarantee of Theorem 2 to a convergence rate by properly tuning the stepsize and using the upper bound of Theorem 1 on the shuffling radius. In particular, if we choose the stepsize as $\gamma = \min\left\{ \frac{1}{L_{\max}}, \frac{\sqrt{\varepsilon\mu}}{\sqrt{2}\sigma_{\mathrm{rad}}} \right\}$, and let $\kappa := L_{\max}/\mu$ and $r_0 := \|x_0 - x_*\|^2$, then we obtain $\mathbb{E}\left[\|x_T - x_*\|^2 \right] = \mathcal{O}(\varepsilon)$ provided that the total number of iterations $K_{\mathrm{RR}} = nT$ is at least

$$K_{\mathrm{RR}} \geq \left[\left(\kappa + \frac{\sqrt{\kappa n}}{\sqrt{\varepsilon\mu}} (\sqrt{n} \|\nabla f(x_*)\| + \sigma_*) \right] \log\left(\frac{2r_0}{\varepsilon} \right). \tag{6}$$

Comparison with vanilla RR. If there is no proximal operator, then $\|\nabla f(x_*)\| = 0$ and we recover the earlier result of Mishchenko et al. (2020) on the convergence of RR without proximal, which is optimal in ε up to logarithmic factors. On the other hand, when the proximal operator is nonzero, we get an extra term in the complexity proportional to $\|\nabla f(x_*)\|$: thus, even when all the functions are the same (i.e., $\sigma_* = 0$), we do not recover the linear convergence of Proximal Gradient Descent (Karimi et al., 2016; Beck, 2017). This can be easily explained by the fact that Algorithm 1 performs n gradient steps per one proximal step. Hence, even if $f_1 = \cdots = f_n$, Algorithm 1 does not reduce to Proximal Gradient Descent. We note that other algorithms for composite optimization which may not take a proximal step at every iteration also suffer from the same dependence (Patrascu & Irofti, 2021).

Comparison with proximal SGD. In order to compare (3.2) against the complexity of Proximal SGD (Algorithm 3 in Appendix C), we recall that Proximal SGD achieves $\mathbb{E}\left[\|x_K - x_*\|^2 \right] = \mathcal{O}(\varepsilon)$ if either f or ψ is μ-strongly convex and

$$K_{\mathrm{SGD}} \geq \left(\kappa + \frac{\sigma_*^2}{\varepsilon\mu^2} \right) \log\left(\frac{2r_0}{\varepsilon} \right). \tag{7}$$

This result is standard (Needell et al., 2016; Gower et al., 2019), with the exception that we do not know any proof in the literature for the case when ψ is strongly convex. For completeness, we prove it in Appendix C.

By comparing K_{SGD} (given by (7)) and K_{RR} (given by (3.2)), we see that ProxRR has milder dependence on ε than Proximal SGD. In particular, ProxRR converges faster whenever the target accuracy ε is small enough to satisfy $\varepsilon \leq \frac{1}{L_{\max} n \mu} \left(\frac{\sigma_*^4}{n \|\nabla f(x_*)\|^2 + \sigma_*^2} \right)$. Furthermore, ProxRR is much better when we consider *proximal iteration complexity* (# of proximal operator access), in which case the complexity of ProxRR (3.2) is reduced by a factor of n (because we take one proximal step every n iterations), while the proximal iteration complexity of Proximal SGD remains the same as (7). In this case, ProxRR is better whenever the accuracy ε satisfies

$$\varepsilon \geq \frac{nG^2}{L_{\max}\mu} \qquad \text{or} \qquad \varepsilon \leq \frac{n\sigma_*^4}{L_{\max}\mu G^2},$$

where $G^2 := n\|\nabla f(x_*)\|^2 + \sigma_*^2$. We can see that if the target accuracy is large enough or small enough, and if the cost of proximal operators dominates the computation, ProxRR is much quicker to converge than Proximal SGD.

Comparison with ProxSVRG. Variance-reduced methods can improve the rate of convergence of SGD from sublinear to linear by using better estimates of the gradients ∇f_i. ProxSVRG (Xiao & Zhang, 2014) is a common variance-reduced method used for solving (1) in practice (Tang et al., 2020). Xiao & Zhang (2014) give the following iteration complexity (in both proximal & gradient oracle calls), up to constant factors:

$$K_{\mathrm{SVRG}} \geq (\kappa + n) \log\left(\frac{4r_0}{\epsilon}\right). \tag{8}$$

Comparing (8) and reveals that for gradient oracle calls, using ProxRR is beneficial if the accuracy satisfies $\epsilon > \frac{\kappa G^2}{\mu n}$ where $G^2 := n\|\nabla f(x_*)\|^2 + \sigma_*^2$. This bounds means that ProxRR is better when the problem is better-conditioned or when the number of functions n is large and the minimizers of both P and f match well (i.e. $\|\nabla f(x_*)\|$ is small). The situation is much better for proximal oracle calls, where ProxRR is better when

$$\epsilon > \frac{\kappa G^2}{\mu n^2} = \frac{\kappa}{\mu n} \cdot \left(\|\nabla f(x_*)\|^2 + \frac{1}{n}\sigma_*^2 \right).$$

Observe that this expression is decreasing in n as long as σ_*^2/n is nonincreasing in n, a very mild requirement satisfied, for example, if the functions f_i are themselves sampled i.i.d. according to some distribution with mean f and bounded variance.

Extension for strongly-convex regularizers. In Theorem 2, we assume that each f_i is μ-strongly convex. This is motivated by the common practice of using ℓ_2 regularization in machine learning. However, applying ℓ_2 regularization in every step of Algorithm 1 can be expensive when the data are sparse and the iterates x_t^i are dense, because it requires accessing each coordinate of x_t^i which can be much more expensive than computing sparse gradients $\nabla f_i(x_t^i)$. Alternatively, we may instead choose to put the ℓ_2 regularization inside ψ and only ask that ψ be strongly convex—this way, we can save a lot of time as we need to access each coordinate of the dense iterates x_t^i only once per epoch rather than every iteration. Theorem 8 in Appendix D.3 gives a convergence guarantee in this setting which is similar to that of Theorem 2.

4. Theory for non-convex objectives

We shall now present our theory for the nonconvex case. To quantify convergence, we define the proximal-gradient mapping, which was also used in the prior literature to show convergence of Proximal SGD.

Definition 2. Given a stepsize $\gamma > 0$, a convex function ψ and arbitrary f, we define the proximal-gradient mapping as

$$\mathcal{G}_\gamma(x) := \tfrac{1}{\gamma}\left[x - \mathrm{prox}_{\gamma\psi}(x - \gamma\nabla f(x))\right].$$

Similarly to Theorem 1, the analysis shows that a gradient term appears in the variance bound. However, in contrast to the convex settings of Theorem 1, there might not exist an optimum to which the iterates would converge and we cannot use $\|\nabla f(x_*)\|^2$ in the variance bound. For this reason, we resort to the following assumption that bounds full gradients in terms of proximal-gradient mapping and an extra constant.

Assumption 2. There exists a constant $\zeta \geq 0$ such that the full gradient of f is uniformly bounded by the proximal-gradient mapping and ζ

$$\|\nabla f(x)\|^2 \leq \|\mathcal{G}_{\gamma n}(x)\|^2 + \zeta^2$$

for any $x \in \mathrm{dom}(\psi)$ and $\gamma > 0$.

We note that this assumption is trivially satisfied with $\zeta = 0$ if $\psi \equiv 0$ because in that case, $\mathcal{G}_\gamma(x) \equiv \nabla f(x)$. Therefore, when there is no proximal term, it is not an extra assumption compared to the analysis of Mishchenko et al. (2020). We will also rely on the following measure of gradient variance, which we need for the same reason that there might be no optimum x_* to measure the variance the way we did for Theorem 1.

Assumption 3. There exists a constant $\sigma > 0$ such that $\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f(x)\|^2 \leq \sigma^2$ for any $x \in \mathbb{R}^d$.

Note that we can relax this assumption by introducing extra terms in the right-hand side as in (Khaled & Richtárik, 2020). Nevertheless, for the sake of simplicity and readability, we prefer the stronger version as presented above. Theorem 3 gives our main convergence result:

Theorem 3 (Convergence result in the nonconvex case)**.** Let Assumptions 2 and 3 hold and choose any $\gamma \leq \frac{1}{5L_{\max}n}$.

Then,

$$\min_{t=0,\ldots,T-1} \mathbb{E}\left[\|\mathcal{G}_{\gamma n}(x_t)\|^2\right] \le \frac{4(P(x_0)-P_*)}{\gamma n T} + 2\gamma^2 L_{\max} n^2 \zeta^2 + 2\gamma^2 L_{\max}^2 n \sigma^2.$$

Instead of obtaining a convergence guarantee on the minimum of the prox-grad mapping norm, we can get the same guarantee by randomly choosing an iterate. This is standard in stochastic nonconvex optimization since, for any oracle with access only to stochastic gradients, obtaining a guarantee for a fixed iterate (e.g. the final iterate) is impossible in general (Drori & Shamir, 2020).

Obtaining a complexity. Set the stepsize γ to

$$\gamma = \min\left\{\frac{1}{5L_{\max}n}, \frac{\varepsilon}{L_{\max}\sqrt{n}\sigma + \sqrt{L_{\max}}n\zeta}\right\}.$$

Then plugging into Theorem 3 and denoting $\delta_0 := P(x_0) - P_*$, we obtain that in order to get an ϵ-stationary solution the complexity in terms of full number of stochastic gradients nT equal to

$$nT = \mathcal{O}\left(\frac{\delta_0 L_{\max} n}{\varepsilon^2} + \frac{\delta_0 L_{\max}\sqrt{n}\sigma}{\varepsilon^3} + \frac{\delta_0\sqrt{L_{\max}}n\zeta}{\varepsilon^3}\right).$$

When the accuracy ϵ is small enough, this rate is better than the $\mathcal{O}(\epsilon^{-4})$ rate of convergence of Proximal SGD (Davis & Drusvyatskiy, 2018).

5. FedRR: application of ProxRR to federated learning

Let us consider now the problem of minimizing the average of $N = \sum_{m=1}^M N_m$ functions that are stored on M devices, which have $N_1, \ldots, N_M$ samples correspondingly,

$$\min_{x\in\mathbb{R}^d} F(x) + R(x), \quad F(x) = \frac{1}{N}\sum_{m=1}^M F_m(x), \tag{9}$$

where $F_m(x) := \sum_{j=1}^{N_m} f_{mj}(x)$. For example, $f_{mj}(x)$ can be the loss associated with a single sample (X_{mj}, y_{mj}), where pairs (X_{mj}, y_{mj}) follow a distribution D_m that is specific to device m. An important instance of such formulation is federated learning, where M devices train a shared model by communicating periodically with a central node. We normalize the objective in (9) by N as this is the total number of functions after we expand each F_m into a sum. We denote the solution of (9) by x_*.

Extending the space. To rewrite the problem as an instance of (1), we are going to consider a bigger product space, which is sometimes used in distributed optimization (Bianchi et al., 2015). Let us define $n := \max\{N_1, \ldots, N_m\}$ and introduce ψ_C, the *consensus* constraint, defined via

$$\psi_C(x_1, \ldots, x_M) := \begin{cases} 0, & x_1 = \cdots = x_M \\ +\infty, & \text{otherwise} \end{cases}.$$

By introducing dummy variables $x_1, \ldots, x_M$ and adding the constraint $x_1 = \cdots = x_M$, we arrive at the intermediate problem

$$\min_{x_1,\ldots,x_M\in\mathbb{R}^p} \frac{1}{N}\sum_{m=1}^M F_m(x_m) + (R+\psi_C)(x_1, \ldots, x_M),$$

where $R + \psi_C$ is defined, with a slight abuse of notation, as $(R+\psi_C)(x_1,\ldots,x_M) = R(x_1)$ if $x_1 = \cdots = x_M$, and $(R+\psi_C)(x_1,\ldots,x_M) = +\infty$ otherwise.

Since we have replaced R with a more complicated regularizer $R + \psi_C$, we need to understand how to compute the proximal operator of the latter. We show (Lemma 8 in the supplementary) that the proximal operator of $(R + \psi_C)$ is merely the projection onto $\{(x_1, \ldots, x_M) \mid x_1 = \cdots = x_M\}$ followed by the proximal operator of R with a smaller stepsize.

Reformulation. To have n functions in every F_m, we write F_m as a sum with extra $n - N_m$ zero functions, $f_{mj}(x) \equiv 0$ for any $j > N_m$, so that $F_m(x_m) = \sum_{j=1}^n f_{mj}(x_m) = \sum_{j=1}^{N_m} f_{mj}(x_m) + \sum_{j=N_m+1}^n 0$. We can now stick the vectors together into $\boldsymbol{x} = (x_1, \ldots, x_M) \in \mathbb{R}^{M\cdot d}$ and multiply the objective by $\frac{N}{n}$, which gives the following reformulation:

$$\min_{\boldsymbol{x}\in\mathbb{R}^{M\cdot d}} \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x}) + \psi(\boldsymbol{x}), \tag{10}$$

where $\psi(\boldsymbol{x}) := \frac{N}{n}(R + \psi_C)$ and

$$f_i(\boldsymbol{x}) = f_i(x_1, \ldots, x_M) := \sum_{m=1}^M f_{mi}(x_m).$$

In other words, function $f_i(\boldsymbol{x})$ includes i-th data sample from each device and contains at most one loss from every device, while $F_m(x)$ combines all data losses on device m. Note that the solution of (10) is $\boldsymbol{x}_* := (x_*^\top, \ldots, x_*^\top)^\top$ and the gradient of the extended function $f_i(\boldsymbol{x})$ is given by $\nabla f_i(\boldsymbol{x}) = (\nabla f_{1i}(x_1)^\top, \cdots, \nabla f_{Mi}(x_M)^\top)^\top$. Therefore, a stochastic gradient step that uses $\nabla f_i(\boldsymbol{x})$ corresponds to updating all local models with the gradient of i-th data sample, without any communication.

Algorithm 1 for this specific problem can be written in terms of $x_1, \ldots, x_M$, which results in Algorithm 2. Note that since $f_{mi}(x_i)$ depends only on x_i, computing its gradient does not require communication. Only once the local epochs are finished, the vectors are averaged as the result of projecting onto the set $\{(x_1, \ldots, x_M) \mid x_1 = \cdots = x_M\}$.

