Convergence of Distributed Stochastic Variance Reduced Methods Without Sampling Extra Data

Shicong Cen, Huishuai Zhang, Yuejie Chi, Wei Chen, and Tie-Yan Liu

Abstract—Stochastic variance reduced methods have gained a lot of interest recently for empirical risk minimization due to its appealing run time complexity. When the data size is large and disjointly stored on different machines, it becomes imperative to distribute the implementation of such variance reduced methods. In this paper, we consider a general framework that directly distributes popular stochastic variance reduced methods in the master/slave model, by assigning outer loops to the parameter server, and inner loops to worker machines. This framework is natural and friendly to implement, but its theoretical convergence is not well understood. We obtain a comprehensive understanding of algorithmic convergence with respect to data homogeneity by measuring the smoothness of the discrepancy between the local and global loss functions. We establish the linear convergence of distributed versions of a family of stochastic variance reduced algorithms, including those using accelerated and recursive gradient updates, for minimizing strongly convex losses. Our theory captures how the convergence of distributed algorithms behaves as the number of machines and the size of local data vary. Furthermore, we show that when the data are less balanced, regularization can be used to ensure convergence at a slower rate. We also demonstrate that our analysis can be further extended to handle nonconvex loss functions.

Index Terms—Distributed optimization, stochastic optimization, master/slave model, variance reduction.

I. INTRODUCTION

Empirical risk minimization arises frequently in machine learning and signal processing, where the objective function is the average of losses computed at different data points. Due to the increasing size of data, distributed computing architectures, which assign the learning task over multiple computing nodes, are in great need to meet the scalability requirement in terms of both computation power and storage space.

In addition, distributed frameworks are suitable for problems where there are privacy concerns to transmit and store all the data in a central location, a scenario related to the nascent field of federated learning [1]. It is, therefore, necessary to develop distributed optimization frameworks that are tailored to solving large-scale empirical risk minimization problems with desirable communication-computation trade-offs, where the data are stored disjointly over different machines.

One prevalent model of distributed systems is the so-called master/slave model, where there is a central parameter server to coordinate the computation and information exchange across different worker machines. Due to the low per-iteration cost, a popular solution is distributed stochastic gradient descent (SGD) [2], where the parameter server aggregates gradients from each worker and does mini-batch gradient updates. However, distributed SGD is not communication-efficient and requires lots of communication rounds to converge, which partially diminishes the benefit of distribution. On the other hand, recent breakthroughs in developing stochastic variance reduced methods have made it possible to achieve fast convergence and small per-iteration cost at the same time, such as the notable SVRG [3] algorithm. Yet, distributed schemes of such variance reduced methods that are both practical and theoretically sound are much less developed.

This paper focuses on a general framework of distributed stochastic variance reduced methods, to be presented in Algorithm 1, which is natural and friendly to implement in a master/slave model. On a high level, SVRG-type algorithms [3] contain inner loops for parameter updates via variance-reduced SGD, and outer loops for global gradient and parameter updates. Our general framework assigns outer loops to the parameter server, and inner loops to worker machines. The parameter server collects gradients from worker machines and then distributes the global gradient to each machine. Each worker machine then runs the inner loop independently in parallel using variance reduction techniques, which might be different when distributing different algorithms, and returns the updates to the parameter server at the end. Per iteration, two communication rounds are required: one communication round is used to average the parameter estimates, and the other is used to average the gradients, which is the same as distributed synchronous SGD. However, the premise is that by performing more efficient local computation using stochastic variance reduced methods, the algorithm converges in fewer iterations and is therefore more communication-efficient.

Due to the simplicity of this framework, similar methods have been implemented in several works [1], [4], [5], and have
achieved great empirical success. Surprisingly, a complete theoretical understanding of its convergence behavior is still missing at large. Moreover, distributed variants using accelerated variance reduction methods are not developed. The main analysis difficulty is that the variance-reduced gradient of each worker is no longer an unbiased gradient estimator when sampling from re-used local data.

To ease this difficulty, several variants of distributed SVRG, e.g. [6]–[8] have been proposed with performance guarantees, which try to bypass the biased gradient estimation issue by simulating the process of i.i.d. sampling from the global data using some complicated random data re-allocation protocol, which requires sampling extra data with or without replacement. These procedures lead to unnecessary data waste and potential privacy leakage, and can be cumbersome and difficult to implement in practice.