Reformulation properties. To analyze FedRR, the only thing that we need to do is understand the properties of the reformulation (10) and then apply Theorem 2 or Theorem 8. The following lemma gives us the smoothness and strong convexity properties of (10).

Lemma 1. Let function f_{mi} be L_i-smooth and μ-strongly convex for every m. Then, f_i from reformulation (10) is L_i-smooth and μ-strongly convex.

Algorithm 2 Federated Random Reshuffling (FedRR)

```
Input: Stepsize $\gamma > 0$, initial vector $x_0 = x_0^0 \in \mathbb{R}^d$, number of epochs $T$, number of functions $N_m$ on each machine $m$, set $N = \sum_{m=1}^M N_m$ and $n = \max_m N_m$.
for epochs $t = 0, 1, \ldots, T-1$ do
    for $m = 1, \ldots, M$ locally in parallel do
        $x_{t,m}^0 = x_t$
        Sample permutation $\pi_{0,m}, \pi_{1,m}, \ldots, \pi_{N_m-1,m}$ of $\{1, 2, \ldots, N_m\}$
        for $i = 0, 1, \ldots, N_m - 1$ do
            $x_{t,m}^{i+1} = x_{t,m}^i - \gamma \nabla f_{\pi_{i,m}}(x_{t,m}^i)$
        end for
        $x_{t,m}^n = x_{t,m}^{N_m}$
    end for
    $x_{t+1} = \mathrm{prox}_{\frac{\gamma N}{n} R}\left(\frac{1}{M}\sum_{m=1}^M x_{t,m}^n\right)$
end for
```

The previous lemma shows that the conditioning of the reformulation is $\kappa = \frac{L_{\max}}{\mu}$ just as we would expect. Moreover, it implies that the requirement on the stepsize remains exactly the same: $\gamma \leq {}^1/_{L_{\max}}$. What remains unknown is the value of σ_{rad}^2, which plays a key role in the convergence bounds for ProxRR and ProxSO. To find an upper bound on σ_{rad}^2, let us define

$$\sigma_{m,*}^2 := \frac{1}{N_m}\sum_{j=1}^n \left\|\nabla f_{mj}(x_*) - \frac{1}{N_m}\nabla F_m(x_*)\right\|^2,$$

which is the variance of local gradients on device m. This quantity characterizes the convergence rate of local SGD (Yuan et al., 2020), so we should expect it to appear in our bounds too. The next lemma explains how to use it to upper bound σ_{rad}^2.

Lemma 2. The shuffling radius σ_{rad}^2 of the reformulation (10) is upper bounded by

$$\sigma_{\mathrm{rad}}^2 \leq L_{\max} \cdot \sum_{m=1}^M \left(\|\nabla F_m(x_*)\|^2 + \frac{n}{4}\sigma_{m,*}^2\right).$$

The lemma shows that the upper bound on σ_{rad}^2 depends on the sum of local variances $\sum_{m=1}^M \sigma_{m,*}^2$ as well as on the local gradient norms $\sum_{m=1}^M \|\nabla F_m(x_*)\|^2$. Both of these sums appear in the existing literature on convergence of Local GD/SGD (Woodworth et al., 2020; Yuan et al., 2020). We are now ready to present formal convergence results. For simplicity, we will consider heterogeneous and homogeneous cases separately and assume that $N_1 = \cdots = N_M = n$. To further illustrate generality of our results, we will present the heterogeneous assuming strong convexity R and the homogeneous under strong convexity of functions f_{mi}.

Heterogeneous data. In the case when the data are heterogeneous, we provide the first local RR method. We can apply either Theorem 2 or Theorem 8, but for brevity, we give only the corollary obtained from Theorem 8.

Theorem 4. Assume that functions f_{mi} are convex and L_i-smooth for each m and i. If R is μ-strongly convex and $\gamma \leq {}^1/_{L_{\max}}$, then we have for the iterates produced by Algorithm 2

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] \leq (1 + 2\gamma\mu n)^{-T}\|x_0 - x_*\|^2 + \frac{\gamma^2 L_{\max}}{M\mu}\sum_{m=1}^M \left(\|\nabla F_m(x_*)\|^2 + \frac{N}{4M}\sigma_{m,*}^2\right).$$

In the supplementary material (Appendix G), we show that our rates for FedRR improve over the best known rates for both Local SGD and Distributed Gradient Descent in the heterogeneous data seting.

For nonconvex analysis, we consider $R \equiv 0$ and require the following standard assumption.

Assumption 4 (Bounded variance and dissimilarity)**.** There exist constants $\sigma, \zeta > 0$ such that for any $x \in \mathbb{R}^d$ and

$$\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_{mi} - \frac{1}{n}\nabla F_m(x)\right\|^2 \leq \sigma^2 \qquad \text{and,}$$
$$\frac{1}{M}\sum_{m=1}^M \left\|\frac{1}{n}\nabla F_m(x) - \nabla F(x)\right\|^2 \leq \zeta^2.$$

Note that above $\frac{1}{n}\nabla F_m(x) = \frac{1}{N_m}\nabla F_m(x)$ is the gradient of a local dataset and $\nabla F(x) = \frac{1}{N}\sum_{l=1}^M \nabla F_l(x)$ is the full gradient on all data.

Theorem 5 (Nonconvex convergence)**.** Let Assumptions 1 and 4 be satisfied, and $R \equiv 0$ (no prox). Then, the communication complexity to achieve $\mathbb{E}\left[\|\nabla F(x_T)\|^2\right] \leq \varepsilon^2$ is

$$T = \mathcal{O}\left(\left(\frac{1}{\varepsilon^2} + \frac{\sigma}{\sqrt{n}\varepsilon^3} + \frac{\zeta}{\varepsilon^3}\right)(F(x_0) - F_*)\right).$$

Notice that by replicating the data locally on each device and thereby increasing the value of n without changing the objective, we can improve the second term in the communication complexity. In particular, if the data are not too dissimilar ($\sigma \gg \zeta$) and ε is small ($\frac{1}{\varepsilon^3} \gg \frac{1}{\varepsilon^2}$), the second term in the complexity dominates, and it helps to have more local steps. However, if the data are less similar, the nodes have to communicate more frequently to get more information about other objectives.

Homogeneous data. For simplicity, in the homogeneous (i.e., i.i.d.) data case we provide guarantees without the proximal operator. Since then we have $F_1(x) = \cdots = F_M(x)$, for any m it holds $\nabla F_m(x_*) = 0$, and thus $\sigma_{m,*}^2 = \frac{1}{n}\sum_{j=1}^n \|\nabla f_{mj}(x_*)\|^2$. The full variance is then given by

$$\sum_{m=1}^M \sigma_{m,*}^2 = \frac{1}{n}\sum_{m=1}^M \sum_{i=1}^n \|\nabla f_{mi}(x_*)\|^2 = \frac{N}{n}\sigma_*^2 = M\sigma_*^2.$$

where $\sigma_*^2 := \frac{1}{N}\sum_{i=1}^n \sum_{m=1}^M \|\nabla f_{mi}(x_*)\|^2$ is the variance of the gradients over all data.

Theorem 6. Let $R(x) \equiv 0$ (no prox) and the data be i.i.d., that is $\nabla F_m(x_*) = 0$ for any m, where x_* is the solution of (9). Let $\sigma_*^2 := \frac{1}{N}\sum_{i=1}^n \sum_{m=1}^M \|\nabla f_{mi}(x_*)\|^2$. If each f_{mj} is $L_{\max}$-smooth and μ-strongly convex, then the iterates of Algorithm 2 satisfy

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] \le (1-\gamma\mu)^{nT}\|x_0 - x_*\|^2 + \frac{\gamma^2 L_{\max} N \sigma_*^2}{M\mu}.$$

Observe that the guarantee given by Theorem 6 scales with the *effective number of functions* per machine N/M, similar to the scaling displayed by single-node RR.

Corollary 5.1. *Choose the stepsize* $\gamma > 0$ *as*

$$\gamma = \min\left(\frac{1}{L_{\max}}, \sqrt{\frac{\epsilon M \mu}{2 L_{\max} N \sigma_*^2}}\right),$$

and suppose that the total number of iterations $K = nT$ *satisfies*

$$K \ge \left(\frac{L_{\max}}{\mu} + \sqrt{\frac{2 L_{\max} N \sigma_*^2}{\epsilon M \mu^{3/2}}}\right) \log \frac{2\|x_0 - x_*\|^2}{\epsilon}.$$

Then $\mathbb{E}\left[\|x_T - x_*\|^2\right] \le \epsilon$.

In the small-accuracy regime, Theorem 5.1 shows that FedRR enjoys a convergence rate depending on $\frac{1}{\sqrt{\epsilon}}$ compared to the $\frac{1}{\epsilon}$ rate of convergence of FedAvg (Karimireddy et al., 2020).

6. Experiments[1]

ProxRR vs SGD. In Figure 1, we look at the logistic regression loss with the elastic net regularization,

$$\frac{1}{N}\sum_{i=1}^N f_i(x) + \lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|^2, \qquad (11)$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is defined as $f_i(x) := -\left(b_i \log\left(h(a_i^\top x)\right) + (1-b_i)\log\left(1 - h(a_i^\top x)\right)\right)$, and where $(a_i, b_i) \in \mathbb{R}^d \times \{0,1\}$, $i = 1, \ldots, N$ are the data samples, $h \colon t \to 1/(1+e^{-t})$ is the sigmoid function, and $\lambda_1, \lambda_2 \ge 0$ are parameters. We set minibatch sizes to 32 for all methods and use theoretical stepsizes, without any tuning. We denote the heuristic version of RR that performs proximal operator step after each iteration as 'RR (iteration prox)'. From the experiments, we can see that all methods behave more or less the same way. However, the algorithm that we propose needs only a small fraction of proximal operator evaluations, which gives it a huge advantage whenever the operator takes more time to compute than stochastic gradients.

FedRR vs Local SGD and Scaffold. We also compare the performance of FedRR, Local SGD and Scaffold (Karimireddy et al., 2020) on homogeneous (i.e., i.i.d.) and heterogeneous data. Since Local SGD and Scaffold require smaller stepsizes to converge, they are significantly slower in the i.i.d. regime, as can be seen in Figure 2. FedRR, however, does not need small initial stepsize and very quickly converges to a noisy neighborhood of the solution. We obtain heterogeneous regime by sorting data with respect to the labels and mixing the sorted dataset with the unsorted one. In this scenario, we also use the same small stepsize for every method to address the data heterogeneity. Clearly, Scaffold is the best in terms of functional values because it does variance reduction with respect to the data. Extending FedRR in the same way might be useful too, but this goes beyond the scope of our paper and we leave it for future work. We also note that in terms of distances from the optimum, FedRR still performs much better than Local SGD and Scaffold.

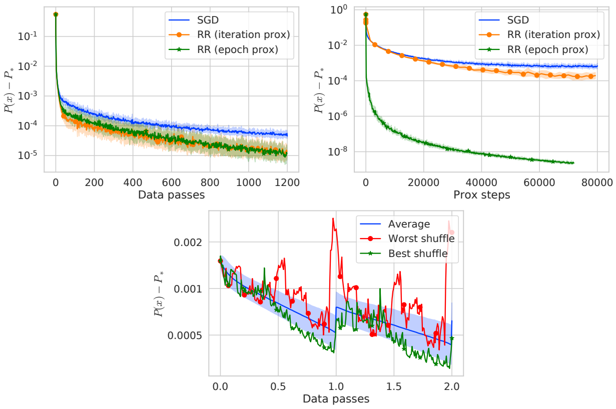

Figure 1. Experimental results for problem (11). The first two plots show with average and confidence intervals estimated on 20 random seeds and clearly demonstrate that one can save a lot of proximal operator computations with our method. The right plot shows the best/worst convergence of ProxSO over 20,000 sampled permutations.

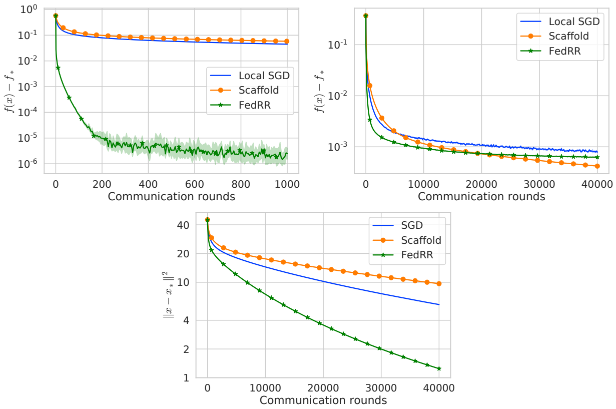

Figure 2. FedRR vs Local-SGD and Scaffold: i.i.d. data (left) and heterogeneous data (middle and right). We set $\lambda_1 = 0$ and estimate the averages and standard deviations by running 10 random seeds for each method.

[1] Our code is available on GitHub: https://github.com/konstmish/rr_prox_fed. More experimental details are in the appendix.

References

Ahn, K., Yun, C., and Sra, S. SGD with shuffling: optimal rates without component convexity and large epoch requirements. *arXiv preprint arXiv:2006.06946. Neural Information Processing Systems (NeurIPS) 2020*, 2020. (Cited on pages 2, 3, 4, 29, and 30)

Beck, A. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997. (Cited on page 4)

Bertsekas, D. P. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. In Sra, S., Nowozin, S., and Wright, S. J. (eds.), *Optimization for Machine Learning*, chapter 4. The MIT Press, 2011. ISBN 9780262298773. (Cited on page 2)

Bianchi, P., Hachem, W., and Iutzeler, F. A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. *IEEE Transactions on Automatic Control*, 61(10):2947–2957, 2015. (Cited on page 6)

Bordes, A., Bottou, L., and Gallinari, P. Sgd-qn: Careful quasi-newton stochastic gradient descent. *J. Mach. Learn. Res.*, 10:1737–1754, dec 2009. ISSN 1532-4435. (Cited on page 2)

Bottou, L. Curiously fast convergence of some stochastic gradient descent algorithms. Unpublished open problem offered to the attendance of the SLDS 2009 conference, 2009. URL http://leon.bottou.org/papers/bottou-slds-open-problem-2009. (Cited on page 2)

Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pp. 421–436. Springer, 2012. (Cited on page 2)

Chen, G. and Teboulle, M. Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993. doi: 10.1137/0803026. (Cited on page 18)

Davis, D. and Drusvyatskiy, D. Stochastic subgradient method converges at the rate $o(k^{-1/4})$ on weakly convex functions. *arXiv preprint*, abs/1802.02988, 2018. URL https://arXiv.org/abs/1802.02988. (Cited on page 6)

Drori, Y. and Shamir, O. The complexity of finding stationary points with stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2658–2667. PMLR, 2020. URL http://proceedings.mlr.press/v119/drori20a.html. (Cited on page 6)

Duchi, J. and Singer, Y. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934, 2009. (Cited on page 2)

Gorbunov, E., Hanzely, F., and Richtárik, P. A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of Machine Learning Research*, volume 108, pp. 680–690, Online, 26–28 Aug 2020. PMLR. (Cited on pages 2, 16, and 29)

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. SGD: General Analysis and Improved Rates. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR. (Cited on pages 3 and 4)

Gower, R. M., Richtárik, P., and Bach, F. Stochastic quasi-gradient methods: variance reduction via Jacobian sketching. *Mathematical Programming*, pp. 1–58, 2020. ISSN 0025-5610. doi: 10.1007/s10107-020-01506-0. (Cited on page 29)

Gürbüzbalaban, M., Özdağlar, A., and Parrilo, P. A. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, Oct 2019. ISSN 1436-4646. doi: 10.1007/s10107-019-01440-w. (Cited on page 2)

Haochen, J. and Sra, S. Random Shuffling Beats SGD after Finite Epochs. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2624–2633, Long Beach, California, USA, 09–15 Jun 2019. PMLR. (Cited on page 2)

Kairouz, P. et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. (Cited on pages 1 and 3)

Karimi, H., Nutini, J., and Schmidt, M. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Łojasiewicz Condition. In *European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 9851*, ECML PKDD 2016, pp. 795–811, Berlin, Heidelberg, 2016. Springer-Verlag. (Cited on page 4)

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S. U., and Suresh, A. T. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020. (Cited on pages 8 and 28)

Khaled, A. and Richtárik, P. Better theory for SGD in the nonconvex world. *arXiv Preprint arXiv:2002.03329*, 2020. (Cited on pages 5 and 30)

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for Local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020. (Cited on page 28)

Konečný, J., McMahan, H. B., Yu, F., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016. (Cited on pages 1 and 3)

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, 1999. (Cited on page 2)

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017. (Cited on pages 1 and 3)

Mishchenko, K., Khaled, A., and Richtárik, P. Random Reshuffling: Simple Analysis with Vast Improvements. *arXiv preprint arXiv:2006.05988. Neural Information Processing Systems (NeurIPS) 2020*, 2020. (Cited on pages 2, 3, 4, 5, 14, 17, and 19)

Nagaraj, D., Jain, P., and Netrapalli, P. SGD without Replacement: Sharper Rates for General Smooth Convex Functions. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4703–4711, Long Beach, California, USA, 09–15 Jun 2019. PMLR. (Cited on page 2)

Needell, D., Srebro, N., and Ward, R. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155(1): 549–573, Jan 2016. ISSN 1436-4646. doi: 10.1007/s10107-015-0864-7. (Cited on pages 4 and 29)

Parikh, N. and Boyd, S. Proximal Algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. (Cited on pages 14 and 28)

Patrascu, A. and Irofti, P. Stochastic proximal splitting algorithm for composite minimization. *Optimization Letters*, pp. 1–19, 2021. (Cited on page 4)

Pham, N. H., Nguyen, L. M., Phan, D. T., and Tran-Dinh, Q. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research*, 21(110):1–48, 2020. (Cited on page 2)

Recht, B. and Ré, C. Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. In Mannor, S., Srebro, N., and Williamson, R. C. (eds.), *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, pp. 11.1–11.24, 2012. Edinburgh, Scotland. (Cited on page 2)

Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. (Cited on page 2)

Safran, I. and Shamir, O. Random shuffling beats SGD only after many epochs on ill-conditioned problems. *arXiv preprint*, abs/2106.06880, 2021. URL https://arXiv.org/abs/2106.06880. (Cited on page 3)

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: from theory to algorithms*. Cambridge University Press, 2014. (Cited on page 1)

Shamir, O. Without-replacement sampling for stochastic gradient methods. In *Advances in neural information processing systems*, pp. 46–54, 2016. (Cited on page 2)

Shang, F., Jiao, L., Zhou, K., Cheng, J., Ren, Y., and Jin, Y. ASVRG: Accelerated Proximal SVRG. In Zhu, J. and Takeuchi, I. (eds.), *Proceedings of Machine Learning Research*, volume 95, pp. 815–830. PMLR, 14–16 Nov 2018. (Cited on page 2)

Stich, S. U. Unified Optimal Analysis of the (Stochastic) Gradient Method. *arXiv preprint arXiv:1907.04232*, 2019. (Cited on pages 4, 29, and 30)

Sun, R.-Y. Optimization for Deep Learning: An Overview. *Journal of the Operations Research Society of China*, 8 (2):249–294, Jun 2020. ISSN 2194-6698. doi: 10.1007/s40305-020-00309-6. (Cited on page 30)

Tang, J., Egiazarian, K., Golbabaee, M., and Davies, M. The practicality of stochastic optimization in imaging inverse problems. *IEEE Transactions on Computational Imaging*, 6:1471–1485, 2020. (Cited on pages 5 and 29)

Tibshirani, R. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. (Cited on page 2)

Tran, T. H., Nguyen, L. M., and Tran-Dinh, Q. Shuffling gradient-based methods with momentum. *arXiv preprint arXiv:2011.11884*, 2020. (Cited on page 30)

Vogel, C. *Computational methods for inverse problems*. Society for Industrial and Applied Mathematics, Philadelphia, 2002. ISBN 9780898715507. (Cited on page 2)

Woodworth, B., Patel, K. K., and Srebro, N. Minibatch vs Local SGD for Heterogeneous Distributed Learning. *arXiv preprint arXiv:2006.04735. Neural Information Processing Systems (NeurIPS) 2020*, 2020. (Cited on pages 3, 7, and 27)

Xiao, L. and Zhang, T. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014. doi: 10.1137/140961791. URL https://doi.org/10.1137/140961791. (Cited on page 5)

Yuan, H., Zaheer, M., and Reddi, S. Federated composite optimization. *arXiv preprint arXiv:2011.08474*, 2020. (Cited on page 7)

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006. (Cited on page 2)

Supplementary Material

Contents

Proofs

A. Basic notions and preliminaries

We say that an extended real-valued function $\phi\colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is proper if its domain, $\mathrm{dom}\ \phi := \{x : \phi(x) < +\infty\}$, is nonempty. We say that it is convex (resp. closed) if its epigraph, epi $\phi := \{(x,t) \in \mathbb{R}^d \times \mathbb{R} \ : \ \phi(x) \le t\}$, is a convex (resp. closed) set. Equivalently, ϕ is convex if $\mathrm{dom}\ \phi$ is a convex set and $\phi(\alpha x + (1-\alpha)y) \le \alpha\phi(x) + (1-\alpha)\phi(y)$ for all $x, y \in \mathrm{dom}\ \phi$ and $\alpha \in (0,1)$. Finally, ϕ is μ-strongly convex if $\phi(x) - \frac{\mu}{2}\|x\|^2$ is convex, and L-smooth if $\frac{L}{2}\|x\|^2 - \phi(x)$ is convex.

One useful fact that we will need is that for any vectors $a_1, \dots, a_M \in \mathbb{R}^d$ we have

$$\sum_{m=1}^{m} \|a_i\|^2 = \frac{1}{M}\left\|\sum_{m=1}^{M} a_m\right\|^2 + \sum_{m=1}^{m}\left\|a_m - \frac{1}{M}\sum_{l=1}^{M} a_l\right\|^2. \tag{12}$$

The identity above is sometimes called bias-variance decomposition.

To prove the upper bound in Theorem 1, we rely on a lemma due to Mishchenko et al. (2020) that bounds the variance when sampling without replacement.

Lemma 3 (Lemma 1 in (Mishchenko et al., 2020))**.** Let $X_1, \dots, X_n \in \mathbb{R}^d$ be fixed vectors, let $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ be their mean, and let $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \left\|X_i - \bar{X}\right\|^2$ be their variance. Fix any $i \in [n]$ and let $X_{\pi_0}, \dots, X_{\pi_{i-1}}$ be sampled uniformly without replacement from $\{X_1, \dots, X_n\}$ and $\bar{X}_\pi = \frac{1}{i}\sum_{j=0}^{i-1} X_{\pi_j}$ be their average. Then, the sample average and variance are given by

$$\mathbb{E}\left[\bar{X}_\pi\right] = \bar{X}, \qquad \mathbb{E}\left[\left\|\bar{X}_\pi - \bar{X}\right\|^2\right] = \tfrac{n-i}{i(n-1)}\sigma^2. \tag{13}$$

Finally, we define $[n] := \{1, 2, \dots, n\}$.

A.1. Bregman divergence

These notions have a more useful characterization in the case of real valued and continuously differentiable functions $\phi\colon \mathbb{R}^d \to \mathbb{R}$. The Bregman divergence of such ϕ is defined by $D_\phi(x,y) := \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle$. A continuously differentiable function ϕ is called μ-strongly convex if

$$\tfrac{\mu}{2}\|x-y\|^2 \le D_\phi(x,y), \qquad \forall x, y \in \mathbb{R}^d.$$

It is convex if this holds with $\mu = 0$. Moreover, a continuously differentiable function ϕ is called L-smooth if

$$-\tfrac{L}{2}\|x-y\|^2 \le D_\phi(x,y) \le \tfrac{L}{2}\|x-y\|^2, \qquad \forall x, y \in \mathbb{R}^d. \tag{14}$$

Note that the first inequality is redundant for convex ϕ because convexity implies $0 \le D_\phi(x,y)$.

A.2. Properties of the proximal operator

Before we proceed to the proofs of convergence, we should state some basic and well-known properties of the regularized objectives. The following lemma explains why the solution of (1) is a fixed point of the proximal-gradient step for *any* stepsize.

Lemma 4. Let Assumption 1 be satisfied.[2] Then point x_* is a minimizer of $P(x) = f(x) + \psi(x)$ if and only if for any $\gamma, b > 0$ we have

$$x_* = \mathrm{prox}_{\gamma b\psi}(x_* - \gamma b\nabla f(x_*)).$$

Proof. This follows by writing the first-order optimality conditions for problem (1), see (Parikh & Boyd, 2014, p.32) for a full proof. ■

[2] We only need the part about ψ.

The lemma above only shows that proximal-gradient step does not hurt if we are at the solution. In addition, we will rely on the following a bit stronger result which postulates that the proximal operator is a contraction (resp. strong contraction) if the regularizer ψ is convex (resp. strongly convex).

Lemma 5. Let Assumption 1 be satisfied.[3] If ψ is μ-strongly convex with $\mu \geq 0$, then for any $\gamma > 0$ we have

$$\|\mathrm{prox}_{\gamma n\psi}(x) - \mathrm{prox}_{\gamma n\psi}(y)\|^2 \leq \frac{1}{1+2\gamma\mu n}\|x-y\|^2, \tag{15}$$

for all $x, y \in \mathbb{R}^d$.

Proof. Let $u := \mathrm{prox}_{\gamma n\psi}(x)$ and $v := \mathrm{prox}_{\gamma n\psi}(y)$. By definition, $u = \mathrm{argmin}_w\{\psi(w) + \frac{1}{2\gamma n}\|w-x\|^2\}$. By first-order optimality, we have $0 \in \partial\psi(u) + \frac{1}{\gamma n}(u-x)$ or simply $x - u \in \gamma n\partial\psi(u)$. Using a similar argument for v, we get $x - u - (y-v) \in \gamma n(\partial\psi(u) - \partial\psi(v))$. Thus, by strong convexity of ψ, we get

$$\langle x-u-(y-v), u-v\rangle \geq \gamma\mu n\|u-v\|^2.$$

Hence,

$$\begin{aligned}\|x-y\|^2 &= \|u-v+(x-u-(y-v))\|^2 \\ &= \|u-v\|^2 + 2\langle x-u-(y-v), u-v\rangle + \|x-u-(y-v)\|^2 \\ &\geq \|u-v\|^2 + 2\langle x-u-(y-v), u-v\rangle \\ &\geq (1+2\gamma\mu n)\|u-v\|^2.\end{aligned}$$

■

B. Proof of Theorem 1 (Bounding the shuffling radius)

Proof. By the L_i-smoothness of f_i and the definition of x_*^i, we can replace the Bregman divergence in (4) with the bound

$$\begin{aligned}\mathbb{E}\left[D_{f_{\pi_i}}(x_*^i, x_*)\right] \overset{(14)}{\leq} \mathbb{E}\left[\frac{L_{\pi_i}}{2}\left\|x_*^i - x_*\right\|^2\right] &\leq \frac{L_{\max}}{2}\mathbb{E}\left[\|x_*^i - x_*\|^2\right] \\ &\overset{(3)}{=} \frac{\gamma^2 L_{\max}}{2}\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\nabla f_{\pi_j}(x_*)\right\|^2\right] \\ &= \frac{\gamma^2 L_{\max} i^2}{2}\mathbb{E}\left[\left\|\frac{1}{i}\sum_{j=0}^{i-1}\nabla f_{\pi_j}(x_*)\right\|^2\right] \\ &= \frac{\gamma^2 L_{\max} i^2}{2}\mathbb{E}\left[\left\|\bar{X}_\pi\right\|^2\right],\end{aligned} \tag{16}$$

where $\bar{X}_\pi = \frac{1}{j}\sum_{j=0}^{i-1} X_{\pi_j}$ with $X_j := \nabla f_j(x_*)$ for $j = 1, 2, \ldots, n$. Since $\bar{X} = \nabla f(x_*)$, by applying Lemma 3 we get

$$\mathbb{E}\left[\left\|\bar{X}_\pi\right\|^2\right] = \left\|\bar{X}\right\|^2 + \mathbb{E}\left[\left\|\bar{X}_\pi - \bar{X}\right\|^2\right] \overset{(13)+(5)}{=} \|\nabla f(x_*)\|^2 + \frac{n-i}{i(n-1)}\sigma_*^2. \tag{17}$$

It remains to combine (16) and (17), use the bounds $i^2 \leq n^2$ and $i(n-i) \leq \frac{n(n-1)}{2}$, which holds for all $i \in \{0, 1, \ldots, n-1\}$, and divide both sides of the resulting inequality by γ^2. ■

C. Proof of Convergence of Proximal SGD

Theorem 7 (Proximal SGD)**.** Let Assumption 1 hold. Further, suppose that either $f := \frac{1}{n}\sum_{i=1}^n f_i$ is μ-strongly convex or that ψ is μ-strongly convex. If Algorithm 3 is run with a constant stepsize $\gamma_k = \gamma > 0$ satisfying $\gamma \leq \frac{1}{2L_{\max}}$, then the final iterate after K steps satisfies

$$\mathbb{E}\left[\|x_K - x_*\|^2\right] \leq (1-\gamma\mu)^K \|x_0 - x_*\|^2 + \frac{2\gamma\sigma_*^2}{\mu}.$$

[3] We only need the part about ψ.