Consequently, a natural question arises: can we provide a mathematical analysis to the convergence of the natural framework of distributed stochastic variance reduced methods, under some simple and intuitive metric?

A. Contributions of This Paper

This paper provides a convergence analysis of a family of naturally distributed stochastic variance reduced methods under the framework described in Algorithm 1, for both convex and nonconvex loss functions. By using different variance reduction schemes at the worker machines, we study distributed variants of three representative algorithms in this paper: SVRG [3], SARAH employing recursive gradient updates [9], [10], and MiG employing accelerated gradient updates [11]. Our methodology can be extended to study other variants in a similar fashion. The contributions of this paper are summarized below.

- We suggest a simple and intuitive metric called distributed smoothness to gauge data balancedness among workers, defined as the smoothness of the difference \( f_k - f \) between the local loss function \( f_k \) and the global loss function \( f \), which is the average of the local loss functions. The metric is deterministic, easy-to-compute, applies for arbitrary dataset splitting, and is shown to play a critical role in the convergence analysis.

- We establish the linear convergence of distributed D-SVRG, D-SARAH, and D-MiG under strongly convex losses, as long as the distributed smoothness parameter is smaller than a constant fraction of the strong convexity parameter \( \sigma \), e.g. \( \sigma / 4 \), where the fraction might change for different algorithms. Our bounds capture the phenomenon that the convergence rate improves as the local loss functions become more similar to the global loss function, by reducing the distributed smoothness parameter. Furthermore, the run time complexity exhibits the so-called “linear speed-up” property in distributed computing, where the complexity depends on the local data size, instead of the global data size, which typically implies an improvement by a factor of \( n \), where \( n \) is the number of machines.

- When the local data are highly unbalanced, the distributed smoothness parameter becomes large, which implies that the algorithm might diverge. We suggest regularization as an effective way to handle this situation, and show that by adding larger regularization to machines that are less distributed smooth, one can still ensure linear convergence in a regularized version of D-SVRG, called D-RSVRG, though at a slower rate of convergence.

- More generally, the notion of distributed smoothness can also be used to establish the convergence under nonconvex losses. We demonstrate this through the convergence analysis of D-SARAH in the nonconvex setting.

B. Related Work

Distributed optimization is a classic topic [14], [15], yet recent trends in data-intensive applications are calling for new developments with a focus on communication and computation efficiency. Examples of deterministic optimization methods include DANE [16], AIDE [5], DiSCo [17], GIANT [18], CoCoA [19], CEASE [20], one-shot averaging [21], [22], etc.

Many stochastic variance reduced methods have been proposed recently, for example, SAG [23], SAGA [24], SVRG [3], SDCA [25], MiG [11], Katyusha [26], Catalyst [27], SCOPE [12], [13], SARAH [9], SPIDER [28], SpiderBoost [29], to name a few. Several previous works have studied distributed variants of SVRG. For example, the D-SVRG algorithm has been empirically studied before in [1], [5] without a theoretical convergence analysis. The pSCOPE algorithm [13] is also a variant of distributed SVRG, and its convergence is studied under an assumption called good data partition in [13], which is hard to interpret and verify in practice. The SCOPE algorithm [12] is similar to the regularized variant D-RSVRG of D-SVRG under large regularization, however our analysis is much more refined by allowing different regularizations to different local workers with respect to the distributed smoothness of local data, and gracefully degenerates to the unregularized case when the distributed smoothness is benign. The general framework of distributed variance-reduced methods covering SARAH and MiG and various loss settings in this paper has not been studied before. For conciseness, Table I summarizes the communication and runtime complexities of the most relevant algorithms to the current paper.1

There are also a lot of recent efforts on reducing the communication cost of distributed GD/SGD by gradient quantization [30]–[33], gradient compression and sparsification [34]–[38]. In comparison, we communicate the exact gradient, and it is an interesting future direction to combine gradient compression schemes in distributed variance reduced stochastic gradient methods. Another line of works [39]–[41] seek to adopt variance reduction techniques in the decentralized setting, where each worker is only able to communicate with its neighbors over a network. In contrast, our work focuses on the master/slave

1Since in a master/slave model, all worker machines perform computations in parallel, the runtime is dictated by the worst-case computation complexity per worker.
model where each worker communicates with a parameter server.