Algorithm 3 Proximal SGD

```
Input: Stepsizes $\gamma_k > 0$, initial vector $x_0 \in \mathbb{R}^d$, number of steps $K$
for steps $k = 0, 1, \ldots, K-1$ do
    Sample $i_k$ uniformly at random from $[n]$
    $x_{k+1} = \mathrm{prox}_{\gamma_k \psi}(x_k - \gamma_k \nabla f_{i_k}(x_k))$
end for
```

Proof. We will prove the case when ψ is μ-strongly convex. The other result follows as a straightforward special case of (Gorbunov et al., 2020, Theorem 4.1). We start by analyzing one step of SGD with stepsize $\gamma_k = \gamma$ and using Lemma 4

$$\begin{aligned}\|x_{k+1} - x_*\|^2 &= \left\|\mathrm{prox}_{\gamma\psi}(x_k - \gamma\nabla f_\xi(x_k)) - \mathrm{prox}_{\gamma\psi}(x_* - \gamma\nabla f(x_*))\right\|^2 \\ &\le \frac{1}{1+2\gamma\mu}\|x_k - \gamma\nabla f_\xi(x_k) - (x_* - \gamma\nabla f(x_*))\|^2.\end{aligned} \tag{18}$$

We may write the squared norm term in (18) as

$$\begin{aligned}\|x_k - \gamma\nabla f_\xi(x_k) - (x_* - \gamma\nabla f(x_*))\|^2 = \|x_k - x_*\|^2 - 2\gamma\left\langle x_k - x_*, \nabla f_\xi(x_k) - \nabla f(x_*)\right\rangle \\ + \gamma^2\|\nabla f_\xi(x_k) - \nabla f(x_*)\|^2.\end{aligned} \tag{19}$$

We denote by $\mathbb{E}_k[\cdot]$ expectation conditional on x_k. Note that the gradient estimate is conditionally unbiased, i.e., that $\mathbb{E}_k[\nabla f_\xi(x_k)] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_k) = \nabla f(x_k)$. Hence, taking conditional expectation in (19) and using unbiasedness we have

$$\begin{aligned}\mathbb{E}_k\left[\|x_k - \gamma\nabla f_\xi(x_k) - (x_* - \gamma\nabla f(x_*))\|^2\right] = \|x_k - x_*\|^2 - 2\gamma\left\langle x_k - x_*, \nabla f(x_k) - \nabla f(x_*)\right\rangle \\ + \gamma^2\mathbb{E}_k\left[\|\nabla f_\xi(x_k) - \nabla f(x_*)\|^2\right].\end{aligned} \tag{20}$$

By the convexity of f we have

$$\langle x_k - x_*, \nabla f(x_k) - \nabla f(x_*)\rangle \ge D_f(x_k, x_*).$$

Furthermore, we may estimate the third term in (20) by first using the fact that $\|x+y\|^2 \le 2\|x\|^2 + 2\|y\|^2$ for any two vectors $x, y \in \mathbb{R}^d$

$$\begin{aligned}\mathbb{E}_k\left[\|\nabla f_\xi(x_k) - \nabla f(x_*)\|^2\right] &\le 2\mathbb{E}_k\left[\|\nabla f_\xi(x_k) - \nabla f_\xi(x_*)\|^2\right] + 2\mathbb{E}_k\left[\|\nabla f_\xi(x_*) - \nabla f(x_*)\|^2\right] \\ &= 2\mathbb{E}_k\left[\|\nabla f_\xi(x_k) - \nabla f_\xi(x_*)\|^2\right] + 2\sigma_*^2.\end{aligned}$$

We now use that by the $L_{\max}$-smoothness of f_i we have that

$$\|\nabla f_i(x_k) - \nabla f_i(x_*)\|^2 \le 2L_{\max} \cdot D_{f_i}(x_k, x_*).$$

Hence

$$\begin{aligned}\mathbb{E}_k\left[\|\nabla f_\xi(x_k) - \nabla f_\xi(x_*)\|^2\right] &= \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x_k) - \nabla f_i(x_*)\|^2 \\ &\le \frac{2L_{\max}}{n}\sum_{i=1}^n \left[f_i(x_k) - f_i(x_*) - \langle\nabla f_i(x_*), x_k - x_*\rangle\right] \\ &= 2L_{\max}\left[f(x_k) - f(x_*) - \langle\nabla f(x_*), x_k - x_*\rangle\right] \\ &= 2L_{\max} D_f(x_k, x_*).\end{aligned} \tag{21}$$

Combining equations (20)–(21) we obtain

$$\mathbb{E}_k\left[\|x_k - \gamma\nabla f_\xi(x_k) - (x_* - \gamma\nabla f(x_*))\|^2\right] \le \|x_k - x_*\|^2 - 2\gamma\left(1 - 2\gamma L_{\max}\right) D_f(x_k, x_*) + 2\gamma^2\sigma_*^2.$$

Since $\gamma \leq \frac{1}{2L_{\max}}$ by assumption we have that $1 - 2\gamma L_{\max} \geq 0$. Since $D_f(x_k, x_*) \geq 0$ by the convexity of f we arrive at

$$\mathbb{E}_k\left[\left\|x_k - \gamma\nabla f_\xi(x_k) - (x_* - \gamma\nabla f(x_*))\right\|^2\right] \leq \|x_k - x_*\|^2 + 2\gamma^2\sigma_*^2.$$

Taking unconditional expectation and combining (43) with the last equation we have

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1} - x_*\|^2\right] &\leq \frac{1}{1+2\gamma\mu}\left(\mathbb{E}\left[\|x_k - x_*\|^2\right] + 2\gamma^2\sigma_*^2\right)\\ &= \frac{1}{1+2\gamma\mu}\mathbb{E}\left[\|x_k - x_*\|^2\right] + \frac{2\gamma^2\sigma_*^2}{1+2\gamma\mu}\\ &\leq \frac{1}{1+2\gamma\mu}\mathbb{E}\left[\|x_k - x_*\|^2\right] + 2\gamma^2\sigma_*^2.\end{aligned}$$

To simplify this further, we use that for any $x \leq \frac{1}{2}$ we have that $\frac{1}{1+2x} \leq 1 - x$ and that $\gamma\mu \leq \frac{\mu}{2L_{\max}} \leq \frac{1}{2}$, hence

$$\mathbb{E}\left[\|x_{k+1} - x_*\|^2\right] \leq (1 - \gamma\mu)\,\mathbb{E}\left[\|x_k - x_*\|^2\right] + 2\gamma^2\sigma_*^2.$$

Recursing the above inequality for K steps yields

$$\begin{aligned}\mathbb{E}\left[\|x_K - x_*\|^2\right] &\leq (1-\gamma\mu)^K\|x_0 - x_*\|^2 + 2\gamma^2\sigma_*^2\left(\sum_{k=0}^{K-1}(1-\gamma\mu)^k\right)\\ &\leq (1-\gamma\mu)^K\|x_0 - x_*\|^2 + 2\gamma^2\sigma_*^2\left(\sum_{k=0}^{\infty}(1-\gamma\mu)^k\right)\\ &= (1-\gamma\mu)^K\|x_0 - x_*\|^2 + \frac{2\gamma\sigma_*^2}{\mu}.\end{aligned}$$

■

Furthermore, by choosing the stepsize γ as $\gamma = \min\left\{\frac{1}{2L_{\max}}, \frac{\varepsilon\mu}{4\sigma_*^2}\right\}$, we get that $\mathbb{E}\left[\|x_K - x_*\|^2\right] = \mathcal{O}(\varepsilon)$ provided that the number of iterations is at least

$$K_{\mathrm{SGD}} \geq \left(\kappa + \frac{\sigma_*^2}{\varepsilon\mu^2}\right)\log\left(\frac{2r_0}{\varepsilon}\right),$$

which we previously stated in (7).

D. Proofs of Theorem 2 and Theorem 8

D.1. A key lemma for shuffling-based methods

The intermediate limit points x_*^i are extremely important for showing tight convergence guarantees for Random Reshuffling even without proximal operator. The following lemma illustrates that by giving a simple recursion, whose derivation follows (Mishchenko et al., 2020, Proof of Theorem 1). The proof is included for completeness.

Lemma 6 (Theorem 1 in (Mishchenko et al., 2020))**.** Suppose that each f_i is L_i-smooth and λ-strongly convex (where $\lambda = 0$ means each f_i is just convex). Then the inner iterates generated by Algorithm 1 satisfy

$$\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \leq (1-\gamma\lambda)\,\mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] - 2\gamma(1-\gamma L_{\max})\,\mathbb{E}\left[D_{f_{\pi_i}}(x_t^i, x_*)\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2, \tag{22}$$

where x_*^i is as in (3), $i = 0, 1, \ldots, n-1$, and x_* is any minimizer of P.

Proof. By definition of x_t^{i+1} and x_*^{i+1}, we have

$$\begin{aligned}\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] = \mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] - 2\gamma\mathbb{E}\left[\langle\nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_*), x_t^i - x_*^i\rangle\right]\\ + \gamma^2\mathbb{E}\left[\left\|\nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_*)\right\|^2\right].\end{aligned} \tag{23}$$

Note that the third term in (23) can be bounded as

$$\left\|\nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_*)\right\|^2 \le 2L_{\max} \cdot D_{f_{\pi_i}}(x_t^i, x_*). \tag{24}$$

We may rewrite the second term in (23) using the three-point identity (Chen & Teboulle, 1993, Lemma 3.1) as

$$\left\langle \nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_*), x_t^i - x_*^i \right\rangle = D_{f_{\pi_i}}(x_*^i, x_t^i) + D_{f_{\pi_i}}(x_t^i, x_*) - D_{f_{\pi_i}}(x_*^i, x_*). \tag{25}$$

Combining (23), (24), and (25) we obtain

$$\begin{aligned}\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \le \mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] - 2\gamma \cdot \mathbb{E}\left[D_{f_{\pi_i}}(x_*^i, x_t^i)\right] + 2\gamma \cdot \mathbb{E}\left[D_{f_{\pi_i}}(x_*^i, x_*)\right] \\ - 2\gamma\left(1 - \gamma L_{\max}\right)\mathbb{E}\left[D_{f_{\pi_i}}(x_t^i, x_*)\right].\end{aligned} \tag{26}$$

Using λ-strong convexity of f_{π_i}, we derive

$$\frac{\lambda}{2}\left\|x_t^i - x_*^i\right\|^2 \le D_{f_{\pi_i}}(x_*^i, x_t^i). \tag{27}$$

Furthermore, by the definition of shuffling radius (Definition 1), we have

$$\mathbb{E}\left[D_{f_{\pi_i}}(x_*^i, x_*)\right] \le \max_{i=0,\ldots,n-1} \mathbb{E}\left[D_{f_{\pi_i}}(x_*^i, x_*)\right] = \gamma^2\sigma_{\text{rad}}^2. \tag{28}$$

Using (27) and (28) in (26) yields (22). ■

D.2. Proof of Theorem 2

Proof. Starting with Lemma 6 with $\lambda = \mu$, we have

$$\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \le (1-\gamma\mu)\,\mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] - 2\gamma\left(1 - \gamma L_{\max}\right)\mathbb{E}\left[D_{f_{\pi_i}}(x_t^i, x_*)\right] + 2\gamma^3\sigma_{\text{rad}}^2.$$

Since $D_{f_\pi}(x_t^i, x_*)$ is a Bregman divergence of a convex function, it is nonnegative. Combining this with the fact that the stepsize satisfies $\gamma \le {}^1/_{L_{\max}}$, we have

$$\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \le (1-\gamma\mu)\,\mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] + 2\gamma^3\sigma_{\text{rad}}^2.$$

Unrolling this recursion for n steps, we get

$$\begin{aligned}\mathbb{E}\left[\left\|x_t^n - x_*^n\right\|^2\right] &\le (1-\gamma\mu)^n\,\mathbb{E}\left[\left\|x_t^0 - x_*^0\right\|^2\right] + 2\gamma^3\sigma_{\text{rad}}^2\left(\sum_{j=0}^{n-1}(1-\gamma\mu)^j\right) \\ &= (1-\gamma\mu)^n\,\mathbb{E}\left[\left\|x_t - x_*\right\|^2\right] + 2\gamma^3\sigma_{\text{rad}}^2\left(\sum_{j=0}^{n-1}(1-\gamma\mu)^j\right),\end{aligned} \tag{29}$$

where we used the fact that $x_t^0 - x_*^0 = x_t - x_*$. Since x_* minimizes P, we have by Lemma 4 that

$$x_* = \operatorname{prox}_{\gamma n\psi}\left(x_* - \gamma\sum_{i=0}^{n-1}\nabla f_{\pi_i}(x_*)\right) = \operatorname{prox}_{\gamma n\psi}\left(x_*^n\right).$$

Moreover, by Lemma 5 we obtain that

$$\left\|x_{t+1} - x_*\right\|^2 = \left\|\operatorname{prox}_{\gamma n\psi}(x_t^n) - \operatorname{prox}_{\gamma n\psi}(x_*^n)\right\|^2 \le \left\|x_t^n - x_*^n\right\|^2.$$

Using this in (29) yields

$$\mathbb{E}\left[\left\|x_{t+1} - x_*\right\|^2\right] \le (1-\gamma\mu)^n\,\mathbb{E}\left[\left\|x_t - x_*\right\|^2\right] + 2\gamma^3\sigma_{\text{rad}}^2\left(\sum_{j=0}^{n-1}(1-\gamma\mu)^j\right).$$

We now unroll this recursion again for T steps

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] \leq (1-\gamma\mu)^{nT}\,\mathbb{E}\left[\|x_0 - x_*\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2 \left(\sum_{j=0}^{n-1}(1-\gamma\mu)^j\right)\left(\sum_{i=0}^{T-1}(1-\gamma\mu)^{ni}\right). \tag{30}$$

Following Mishchenko et al. (2020), we rewrite and bound the product in the last term as

$$\begin{aligned}\left(\sum_{j=0}^{n-1}(1-\gamma\mu)^j\right)\left(\sum_{i=0}^{T-1}(1-\gamma\mu)^{ni}\right) &= \sum_{j=0}^{n-1}\sum_{i=0}^{T-1}(1-\gamma\mu)^{ni+j} \\ &= \sum_{k=0}^{nT-1}(1-\gamma\mu)^k \\ &\leq \sum_{k=0}^{\infty}(1-\gamma\mu)^k \quad = \frac{1}{\gamma\mu}.\end{aligned}$$

It remains to plug this bound into (30). ■

D.3. Theorem 8 and its proof

Theorem 8. Let Assumption 1 hold and $f_1, \ldots, f_n$ be convex. Further, assume that ψ is μ-strongly convex. If Algorithm 1 is run with constant stepsize $\gamma_t = \gamma \leq 1/L_{\max}$, where $L_{\max} = \max_i L_i$, then its iterates satisfy

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] \leq (1+2\gamma\mu n)^{-T}\|x_0 - x_*\|^2 + \frac{\gamma^2\sigma_{\mathrm{rad}}^2}{\mu}.$$

Proof. Starting with Lemma 6 with $\lambda = 0$, we have

$$\mathbb{E}\left[\|x_t^{i+1} - x_*^{i+1}\|^2\right] \leq \mathbb{E}\left[\|x_t^i - x_*^i\|^2\right] - 2\gamma\left(1-\gamma L_{\max}\right)\mathbb{E}\left[D_{f_{\pi_i}}(x_t^i, x_*)\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2.$$

Since $\gamma \leq 1/L_{\max}$ and $D_{f_\pi}(x_t^i, x_*)$ is nonnegative we may simplify this to

$$\mathbb{E}\left[\|x_t^{i+1} - x_*^{i+1}\|^2\right] \leq \mathbb{E}\left[\|x_t^i - x_*^i\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2.$$

Unrolling this recursion over an epoch we have

$$\mathbb{E}\left[\|x_t^n - x_*^n\|^2\right] \leq \mathbb{E}\left[\|x_t^0 - x_*^0\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2 n = \mathbb{E}\left[\|x_t - x_*\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2 n. \tag{31}$$

Since x_* minimizes P, we have by Lemma 4 that

$$x_* = \mathrm{prox}_{\gamma n\psi}\left(x_* - \gamma\sum_{i=0}^{n-1}\nabla f_{\pi_i}(x_*)\right) = \mathrm{prox}_{\gamma n\psi}\left(x_*^n\right).$$

Hence, $x_{t+1} - x_* = \mathrm{prox}_{\gamma n\psi}(x_t^n) - \mathrm{prox}_{\gamma n\psi}(x_*^n)$. We may now use Lemma 5 to get

$$(1+2\gamma\mu n)\,\mathbb{E}\left[\|x_{t+1} - x_*\|^2\right] \leq \mathbb{E}\left[\|x_t^n - x_*^n\|^2\right].$$

Combining this with (31), we obtain

$$\mathbb{E}\left[\|x_{t+1} - x_*\|^2\right] \leq \frac{1}{1+2\gamma\mu n}\mathbb{E}\left[\|x_t - x_*\|^2\right] + \frac{2\gamma^3\sigma_{\mathrm{rad}}^2 n}{1+2\gamma\mu n}.$$

We may unroll this recursion again, this time for T steps, and then use that $\sum_{j=1}^{T-1}(1+2\gamma\mu n)^{-j} \le \sum_{j=1}^{\infty}(1+2\gamma\mu n)^{-j} = 1/(2\gamma\mu n)$:

$$\begin{aligned}\mathbb{E}\left[\|x_T - x_*\|^2\right] &\le (1+2\gamma\mu n)^{-T}\,\mathbb{E}\left[\|x_0 - x_*\|^2\right] + \frac{2\gamma^3\sigma_{\mathrm{rad}}^2 n}{1+2\gamma\mu n}\left(\sum_{j=0}^{T-1}(1+2\gamma\mu n)^{-j}\right)\\ &= (1+2\gamma\mu n)^{-T}\,\mathbb{E}\left[\|x_0 - x_*\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2 n\left(\sum_{j=1}^{T}(1+2\gamma\mu n)^{-j}\right)\\ &\le (1+2\gamma\mu n)^{-T}\,\mathbb{E}\left[\|x_0 - x_*\|^2\right] + 2\gamma^3\sigma_{\mathrm{rad}}^2 n\frac{1}{2\gamma\mu n}\\ &= (1+2\gamma\mu n)^{-T}\,\mathbb{E}\left[\|x_0 - x_*\|^2\right] + \frac{\gamma^2\sigma_{\mathrm{rad}}^2}{\mu}. \qquad \blacksquare\end{aligned}$$

Using Theorem 8 and choosing the stepsize as

$$\gamma = \min\left\{\tfrac{1}{L_{\max}}, \tfrac{\sqrt{\varepsilon\mu}}{\sigma_{\mathrm{rad}}}\right\}, \tag{32}$$

we get $\mathbb{E}\left[\|x_T - x_*\|^2\right] = \mathcal{O}(\varepsilon)$ provided that the total number of iterations satisfies

$$K \ge \left(\kappa + \tfrac{\sigma_{\mathrm{rad}}/\mu}{\sqrt{\varepsilon\mu}} + n\right)\log\left(\tfrac{2r_0}{\varepsilon}\right). \tag{33}$$

This can be converted to a bound similar to (3.2) by using Theorem 1, in which case the only difference between the two cases is an extra $n\log\left(\frac{1}{\varepsilon}\right)$ term when only the regularizer ψ is μ-strongly convex. Since for small enough accuracies the $1/\sqrt{\varepsilon}$ term dominates, this difference is minimal.

E. Nonconvex analysis

E.1. A key lemma

For notational convenience, we define

$$g_t := \frac{1}{\gamma n}(x_t - x_t^n) = \frac{1}{n}\sum_{i=0}^{n-1}\nabla f_{\pi_i}(x_t^i),$$

which is equivalent to $x_t^n = x_t - \gamma n g_t$.

Lemma 7. Let functions $f_1, \ldots, f_n$ be $L_{\max}$-smooth, Assumptions 2 and 3 be satisfied and $\gamma \le \frac{1}{2L_{\max}n}$. Then,

$$\mathbb{E}_t\left[\|\nabla f(x_t) - g_t\|^2\right] \le \gamma^2 L_{\max}^2 n^2 (\|\mathcal{G}_{\gamma n}(x_t)\|^2 + \zeta^2) + \gamma^2 L_{\max}^2 n\sigma^2. \tag{34}$$

Proof. We start with the observation that gradient Lipschitzness reduces the left-hand side to a difference of iterates:

$$\begin{aligned}\|\nabla f(x_t) - g_t\|^2 &= \left\|\frac{1}{n}\sum_{i=0}^{n-1}\left[\nabla f_{\pi_i}(x_t) - \nabla f_{\pi_i}(x_t^i)\right]\right\|^2\\ &\le \frac{1}{n}\sum_{i=0}^{n-1}\left\|\nabla f_{\pi_i}(x_t) - \nabla f_{\pi_i}(x_t^i)\right\|^2\\ &\le \frac{1}{n}\sum_{i=0}^{n-1}L_{\max}^2\|x_t - x_t^i\|^2.\end{aligned}$$

Define $V_t := \sum_{i=0}^{n-1}\|x_t^i - x_t\|^2$. Clearly, it is sufficient to bound $\mathbb{E}\left[V_t\right]$ to finish the proof. Also note that for any intermediate iterate x_t^k within epoch t we do not use proximal step, so the following identity holds:

$$x_t^k = x_t - \gamma\sum_{i=0}^{k-1}\nabla f_{\pi_i}(x_t^i).$$

This identity only includes gradients, so to bound the deviation of x_t^k from x_t we apply Jensen's inequality and gradient Lipschitzness

$$\begin{aligned}
\mathbb{E}_t\left[\|x_t^k - x_t\|^2\right] &= \gamma^2 \mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \nabla f_{\pi_i}(x_t^i)\right\|^2\right] \\
&\le 2\gamma^2 \mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \left(\nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_t)\right)\right\|^2\right] + 2\gamma^2 \mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \nabla f_{\pi_i}(x_t)\right\|^2\right] \\
&\le 2\gamma^2 k \sum_{i=0}^{k-1} \mathbb{E}_t\left[\left\|\nabla f_{\pi_i}(x_t^i) - \nabla f_{\pi_i}(x_t)\right\|^2\right] + 2\gamma^2 \mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \nabla f_{\pi_i}(x_t)\right\|^2\right] \\
&\le 2\gamma^2 L_{\max}^2 k \sum_{i=0}^{k-1} \mathbb{E}_t\left[\|x_t^i - x_t\|^2\right] + 2\gamma^2 \mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \nabla f_{\pi_i}(x_t)\right\|^2\right].
\end{aligned}$$

Now we are going to use the fact that for any i in RR we have $\mathbb{E}_t\left[\nabla f_{\pi_i}(x_t)\right] = \nabla f(x_t)$. Note that this property does not hold if x_t is not independent of π_i, which is why the result does not hold for SO. Let us also define $\sigma_t^2 := \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x_t) - \nabla f(x_t)\|^2$. By Lemma 3 we have

$$\begin{aligned}
\mathbb{E}_t\left[\left\|\sum_{i=0}^{k-1} \nabla f_{\pi_i}(x_t)\right\|^2\right] &= k^2\|\nabla f(x_t)\|^2 + k^2 \mathbb{E}_t\left[\left\|\frac{1}{k}\sum_{i=0}^{k-1}(\nabla f_{\pi_i}(x_t) - \nabla f(x_t))\right\|^2\right] \\
&\overset{(13)}{=} k^2\|\nabla f(x_t)\|^2 + \frac{k(n-k)}{n-1}\sigma_t^2.
\end{aligned}$$

Plugging this back and using Assumption 3, we derive

$$\begin{aligned}
\mathbb{E}_t\left[\left\|x_t^k - x_t\right\|^2\right] &\le 2\gamma^2 L_{\max}^2 k \sum_{i=0}^{k-1} \mathbb{E}_t\left[\left\|x_t^i - x_t\right\|^2\right] + 2\gamma^2 k^2 \|\nabla f(x_t)\|^2 + 2\gamma^2 \frac{k(n-k)}{n-1}\sigma^2 \\
&\le 2\gamma^2 L_{\max}^2 k \mathbb{E}\left[V_t\right] + 2\gamma^2 k^2 \|\nabla f(x_t)\|^2 + 2\gamma^2 \frac{k(n-k)}{n-1}\sigma^2.
\end{aligned}$$

Let us use the obtained bound on a single iterate distance $\mathbb{E}_t\left[\left\|x_t^k - x_t\right\|^2\right]$ to upper bound $\mathbb{E}\left[V_t\right]$:

$$\begin{aligned}
\mathbb{E}_t\left[V_t\right] &= \sum_{i=0}^{n-1} \mathbb{E}_t\left[\|x_t^i - x_t\|^2\right] \\
&\le \gamma^2 L_{\max}^2 n(n-1)\mathbb{E}_t\left[V_t\right] + \frac{1}{3}\gamma^2(n-1)n(2n-1)\|\nabla f(x_t)\|^2 + \frac{1}{3}\gamma^2 n(n+1)\sigma^2.
\end{aligned}$$

This inequality has $\mathbb{E}_t\left[V_t\right]$ in both sides, so we can rearrange it and use the assumption $\gamma \le \frac{1}{2L_{\max} n}$, which results in

$$\begin{aligned}
\mathbb{E}_t\left[V_t\right] &\le \frac{4}{3}(1 - \gamma^2 L_{\max}^2 n(n-1))\mathbb{E}_t\left[V_t\right] \\
&\le \frac{4}{9}\gamma^2(n-1)n(2n-1)\|\nabla f(x_t)\|^2 + \frac{4}{9}\gamma^2 n(n+1)\sigma^2 \\
&\le \gamma^2 n^3 \|\nabla f(x_t)\|^2 + \gamma^2 n^2 \sigma^2.
\end{aligned}$$

To conclude the proof, apply Assumption 2 to $x_t \in \mathrm{dom}(\psi)$ and plug-in the bound on $\mathbb{E}_t\left[V_t\right]$ into the bound on $\mathbb{E}_t\left[\|\nabla f(x_t) - g_t\|^2\right]$. ■

E.2. Proof of Theorem 3

Proof. Let us introduce

$$w_t := \operatorname{prox}_{\gamma n \psi}(x_t - \gamma n \nabla f(x_t)).$$

The idea of our proof is to first obtain a descent recursion for $P(w_t)$ and then bound $P(x_{t+1}) - P(w_t)$.