C. Paper Organization

The rest of this paper is organized as follows. Section II presents the problem setup and a general framework of distributed stochastic optimization with variance-reduced local updates. Section III presents the convergence guarantees of D-SVRG, D-SARAH and D-MiG under appropriate distributed smoothness assumptions. Section IV introduces regularization to D-SVRG to handle unbalanced data when distributed smoothness does not hold. Section V presents extensions to nonconvex losses for D-SARAH. Section VI provides an outline to the analysis and the rest of the proofs are deferred to the appendix. Section VII presents numerical experiments to corroborate the theoretical findings. Finally, we conclude in Section VIII.

II. PROBLEM SETUP

Suppose we have a data set \( \mathcal{M} = \{z_1, \ldots, z_N\} \), where \( z_j \in \mathbb{R}^d \) is the \( j \)th data point for \( j = 1, \ldots, N \), and \( N \) is the total number of data points. In particular, we do not make any assumptions on their statistical distribution. Consider the following empirical risk minimization problem

\[
\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{N} \sum_{z \in \mathcal{M}} \ell(x; z),
\]

where \( x \in \mathbb{R}^d \) is the parameter to be optimized and \( \ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \) is the sample loss function. For brevity, we use \( \ell_z(x) \) to denote \( \ell(x; z) \) throughout the paper.

In a distributed setting, where the data are distributed to \( n \) machines or workers, we define a partition of the data set \( \mathcal{M} \) as \( \mathcal{M} = \bigcup_{k=1}^n \mathcal{M}_k \), where \( \mathcal{M}_i \cap \mathcal{M}_k = \emptyset, \forall i \neq k \). The \( k \)th worker, correspondingly, is in possession of the data subset \( \mathcal{M}_k \), \( 1 \leq k \leq n \). We assume there is a parameter server (PS) that coordinates the parameter sharing among the workers. The sizes of data held by each worker machine is \( N_k = |\mathcal{M}_k| \). When the data is split equally, we have \( N_k = N/n \). The original problem (1) can be rewritten as minimizing the following objective function:

\[
f(x) := \frac{1}{n} \sum_{k=1}^n f_k(x),
\]

where \( f_k(x) = \frac{1}{N/n} \sum_{z \in \mathcal{M}_k} \ell_z(x) \) is the local loss function at the \( k \)th worker machine.

Algorithm 1: A General Distributed Framework for Empirical Risk Minimization.

1: Input: initial point \( \tilde{x}^0 \).
2: Initialization: Compute \( \nabla f(\tilde{x}^0) \) and distribute it to all machines.
3: for \( t = 0, 1, 2, \cdots \) do
4:     for workers \( 1 \leq k \leq n \) in parallel do
5:         Compute \( y_k^{t+} = \text{LocalUpdate}(f_k, \tilde{x}^t, \nabla f(\tilde{x}^t)) \);
6:         Send \( y_k^{t+} \) to PS;
7:     end for
8:     PS: randomly select \( \tilde{x}^{t+1} \) from all \( y_k^{t+} \) and push \( \tilde{x}^{t+1} \) to all workers;
9:     for workers \( 1 \leq k \leq n \) in parallel do
10:        Compute \( \nabla f_k(\tilde{x}^{t+1}) \) and send it to PS;
11:    end for
12: PS: average \( \nabla f(\tilde{x}^{t+1}) = \frac{1}{n} \sum_{k=1}^n \nabla f_k(\tilde{x}^{t+1}) \) and push \( \nabla f(\tilde{x}^{t+1}) \) to all workers.
13: end for
14: return \( \tilde{x}^{t+1} \)

where \( f_k(x) = \frac{1}{N/n} \sum_{z \in \mathcal{M}_k} \ell_z(x) \) is the local loss function at the \( k \)th worker machine.

A. A General Distributed Framework

Algorithm 1 presents a general framework for distributed stochastic variance reduced methods, which assigns the outer loops of an SVRG-type algorithm [3], [9], [11] to PS and the inner loops to local workers. By using different variance reduction schemes at the worker machines (i.e. \texttt{LocalUpdate}), we obtain distributed variants of different algorithms. On a high level, the framework alternates between local computation by individual workers in parallel (i.e. Line 5), and global information sharing coordinated by the PS (i.e. Lines 8-12).