By convexity of ψ, we have for any $g \in \partial \psi(w_t)$

$$\psi(w_t) \le \psi(x_t) + \langle g, w_t - x_t \rangle.$$

Furthermore, the definition of w_t implies by first-order optimality that $x_t - \gamma n \nabla f(x_t) - w_t \in \gamma n \partial \psi(w_t)$, so we can plug it into the bound above to get

$$\begin{aligned} \psi(w_t) &\le \psi(x_t) + \frac{1}{\gamma n}\langle x_t - \gamma n \nabla f(x_t) - w_t, w_t - x_t \rangle \\ &= \psi(x_t) - \langle \nabla f(x_t), w_t - x_t \rangle - \frac{1}{\gamma n}\|w_t - x_t\|^2. \end{aligned}$$

At the same time, by $L_{\max}$-smoothness of f we have

$$f(w_t) \le f(x_t) + \langle \nabla f(x_t), w_t - x_t \rangle + \frac{L_{\max}}{2}\|w_t - x_t\|^2.$$

Adding the two recursion together yields

$$P(w_t) = f(x_t) + \psi(w_t) \le P(x_t) + \left(\frac{L_{\max}}{2} - \frac{1}{\gamma n}\right)\|w_t - x_t\|^2.$$

Now we shall upper bound $P(x_{t+1})$. Using the convexity of ψ for $x_t^n - x_{t+1} \in \gamma n \partial \psi(x_{t+1})$, we derive

$$\begin{aligned} \psi(x_{t+1}) &\le \psi(w_t) + \frac{1}{\gamma n}\langle x_t^n - x_{t+1}, x_{t+1} - w_t \rangle = \psi(w_t) - \langle g_t, x_{t+1} - w_t \rangle + \frac{1}{\gamma n}\langle x_t - x_{t+1}, x_{t+1} - w_t \rangle \\ &= \psi(w_t) - \langle g_t, x_{t+1} - w_t \rangle + \frac{1}{2\gamma n}\left(\|x_t - w_t\|^2 - \|x_t - x_{t+1}\|^2 - \|x_{t+1} - w_t\|^2\right). \end{aligned}$$

Next, we apply $L_{\max}$-smoothness of f two times, to upper bound $D_f(x_{t+1}, x_t)$ and to lower bound $D_f(w_t, x_t)$:

$$\begin{aligned} & f(x_{t+1}) \le f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L_{\max}}{2}\|x_{t+1} - x_t\|^2, \\ \text{and} \qquad & f(x_t) \le f(w_t) + \langle \nabla f(x_t), x_t - w_t \rangle + \frac{L_{\max}}{2}\|x_t - w_t\|^2. \end{aligned}$$

Therefore,

$$f(x_{t+1}) \le f(w_t) + \langle \nabla f(x_t), x_{t+1} - w_t \rangle + \frac{L_{\max}}{2}\left(\|x_{t+1} - x_t\|^2 + \|w_t - x_t\|^2\right).$$

Combining the inequalities for $\psi(x_{t+1})$ and $f(x_{t+1})$, we obtain

$$\begin{aligned} P(x_{t+1}) \le P(w_t) + \langle \nabla f(x_t) - g_t, x_{t+1} - w_t \rangle + \left(\frac{L_{\max}}{2} - \frac{1}{2\gamma n}\right)\|x_{t+1} - x_t\|^2 \\ + \left(\frac{L_{\max}}{2} + \frac{1}{2\gamma n}\right)\|x_t - w_t\|^2 - \frac{1}{2\gamma n}\|x_{t+1} - w_t\|^2. \end{aligned}$$

By Young's inequality and Lemma 7 we have

$$\begin{aligned} & \mathbb{E}_t\left[\langle \nabla f(x_t) - g_t, x_{t+1} - w_t \rangle\right] \\ & \le \mathbb{E}_t\left[\frac{\gamma n}{2}\|\nabla f(x_t) - g_t\|^2 + \frac{2}{\gamma n}\|x_{t+1} - w_t\|^2\right] \\ & \overset{(34)}{\le} \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^3}{2}\|\mathcal{G}_{\gamma n}(x_t)\|^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 + \frac{2}{\gamma n}\mathbb{E}_t\left[\|x_{t+1} - w_t\|^2\right]. \end{aligned}$$

If we plug this back, the term $\|x_{t+1} - w_t\|^2$ will cancel out, giving us for $\gamma \le \frac{1}{L_{\max} n}$

$$\begin{aligned}
&\mathbb{E}_t\left[P(x_{t+1})\right] \\
&\le P(w_t) + \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^3}{2}\|\mathcal{G}_{\gamma n}(x_t)\|^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 + \left(\frac{L_{\max}}{2} + \frac{1}{2\gamma n}\right)\|x_t - w_t\|^2 \\
&\quad + \left(\frac{L_{\max}}{2} - \frac{1}{2\gamma n}\right)\mathbb{E}_t\left[\|x_{t+1} - x_t\|^2\right] \\
&\le P(w_t) + \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^3}{2}\|\mathcal{G}_{\gamma n}(x_t)\|^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 + \left(\frac{L_{\max}}{2} + \frac{1}{2\gamma n}\right)\|x_t - w_t\|^2.
\end{aligned}$$

Using the recursion for $P(w_t)$ and our choice $\gamma \le \frac{1}{5L_{\max} n}$, we finally obtain, after plugging-in $\|x_t - w_t\|^2 = \gamma^2 n^2 \mathcal{G}_{\gamma n}(x_t)$,

$$\begin{aligned}
&\mathbb{E}_t\left[P(x_{t+1})\right] \\
&\le P(x_t) + \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 + \left(\frac{\gamma n L_{\max}^2}{2} + \frac{L_{\max}}{2} + \frac{1}{2\gamma n} + \frac{L_{\max}}{2} - \frac{1}{\gamma n}\right)\gamma^2 n^2 \|\mathcal{G}_{\gamma n}(x_t)\|^2 \\
&\le P(x_t) + \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 + \left(\frac{L_{\max}}{10} + L_{\max} - \frac{1}{2\gamma n}\right)\gamma^2 n^2 \|\mathcal{G}_{\gamma n}(x_t)\|^2 \\
&\le P(x_t) + \frac{\gamma^3 L_{\max}^2 n^3}{2}\zeta^2 + \frac{\gamma^3 L_{\max}^2 n^2}{2}\sigma^2 - \frac{1}{4\gamma n}\gamma^2 n^2 \|\mathcal{G}_{\gamma n}(x_t)\|^2.
\end{aligned}$$

Recursing this to $P(x_0)$ and using $P_* \le P(x_T)$, we get the Theorem's claim. ■

F. Proofs for federated learning

F.1. Lemma for the extended proximal operator

Lemma 8. Let ψ_C be the consensus constraint and R be a closed convex proximable function. Suppose that $x_1, x_2, \ldots, x_M$ are all in $\mathbb{R}^d$. Then,

$$\operatorname{prox}_{\gamma(R+\psi_C)}(x_1, \ldots, x_M) = \operatorname{prox}_{\frac{\gamma}{M}R}(\overline{x}),$$

where $\overline{x} = \frac{1}{M}\sum_{m=1}^M x_m$.

Proof. We have,

$$\operatorname{prox}_{\gamma(R+\psi_C)}(x_1, \ldots, x_M) = \begin{pmatrix} \operatorname{prox}_{\frac{\gamma}{M}R}(\overline{x}) \\ \vdots \\ \operatorname{prox}_{\frac{\gamma}{M}R}(\overline{x}) \end{pmatrix} \quad \text{with} \quad \overline{x} = \frac{1}{M}\sum_{m=1}^M x_m.$$

This is a simple consequence of the definition of the proximal operator. Indeed, the result of $\operatorname{prox}_{\gamma(R+\psi_C)}$ must have blocks equal to some vector z such that

$$\begin{aligned}
z &= \operatorname*{argmin}_x \left\{\gamma R(x) + \frac{1}{2}\sum_{m=1}^M \|x - x_m\|^2\right\} \\
&= \operatorname*{argmin}_x \left\{\gamma R(x) + \frac{1}{2}\sum_{m=1}^M \left(\|x - \overline{x}\|^2 + 2\langle x - \overline{x}, \overline{x} - x_m\rangle) + \|\overline{x} - x_m\|^2\right)\right\} \\
&= \operatorname*{argmin}_x \left\{\gamma R(x) + \frac{1}{2}M\|x - \overline{x}\|^2\right\} \qquad = \operatorname{prox}_{\frac{\gamma}{M}R}(\overline{x}).
\end{aligned}$$

■

F.2. Proof of Lemma 1

Proof. Given some vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d\cdot M}$, let us use their block representation $\boldsymbol{x} = (x_1^\top, \ldots, x_M^\top)^\top$, $\boldsymbol{y} = (y_1^\top, \ldots, y_M^\top)^\top$. Since we use the Euclidean norm, we have

$$\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\|^2 = \sum_{m=1}^{M} \|\nabla f_{mi}(x_m) - \nabla f_{mi}(y_m)\|^2 \le \sum_{m=1}^{M} L_i^2 \|x_m - y_m\|^2 = L_i^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

We can obtain a lower bound by doing the same derivation and applying strong convexity instead of smoothness:

$$\sum_{m=1}^{M} \|\nabla f_{mi}(x_m) - \nabla f_{mi}(y_m)\|^2 \ge \mu^2 \sum_{m=1}^{M} \|x_m - y_m\|^2 = \mu^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Thus, we have $\mu\|\boldsymbol{x} - \boldsymbol{y}\| \le \|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \le L_i \|\boldsymbol{x} - \boldsymbol{y}\|$, which is exactly μ-strong convexity and L_i-smoothness of f_i. ■

F.3. Proof of Lemma 2

Proof. By Theorem 1 we have

$$\sigma_{\mathrm{rad}}^2 \le \frac{L_{\max}}{2}\Big(n^2 \|\nabla f(\boldsymbol{x}_*)\|^2 + \frac{n}{2}\sigma_*^2\Big).$$

Due to the separable structure of f, we have for the variance term

$$n\sigma_*^2 := \sum_{i=1}^{n} \|\nabla f_i(\boldsymbol{x}_*) - \nabla f(\boldsymbol{x}_*)\|^2 = \sum_{i=1}^{n}\sum_{m=1}^{M} \left\|\nabla f_{mi}(x_*) - \frac{1}{n}\nabla F_m(x_*)\right\|^2.$$

The expression inside the summation is not exactly the variance due to the different normalization: $\frac{1}{n}$ instead of $\frac{1}{N_m}$. Nevertheless, we can expand the norm and try to get the actual variance:

$$\begin{aligned}
\sum_{i=1}^{n} \left\|\nabla f_{mi}(x_*) - \frac{1}{n}\nabla F_m(x_*)\right\|^2 &= \sum_{i=1}^{N_m}\left(\left\|\nabla f_{mi}(x_*) - \frac{1}{N_m}\nabla F_m(x_*)\right\|^2 + \Big(\frac{1}{N_m} - \frac{1}{n}\Big)^2 \|\nabla F_m(x_*)\|^2\right) \\
&\quad + 2\sum_{i=1}^{N_m}\Big\langle \nabla f_{mi}(x_*) - \frac{1}{N_m}\nabla F_m(x_*), \Big(\frac{1}{N_m} - \frac{1}{n}\Big)\nabla F_m(x_*)\Big\rangle \\
&= N_m \sigma_{m,*}^2 + N_m\Big(\frac{1}{N_m} - \frac{1}{n}\Big)^2 \|\nabla F_m(x_*)\|^2 \\
&\le n\sigma_{m,*}^2 + \|\nabla F_m(x_*)\|^2.
\end{aligned}$$

Moreover, the gradient term has the same block structure, so

$$n^2\|\nabla f(\boldsymbol{x}_*)\|^2 = n^2 \left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_*)\right\|^2 = \sum_{m=1}^{M}\left\|\sum_{i=1}^{n}\nabla f_{mi}(x_*)\right\|^2 = \sum_{m=1}^{M}\|\nabla F_m(x_*)\|^2.$$

Plugging the last two bounds back inside the upper bound on σ_{rad}^2, we deduce the lemma's statement. ■

F.4. Proof of Theorem 4

Proof. Since we assume that $N_1 = \cdots = N_M = n$, we have $\frac{N}{M} = n$ and the strong convexity constant of $\psi = \frac{N}{n}(R + \psi_C)$ is equal to $\frac{N}{n}\cdot\frac{\mu}{M} = \mu$. By applying Theorem 8 we obtain

$$\mathbb{E}\left[\|\boldsymbol{x}_T - \boldsymbol{x}_*\|^2\right] \le (1 + 2\gamma\mu n)^{-T}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|^2 + \frac{\gamma^2\sigma_{\mathrm{rad}}^2}{\mu}.$$

Since $\boldsymbol{x}_T = \mathrm{prox}_{\gamma N(R+\psi_C)}(\boldsymbol{x}_{T-1}^n)$, we have $\boldsymbol{x}_T \in C$, i.e., all of its blocks are equal to each other and we have $\boldsymbol{x}_T = (x_T^\top, \ldots, x_T^\top)^\top$. Since we use the Euclidean norm, it also implies

$$\mathbb{E}\left[\|\boldsymbol{x}_T - \boldsymbol{x}_*\|^2\right] = M\|x_T - x_*\|^2.$$

The same is true for $\boldsymbol{x}_0$, so we need to divide both sides of the upper bound on $\|\boldsymbol{x}_T - \boldsymbol{x}_*\|^2$ by M. Doing so together with applying Lemma 2 yields

$$
\begin{aligned}
\mathbb{E}\left[\|x_T - x_*\|^2\right] &\le (1+2\gamma\mu n)^{-T}\|x_0 - x_*\|^2 + \frac{\gamma^2\sigma_{\mathrm{rad}}^2}{M\mu} \\
&\le (1+2\gamma\mu n)^{-T}\|x_0 - x_*\|^2 + \frac{\gamma^2 L_{\max}}{M\mu}\sum_{m=1}^{M}\left(\|\nabla F_m(x_*)\|^2 + \frac{n}{4}\sigma_{m,*}^2\right) \\
&= (1+2\gamma\mu n)^{-T}\|x_0 - x_*\|^2 + \frac{\gamma^2 L_{\max}}{M\mu}\sum_{m=1}^{M}\left(\|\nabla F_m(x_*)\|^2 + \frac{N}{4M}\sigma_{m,*}^2\right).
\end{aligned}
$$

■

F.5. Proof of Theorem 6

Proof. According to Lemma 1, each f_i is μ-strongly convex and $L_{\max}$-smooth, so we obtain the result by trivially applying Theorem 2 and upper bounding σ_{rad}^2 the same way as in the proof of Theorem 4. ■

F.6. Proposition 1 and its proof

An important property of Assumption 2 is that it is equivalent to the bounded dissimilarity assumption that was previously used for the nonconvex analysis of Local SGD. We formalize this in the following proposition.