- The local worker conducts local computation \texttt{LocalUpdate} based on the current estimate \( \tilde{x}^t \), the global gradient \( \nabla f(\tilde{x}^t) \), and its local data \( f_k(\cdot) \); in this paper, we are primarily interested in local updates using proper rescaling.

TABLE I

| Algorithm          | Communication Rounds | Runtime       | Assumptions         |
|--------------------|----------------------|---------------|---------------------|
| D-PSVRG [6]        | \((1 + \kappa/(N/n)) \log(1/\epsilon)\) | \((N/n + \kappa) \log(1/\epsilon)\) | extra data          |
| DASVRG [6]         | \((1 + \sqrt{\kappa}/(N/n)) \log(1/\epsilon)\) | \((N/n + \sqrt{\kappa}/(N/n)) \log(1/\epsilon)\) | extra data          |
| Dist. AGD          | \sqrt{\kappa} \log(1/\epsilon) | \((N/n) \sqrt{\kappa} \log(1/\epsilon)\) | non-distributed     |
| ADMM               | \kappa \log(1/\epsilon) | \((N/n) \kappa \log(1/\epsilon)\) | uniform regulariza-
| SCOPE [12]         | \kappa \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | distributed smooth-
| pSCOPE [13]        | \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | good partition      |
| D-SVRG*            | \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | distributed smooth-
| D-SARAH*           | \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | distributed smooth-
| D-MiG*             | \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | large regulariza-
| D-RSVRG*           | \log(1/\epsilon) | \((N/n + \kappa) \log(1/\epsilon)\) | small regulariza-

2It is straightforward to state our results under unequal data splitting with proper rescaling.
stochastic variance-reduced gradients. A little additional information about the previous update is needed when employing acceleration, which will be specified in Algorithm 3.

- The PS randomly selects from the local estimates \( y_{k}^{t+c} \) from all workers in each round, which is set as the global estimate \( \tilde{x}^{t+1} \), then computes the global gradient \( \nabla f(\tilde{x}^{t+1}) \) by pulling the local gradient \( \nabla f_k(\tilde{x}^{t+1}) \). In total, each iteration requires two rounds of communications.

Remark 1: A careful reader may suggest that there are better ways to set the global estimate \( \tilde{x}^{t+1} \). One way is to let the PS randomly select a worker to compute a local estimate \( y_{k}^{t+c} \) and set other workers in an idle mode, which saves local computation. Another way is to set \( \tilde{x}^{t+1} = \frac{1}{n} \sum_{k=1}^{n} y_{k}^{t+c} \) as the average of all local updates, which may potentially improve the performance. The specific choice in Algorithm 1 allows us to provide theoretical analysis for all the LocalUpdate rules considered in this paper.

B. Assumptions

Throughout, we invoke one or several of the following standard assumptions of the loss function in the convergence analysis.

Assumption 1 (Smoothness): The sample loss \( \ell_z(\cdot) \) is \( L \)-smooth, i.e., the gradient of \( \nabla \ell_z(\cdot) \) is \( L \)-Lipschitz for all \( z \in M \).

Assumption 2 (Convexity): The sample loss \( \ell_z(\cdot) \) is convex for all \( z \in M \).

Assumption 3 (Strong Convexity): The empirical risk \( f(\cdot) \) is \( \sigma \)-strongly convex.

When \( f \) is strongly convex, the condition number of \( f \) is defined as \( \kappa := L/\sigma \). Denote the unique minimizer and the optimal value of \( f(x) \) as

\[
x^* := \arg\min_{x \in \mathbb{R}^d} f(x), \quad f^* := f(x^*).
\]

As it turns out, the smoothness of the deviation \( f_k - f \) between the local loss function \( f_k \) and the global loss function \( f \) plays a key role in the convergence analysis, as it measures the balancedness between local data in a simple and intuitive manner. We refer to this as the “distributed smoothness”. In some cases, a weaker notion called restricted smoothness is sufficient, which is defined below.