Proposition 1. Consider federated learning reformulation (10). If $\psi \equiv \psi_C$, i.e., $R \equiv 0$, then Assumption 2 with constant $\overline{\zeta}^2 := M\zeta^2$ is equivalent to ζ-bounded dissimilarity (Assumption 4):

$$
\frac{1}{M}\sum_{m=1}^{M}\left\|\nabla F_m(x) - \frac{1}{M}\sum_{l=1}^{M}\nabla F_l(x)\right\|^2 \le \zeta^2.
$$

Proof. First, observe that if $\boldsymbol{x} \in \mathrm{dom}(\psi)$, then $\boldsymbol{x}$ has all blocks equal to some $x \in \mathbb{R}^d$, $\boldsymbol{x} = (x^\top, \dots, x^\top)^\top$. Therefore, for the objective in reformulation (10) and $\boldsymbol{x} \in \mathrm{dom}(\psi)$, we have

$$
\begin{aligned}
\nabla f(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{M}\nabla f_{mi}(\boldsymbol{x}) = \frac{1}{n}\sum_{m=1}^{M}\sum_{i=1}^{n}\nabla f_{mi}(\boldsymbol{x}) \\
&= \frac{1}{n}\sum_{m=1}^{M}F_m(\boldsymbol{x}) = \begin{pmatrix} \frac{1}{n}\nabla F_1(x_1) \\ \vdots \\ \frac{1}{n}\nabla F_M(x_M)\end{pmatrix} = \begin{pmatrix} \frac{1}{n}\nabla F_1(x) \\ \vdots \\ \frac{1}{n}\nabla F_M(x)\end{pmatrix}.
\end{aligned}
\tag{35}
$$

With the help of bias-variance decomposition, the left-hand side of Assumption 2 can be written as

$$
\begin{aligned}
\|\nabla f(\boldsymbol{x})\|^2 &\overset{(35)}{=} \frac{1}{n^2}\sum_{m=1}^{M}\|\nabla F_m(x)\|^2 \\
&\overset{(12)}{=} \frac{1}{Mn^2}\left\|\sum_{m=1}^{M}\nabla F_m(x)\right\|^2 + \frac{1}{n^2}\sum_{m=1}^{M}\left\|\nabla F_m(x) - \frac{1}{M}\sum_{l=1}^{M}\nabla F_l(x)\right\|^2.
\end{aligned}
$$

Let us now work out the proximal-gradient mapping. According to Lemma 8, the proximal operator of ψ is simply the averaging of all blocks, while the full gradient is given in (35), which give when combined

$$
\mathrm{prox}_{\gamma n\psi}(\boldsymbol{x} - \gamma n\nabla f(\boldsymbol{x})) = \begin{pmatrix} \frac{1}{M}\sum_{m=1}^{M}(x - \gamma\nabla F_m(x)) \\ \vdots \\ \frac{1}{M}\sum_{m=1}^{M}(x - \gamma\nabla F_m(x))\end{pmatrix}.
\tag{36}
$$

Therefore,

$$\begin{aligned}\|\mathcal{G}_{\gamma n}(\boldsymbol{x})\|^2 &= \frac{1}{\gamma^2 n^2}\|\boldsymbol{x} - \mathrm{prox}_{\gamma n \psi}(\boldsymbol{x} - \gamma n \nabla f(\boldsymbol{x}))\|^2 \\ &\overset{(36)}{=} \frac{1}{\gamma^2 n^2}\sum_{l=1}^{M}\left\|x - \frac{1}{M}\sum_{m=1}^{M}(x - \gamma \nabla F_m(x))\right\|^2 = \frac{M}{n^2}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla F_m(x)\right\|^2. \qquad (37)\end{aligned}$$

Having the expressions for both sides, we can write

$$\|\nabla f(\boldsymbol{x})\|^2 = \|\mathcal{G}_{\gamma n}(\boldsymbol{x})\|^2 + \sum_{m=1}^{M}\left\|\frac{1}{n}\nabla F_m(x) - \frac{1}{N}\sum_{l=1}^{M}\nabla F_l(x)\right\|^2.$$

From this expression and the fact $\frac{1}{N}\sum_{l=1}^{M}\nabla F_l(x) = \nabla F(x)$, it is easy to see the equivalence. ■

F.7. Proof of Theorem 5

The federated learning reformulation (10) has different constant scaling than the finite-sum federated learning problem (9), and the only constant that does not change at all is $L_{\max}$. For the initial error $\overline{\delta}_0$ of the reformulation we have

$$\overline{\delta}_0 = \frac{N}{n}\delta_0 = M\delta_0,$$

where $\delta_0 := \frac{1}{N}\sum_{m=1}^{M}F_m(x_0) - \min_x \frac{1}{N}\sum_{m=1}^{M}F_m(x)$ and we use only consider the simplified case $N_1 = \cdots = N_M = n$ so $\frac{N}{n} = M$. For the variance, we have

$$\overline{\sigma}^2 = \sup_{\boldsymbol{x}}\mathbb{E}\left[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2\right] = \sup_{\boldsymbol{x}}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\nabla f_{mi}(x_m) - \frac{1}{n}\nabla F_m(x_m)\right\|^2\right] = M\sigma^2.$$

As we derived in (37), the proximal-gradient mapping norm is equal to

$$\mathbb{E}\left[\|\mathcal{G}_{\gamma n}(\boldsymbol{x})\|^2\right] = \frac{M}{n^2}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla F_m(x)\right\|^2 = M\left\|\frac{1}{N}\sum_{m=1}^{M}\nabla F_m(x)\right\|^2 = M\|\nabla F(x)\|^2,$$

so to have $\|\nabla F(x_T)\|^2 \le \varepsilon^2$, we need $\mathbb{E}\left[\|\mathcal{G}_{\gamma n}(\boldsymbol{x}_T)\|^2\right] \le \overline{\varepsilon}^2 := M\varepsilon^2$. In addition, notice that by Proposition 1 the constant from Assumption 2 is $\overline{\zeta} = \sqrt{M}\zeta$.

Thus, Theorem 3 implies, if we ignore $L_{\max}$, that we need

$$T = \mathcal{O}\left(\frac{\overline{\delta}_0}{\overline{\varepsilon}^2} + \frac{\overline{\delta}_0\overline{\sigma}}{\sqrt{n}\overline{\varepsilon}^3} + \frac{\overline{\delta}_0\overline{\zeta}}{\overline{\varepsilon}^3}\right) = \mathcal{O}\left(\frac{\delta_0}{\varepsilon^2} + \frac{\delta_0\sigma}{\sqrt{n}\varepsilon^3} + \frac{\delta_0\zeta}{\varepsilon^3}\right)$$

communication rounds to achieve $\min_{t=0,\ldots,T-1}\mathbb{E}\left[\|\nabla F(x_T)\|^2\right] = \mathcal{O}(\varepsilon^2)$.

G. Comparison of FedRR with other algorithms for federated learning

G.1. Heterogeneous Data

In this section we compare between FedRR and several known baseline algorithms for Federated Learning. In particular, we consider the following algorithms:

1. Distributed gradient descent (DGD)
2. Local SGD (with M nodes and n local steps per node)

To be clear, the problem we are considering is

$$\min_{x\in\mathbb{R}^d} f(x) := \left[\frac{1}{M}\sum_{m=1}^{M} F_m(x) + R(x)\right],$$

where each objective f_m can be written as

$$F_m(x) = \frac{1}{n}\sum_{i=1}^{n} f_{m,i}(x).$$

We further assume that each objective is L-smooth and convex, and that R is μ-strongly convex. This implies that f is L-smooth and μ-strongly convex. Note that this is a special case of (9) where we keep $N_1 = N_2 = \ldots = n$ for simplicity.

Corollary 1. Let $c^2 = \zeta_*^2 + \frac{n}{4}\sigma_*^2$, where $\zeta_*^2 := \frac{1}{M}\sum_{m=1}^{M}\|\nabla F_m(x)\|^2$ and $\sigma_*^2 = \frac{1}{M}\sum_{m=1}^{M}\|\nabla F(x_*) - \nabla F_m(x_*)\|^2$. Then the communication complexity required by FedRR to reach an ϵ-accurate solution is

$$T = \Omega\left(\left(\frac{\kappa}{n} + \frac{c}{\mu n}\sqrt{\frac{\kappa}{\epsilon}}\right)\log\left(\frac{r_0}{\epsilon}\right)\right), \tag{38}$$

where $r_0 = \|x_0 - x_*\|^2$.

Proof. This is a straightforward consequence of Theorem 4. ■

G.1.1. Distributed gradient descent

When we compute n gradients on each node per communication round, we are essentially running distributed gradient descent (DGD). In order to reach an ϵ-accurate solution, DGD requires the following number of iterations

$$T = \Omega\left(\kappa\log\left(\frac{r_0}{\epsilon}\right)\right).$$

Comparing against the result of Corollary 1, we see that FedRR is better whenever the accuracy ϵ satisfies

$$\frac{1}{\mu L}\left(\frac{\zeta_*^2}{n^2} + \frac{\sigma_*^2}{n}\right) = \frac{c^2}{\mu n^2 L} < \epsilon.$$

Note that this guarantee grows more rigorous with increasing levels of heterogeneity– this has been observed for other local methods as well, such as Local SGD (Woodworth et al., 2020).

G.1.2. Local SGD

The best current lower bound for Local SGD is given by (Woodworth et al., 2020) in the *stochastic* case. By stochastic case, we mean that the problem considered is

$$\min_{x\in\mathbb{R}^d} \mathbb{E}_{\xi\sim\mathcal{D}}\left[f_\xi(x)\right].$$

This is a more general problem than the finite-sum minimization problem (1) and is usually strictly harder to solve (i.e., requires more iterations to achieve an ϵ-accurate solution). We are not aware of any analysis of Local SGD specifically for the finite-sum problem, and thus we specialize the result of Woodworth et al. (2020) anyway. For Local SGD on μ-strongly convex and L smooth functions, and with n steps of local steps per node, the lower bound they give after T communication rounds is

$$\min\left(\Delta\exp\left(-\frac{\mu T}{L}\right), \frac{L\zeta_*^2}{\mu^2 T^2}\right) + \frac{\sigma^2}{\mu M n T} + \min\left(\Delta, \frac{L\sigma^2}{\mu^2 n^2 T^2}\right), \tag{39}$$

where σ^2 is a uniform bound on the variance (i.e., $\mathbb{E}[\|\nabla f_\xi(x) - f(x)\|^2] \le \sigma^2$ for all $x \in \mathbb{R}^d$), ζ_*^2 is defined as in Corollary 1, and Δ is an upper bound on $f(x_0) - f_*$. We note that this lower bound is *not* actually met by any of the existing analysis for Local SGD. Even ignoring the dependence on σ (which may not be tight because this is the stochastic case), the first term (i.e., the "optimization term") in (39) scales with κ when T is large and $\frac{\sqrt{\kappa}\zeta_*}{\sqrt{\mu\epsilon}}$ when T is small. This is clearly worse than (38) for large n.

H. Further experimental details

Implementation details. For each i, we have $L_i = \frac{1}{4}\|a_i\|$. For the ℓ_1-regularized problem, we set $\lambda_2 = 3 \cdot 10^{-5} \cdot L$ and tune λ_1 to obtain a solution with about 25% zero coordinates, which gives $\lambda_1 = 5 \cdot 10^{-5}$. We use stepsizes decreasing as $\mathcal{O}(\frac{1}{t})$ for all methods. We use the 'w8a' dataset[4] for the experiment with ℓ_1 regularization.

Proximal operator calculation. It is well-known (see, for instance, (Parikh & Boyd, 2014)) that the proximal operator for $\psi(x) = \lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|^2$ is given by

$$\mathrm{prox}_{\gamma\psi}(x) = \frac{1}{1+\gamma\lambda_2}\mathrm{prox}_{\gamma\lambda_1\|\cdot\|_1}(x),$$

where the j-th coordinate of $\mathrm{prox}_{\gamma\lambda_1\|\cdot\|_1}(x)$ is

$$[\mathrm{prox}_{\gamma\lambda_1\|\cdot\|_1}(x)]_j = \begin{cases} \mathrm{sign}([x]_j)(|[x]_j| - \gamma\lambda_1), & \text{if } |[x]_j| \geq \gamma\lambda_1, \\ 0, & \text{otherwise.} \end{cases}$$

Federated experiments. The experiments for the comparison of FedRR, Local SGD and Scaffold use no ℓ_1 regularization and $\lambda_2 = 10^{-5} \cdot L$. To make comparison fair, all methods use n local steps. For FedRR, the initial stepsize was $\frac{1}{L}$ in the i.i.d. regime and $\frac{1}{Ln}$ in the heterogeneous regime. As per Theorem 3 in (Khaled et al., 2020), the stepsizes for Local SGD must satisfy $\gamma_t = \mathcal{O}(1/(LH))$, where H is the number of local steps, a similar result holds for Scaffold (Karimireddy et al., 2020). The parallelization of local runs is done using the Ray package[5]. We use the 'w8a' dataset for the i.i.d. experiment. For the heterogeneous experiment, we sort 'a9a' dataset with respect to the target labels $b \in \{0, 1\}$ and then mix it with the original order in proportion 2:1. For all methods, the local workers used minibatch size 16. Exact implementation can be found in our code.

[4] The datasets were downloaded from LibSVM https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

[5] https://ray.io/

Extensions

Here we discuss two extensions of our theory that significantly matter in practice: using decreasing stepsizes and applying importance resampling.

I. Extension: Importance resampling

Suppose that each f_i is L_i-smooth. Then the iteration complexities of both SGD and RR depend on $L_{\max}/\mu$, where $L_{\max}$ is the maximum smoothness constant among the smoothness constants $L_1, L_2, \dots, L_n$. The maximum smoothness constant can be arbitrarily worse than the average smoothness constant $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$. This situation is in contrast to the complexity of gradient descent which depends on the smoothness constant L_f of $f = \frac{1}{n}\sum_{i=1}^n f_i$, for which we have $L_f \le \bar{L}$. This is a problem commonly encountered with stochastic optimization methods and may cause significantly degraded performance in practical optimization tasks in comparison with deterministic methods (Tang et al., 2020).

Importance sampling is a common technique to improve the convergence of SGD (Algorithm 3): we sample function $(\bar{L}/L_i) f_i$ with probability p_i proportional to L_i, where $\bar{L} := \frac{1}{n}\sum_{i=1}^n L_i$. In that case, the SGD update is still unbiased since

$$\mathbb{E}_i\left[\frac{\bar{L}}{L_i} f_i\right] = \sum_{i=1}^n p_i \frac{\bar{L}}{L_i} f_i = f.$$

Moreover, the smoothness of function $(\bar{L}/L_i) f_i$ is $\bar{L}$ for any i, so the guarantees would depend on $\bar{L}$ instead of $\max_{i=1,\dots,n} L_i$. Importance sampling successfully improves the iteration complexity of SGD to depend on $\bar{L}$ (Needell et al., 2016), and has been investigated in a wide variety of settings (Gower et al., 2020; Gorbunov et al., 2020).

Importance sampling is a neat technique but it relies heavily on the fact that we use *unbiased* sampling. How can we obtain a similar result if inside any permutation the sampling is biased? The answer requires us to think again as to what happens when we replace f_i with $(\bar{L}/L_i) f_i$. To make sure the problem remains the same, it is sufficient to have $(\bar{L}/L_i) f_i$ inside a permutation exactly $L_i/\bar{L}$ times. And since $L_i/\bar{L}$ is not necessarily integer, we should use $n_i = \lceil L_i/\bar{L} \rceil$ and solve

$$\min_{x \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^n \underbrace{\left(\frac{1}{n_i} f_i(x) + \dots + \frac{1}{n_i} f_i(x)\right)}_{n_i \text{ times}} + \psi(x), \tag{40}$$

where

$$N := n_1 + \dots + n_n = \left\lceil \frac{L_1}{\bar{L}} \right\rceil + \dots + \left\lceil \frac{L_n}{\bar{L}} \right\rceil.$$

Clearly, this problem is equivalent to the original formulation in (1). At the same time, we have improved all smoothness constants to $\bar{L}$. It might seem that that the new problem has more functions, but it turns out that the new number of functions satisfies $N \le 2n$, so any related costs, such as longer loops or storing duplicates of the data, are negligible, as the next theorem shows.

Theorem 9. For every i, assume that each f_i is convex and L_i-smooth, and let ψ be μ-strongly convex. Then, the number of functions N in (40) satisfies $N \le 2n$, and Algorithm 1 applied to problem (40) has the same complexity as (33) but proportional to $\bar{L}$ rather than $L_{\max}$.

Proof. We show that $N \le 2n$ as the rest of the theorem's claim trivially follows from Theorem 8. Firstly, note that for any number $a \in \mathbb{R}$ we have $\lceil a \rceil \le a + 1$. Therefore,

$$N = \sum_{i=1}^n \left\lceil \frac{L_i}{\bar{L}} \right\rceil \le \sum_{i=1}^n \left(\frac{L_i}{\bar{L}} + 1\right) = n + \sum_{i=1}^n \frac{L_i}{\bar{L}} = 2n. \qquad \blacksquare$$

J. Extension: Decreasing stepsizes

Using the theoretical stepsize (32) requires knowing the desired accuracy ε ahead of time as well as estimating σ_{rad}. It also results in extra polylogarithmic factors in the iteration complexity (33), a phenomenon observed and fixed by using decreasing stepsizes in both vanilla RR (Ahn et al., 2020) and in SGD (Stich, 2019).

We show that we can adopt the same technique to our setting. However, we depart from the stepsize scheme of Ahn et al. (2020) by only varying the stepsize *once per epoch* rather than every iteration. This is closer to the common practical heuristic of decreasing the stepsize once every epoch or once every few epochs (Sun, 2020; Tran et al., 2020). The stepsize scheme we use is inspired by the schemes of (Stich, 2019; Khaled & Richtárik, 2020): in particular, we fix $T > 0$, let $t_0 = \lceil T/2 \rceil$, and choose the stepsizes $\gamma_t > 0$ by

$$\gamma_t = \begin{cases} \frac{1}{L_{\max}} & \text{if } T \leq \frac{L_{\max}}{2\mu n} \text{ or } t \leq t_0, \\ \frac{7}{\mu n (s+t-t_0)} & \text{if } T > \frac{L_{\max}}{2\mu n} \text{ and } t > t_0, \end{cases} \tag{41}$$

where $s := 7L_{\max}/(4\mu n)$. Hence, we fix the stepsize used in the first $T/2$ iterations and then start decreasing it every epoch afterwards. Using this stepsize schedule, we can obtain the following convergence guarantee when each f_i is smooth and convex and the regularizer ψ is μ-strongly convex.

Theorem 10. Suppose that each f_i is $L_{\max}$-smooth and convex, and that the regularizer ψ is μ-strongly convex. Fix $T > 0$. Then choosing stepsizes γ_t according to (41) we have that $\gamma_t \leq 1/L_{\max}$ for all t and the final iterate generated by Algorithm 1 satisfies

$$\mathbb{E}\left[\|x_T - x_*\|^2\right] = \mathcal{O}\left(\exp\left(-\frac{nT}{\kappa+2n}\right) r_0 + \frac{\sigma_{\mathrm{rad}}^2}{\mu^3 n^2 T^2}\right),$$

where $\kappa := L_{\max}/\mu$, $r_0 := \|x_0 - x_*\|^2$ and $\mathcal{O}(\cdot)$ hides absolute (non-problem-specific) constants.

This guarantee holds for any number of epochs $T > 0$. We believe a similar guarantee can be obtained in the case each f_i is strongly-convex and the regularizer ψ is just convex, but we did not include it as it adds little to the overall message.

In the rest of the section we provide a proof of Theorem 10.

J.1. A recursion Lemma

We first state and prove the following algorithm-independent lemma. This lemma plays a key role in the proof of Theorem 10 and is heavily inspired by the stepsize schemes of Stich (2019) and Khaled & Richtárik (2020) and their proofs.

Lemma 9. Suppose that there exist constants $a, b, c \geq 0$ such that for all $\gamma_t \leq \frac{1}{b}$ we have

$$(1 + \gamma_t a n) r_{t+1} \leq r_t + \gamma_t^3 c. \tag{42}$$

Fix $T > 0$. Let $t_0 = \lceil \frac{T}{2} \rceil$. Then choosing stepsizes $\gamma_t > 0$ by

$$\gamma_t = \begin{cases} \frac{1}{b}, & \text{if } t \leq t_0 \text{ or } T \leq \frac{b}{an}, \\ \frac{7}{an(s+t-t_0)} & \text{if } t > t_0 \text{ and } T > \frac{b}{an}, \end{cases}$$

where $s = \frac{7b}{2an}$. Then

$$r_T \leq \exp\left(-\frac{nT}{2\left(b/a + n\right)}\right) r_0 + \frac{1421c}{a^3 n^3 T^2}.$$

Proof. If $T \leq \frac{7b}{an}$, then we have $\gamma_t = \gamma = \frac{1}{b}$ for all t. Hence recursing we have,

$$r_T \leq (1 + \gamma a n)^{-T} r_0 + \frac{\gamma^3 c}{\gamma a n} = (1 + \gamma a n)^{-T} r_0 + \frac{\gamma^2 c}{an}.$$

Note that $\frac{1}{1+x} \leq \exp(-\frac{x}{1+x})$ for all x, hence

$$r_T \leq \exp\left(-\frac{\gamma a n T}{1 + \gamma a n}\right) r_0 + \frac{\gamma^2 c}{an}$$

Substituting for γ yields

$$r_T \leq \exp\left(-\frac{nT}{b/a + n}\right) r_0 + \frac{c}{b^2 a n}.$$

Note that by assumption we have $\frac{1}{b} \le \frac{7}{Tan}$, hence

$$r_T \le \exp\left(-\frac{nT}{b/a+n}\right) r_0 + \frac{49c}{T^2a^3n^3}. \tag{43}$$

If $T > \frac{7b}{an}$, then we have for the first phase when $t \le t_0$ with stepsize $\gamma_t = \frac{1}{b}$ that

$$r_{t_0} \le \exp\left(-\frac{nt_0}{b/a+n}\right) r_0 + \frac{c}{b^2an} \le \exp\left(-\frac{nT}{2(b/a+n)}\right) r_0 + \frac{c}{b^2an}. \tag{44}$$

Then for $t > t_0$ we have

$$(1+\gamma_t an)\, r_{t+1} \le r_t + \gamma_t^3 c = r_t + \frac{7^3 c}{a^3n^3\left(s+t-t_0\right)^3}.$$

Multiplying both sides by $(s+t-t_0)^3$ yields

$$\left(s+t-t_0\right)^3 (1+\gamma_t an)\, r_{t+1} \le \left(s+t-t_0\right)^3 r_t + \frac{7^3c}{a^3n^3}. \tag{45}$$

Note that because t and t_0 are integers and $t > t_0$, we have that $t - t_0 \ge 1$ and therefore $s+t-t_0 \ge 1$. We may use this to lower bound the multiplicative factor in the left hand side of (45) as

$$\begin{aligned}
\left(s+t-t_0\right)^3 (1+\gamma_t an) &= \left(s+t-t_0\right)^3 \left(1+\frac{7}{s+t-t_0}\right) \\
&= \left(s+t-t_0\right)^3 + 7\left(s+t-t_0\right)^2 \\
&= \left(s+t-t_0\right)^3 + 3\left(s+t-t_0\right)^2 + 3\left(s+t-t_0\right)^2 + \left(s+t-t_0\right)^2 \\
&\ge \left(s+t-t_0\right)^3 + 3\left(s+t-t_0\right)^2 + 3\left(s+t-t_0\right) + 1 \\
&= \left(s+t+1-t_0\right)^3.
\end{aligned} \tag{46}$$

Using (46) in (45) we obtain

$$\left(s+t+1-t_0\right)^3 r_{t+1} \le \left(s+t-t_0\right)^3 r_t + \frac{7^3c}{a^3n^3}.$$

Let $w_t = \left(s+t-t_0\right)^3$. Then we can rewrite the last inequality as

$$w_{t+1}r_{t+1} - w_t r_t \le \frac{7^3c}{a^3n^3}.$$

Summing up and telescoping from $t = t_0$ to T yields

$$w_T r_T \le w_{t_0} r_{t_0} + \frac{7^3c}{a^3n^3}\left(T-t_0\right).$$

Note that $w_{t_0} = s^3$ and $w_T = \left(s+T-t_0\right)^3$. Hence,

$$\begin{aligned}
r_T &\le \frac{s^3}{\left(s+T-t_0\right)^3} r_{t_0} + \frac{7^3c}{a^3n^3\left(s+T-t_0\right)^2}\frac{T-t_0}{s+T-t_0} \\
&\le \frac{s^3}{\left(s+T-t_0\right)^3} r_{t_0} + \frac{7^3c}{a^3n^3\left(s+T-t_0\right)^2}.
\end{aligned}$$

Since we have $s+T-t_0 \ge T-t_0 \ge T/2$, it holds

$$r_T \le \frac{8s^3}{T^3} r_{t_0} + \frac{4\cdot 7^3c}{a^3n^3T^2}. \tag{47}$$

The bound in (44) can be rewritten as

$$\frac{s^3}{T^3} r_{t_0} \le \frac{s^3}{T^3}\exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{s^3c}{b^2anT^3}.$$

We now rewrite the last inequality, use that $T > 2s$ and further use the fact that $s = \frac{7b}{2an}$:

$$
\begin{aligned}
\frac{s^3}{T^3} r_{t_0} &\leq \underbrace{\left(\frac{s}{T}\right)^3}_{\leq 1/8} \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{s^2 c}{b^2 a n T^2} \underbrace{\left(\frac{s}{T}\right)}_{\leq 1/2} \\
&\leq \frac{1}{8} \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{s^2 c}{2b^2 a n T^2} \\
&= \frac{1}{8} \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{7^2 c}{8 a^3 n^3 T^2}. \qquad (48)
\end{aligned}
$$

Plugging in the estimate of (48) into (47) we obtain

$$
\begin{aligned}
r_T &\leq \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{7^2 c}{a^3 n^3 T^2} + \frac{4 \cdot 7^3 c}{a^3 n^3 T^2} \\
&= \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{1421 c}{a^3 n^3 T^2}. \qquad (49)
\end{aligned}
$$

Taking the maximum of (43) and (49) we see that for any $T > 0$ we have

$$
r_T \leq \exp\left(-\frac{nT}{2\left(b/a+n\right)}\right) r_0 + \frac{1421 c}{a^3 n^3 T^2}. \qquad \blacksquare
$$

J.2. Proof of Theorem 10

Proof. Start with Lemma 6 with $\lambda = 0$, $L = L_{\max}$, and $\gamma = \gamma_t$,

$$
\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \leq \mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] - 2\gamma\left(1 - \gamma L_{\max}\right) \mathbb{E}\left[D_{f_{\pi_i}}(x_t^i, x_*)\right] + 2\gamma_t^3 \sigma_{\mathrm{rad}}^2.
$$

Since $\gamma \leq {}^1/_{L_{\max}}$ and $D_{f_\pi}(x_t^i, x_*)$ is nonnegative we may simplify this to

$$
\mathbb{E}\left[\left\|x_t^{i+1} - x_*^{i+1}\right\|^2\right] \leq \mathbb{E}\left[\left\|x_t^i - x_*^i\right\|^2\right] + 2\gamma_t^3 \sigma_{\mathrm{rad}}^2.
$$

Unrolling this recursion for n steps we get

$$
\mathbb{E}\left[\left\|x_t^n - x_*^n\right\|^2\right] \leq \mathbb{E}\left[\left\|x_t^0 - x_*^0\right\|^2\right] + 2n\gamma_t^3 \sigma_{\mathrm{rad}}^2.
$$

By Lemma 5 we have

$$
(1 + 2\gamma_t \mu n)\, \mathbb{E}\left[\left\|x_{t+1} - x_*\right\|^2\right] \leq \mathbb{E}\left[\left\|x_t - x_*\right\|^2\right] + 2n\gamma_t^3 \sigma_{\mathrm{rad}}^2.
$$

We may then use Lemma 9 to obtain that

$$
\begin{aligned}
\mathbb{E}\left[\left\|x_T - x_*\right\|^2\right] &\leq \exp\left(-\frac{nT}{2(L_{\max}/\mu + n)}\right) \left\|x_0 - x_*\right\|^2 + \frac{356 \sigma_{\mathrm{rad}}^2}{\mu^3 n^2 T^2} \\
&= \mathcal{O}\left(\exp\left(-\frac{nT}{\kappa + 2n}\right) \left\|x_0 - x_*\right\|^2 + \frac{\sigma_{\mathrm{rad}}^2}{\mu^3 n^2 T^2}\right). \qquad \blacksquare
\end{aligned}
$$