Definition 1 (Restricted Smoothness): A differentiable function \( f : \mathbb{R}^d \to \mathbb{R} \) is called \( c \)-restricted smooth with regard to \( x^* \) if

\[
\| \nabla f(x^*) - \nabla f(y) \| \leq c \| x^* - y \|, \quad \text{for all } y \in \mathbb{R}^d.
\]

The restricted smoothness, compared to standard smoothness, fixes one of the arguments to \( x^* \), and is therefore a much weaker requirement. The following assumption quantifies the distributed smoothness using either restricted smoothness or standard smoothness.

Assumption 4a (Distributed Restricted Smoothness): The deviation \( f - f_k \) is \( c_k \)-restricted smooth with regard to \( x^* \) for all \( 1 \leq k \leq n \).

Assumption 4b (Distributed Smoothness): The deviation \( f - f_k \) is \( c_k \)-smooth for all \( 1 \leq k \leq n \).

It is straightforward to check that \( c_k \leq L \) for all \( 1 \leq k \leq n \). If all the data samples are generated following certain statistical distribution in an i.i.d. fashion, one can further link the distributed smoothness to the local sample size \( N/n \), where \( c_k \) decreases with the increase of \( N/n \), see e.g. [16], [20] for further discussion.

Remark 2: We provide a toy example to illustrate the difference between Assumptions 4a and 4b. Let \( x \in \mathbb{R} \) and \( \ell_z(x) = L\delta(\sigma)(x) + \frac{L}{2} x^2 \), where \( L\delta(\sigma)(\cdot) \) denotes Huber loss with \( \delta(z) > 0 \). We set \( \delta(z) = 0.99 \) for half of \( z \in M \), and 1 for the rest. Therefore, \( f(x) = \frac{1}{2}(L_{0.99}(x) + L_1(x)) + \frac{L}{2} x^2 \) with \( x^* \neq 0 \). Now, consider some local loss function \( f_k(x) = (\frac{1}{2} + c_k) L_{0.99}(x) + (\frac{1}{2} - c_k) L_1(x) + \frac{L}{2} x^2 \) for some \( c_k \). We can verify that Assumption 4a holds with \( c_k = |c_k|/100 \) while Assumption 4b holds with \( c_k = |c_k| \). Therefore, the distributed smoothness parameter can be much smaller when invoking the weaker Assumption 4a.

III. CONVERGENCE IN THE STRONGLY CONVEX CASE

In this section, we describe three variance-reduced routines for LocalUpdate used in Algorithm 1, namely SVRG [3], SARAH [9, 10], and MIG [11], and analyze their convergence when \( f(\cdot) \) is strongly convex, respectively.

A. Distributed SVRG (D-SVRG)

The LocalUpdate routine of D-SVRG is described in Algorithm 2. Theorem 1 provides the convergence guarantee of D-SVRG as long as the distributed restricted smoothness parameter is small enough.

Theorem 1 (D-SVRG): Suppose that Assumptions 1, 2 and 3 hold, and Assumption 4a holds with \( c_k \leq c < \sigma/4 \). With \( m = O(\kappa(1 - 4c/\sigma)^{-2}) \) and proper step size \( \eta \), the iterates of D-SVRG satisfy

\[
\mathbb{E} \left[ f(\tilde{x}^{t+1}) - f^* \right] < \frac{\sigma - 2c}{2(\sigma - 3c)} \mathbb{E} \left[ f(\tilde{x}^t) - f^* \right].
\]

The communication and runtime complexities of finding an \( \epsilon \)-optimal solution (in terms of function value) are

\[
O(\zeta^{-1} \log(1/\epsilon)) \quad \text{and} \quad O((N/n + c^{-2}\kappa)^{-1} \log(1/\epsilon))
\]

respectively, where \( \zeta = 1 - 4c/\sigma \).

Theorem 1 establishes the linear convergence of function values in expectation for D-SVRG, as long as the parameter \( c \) is sufficiently small, e.g. \( c < \sigma/4 \). From the expressions of communication and runtime complexities, it can be seen that the smaller \( c \), the faster D-SVRG converges—suggesting that the homogeneity of distributed data plays an important role in the efficiency of distributed optimization. When \( c \) is set such that \( c/\sigma \) is bounded above by a constant smaller than 1/4, i.e. \( \zeta = O(1) \), the runtime complexity becomes \( O((N/n + \kappa)^{-1} \log(1/\epsilon)) \), which improves the counterpart of SVRG \( O((N + \kappa)^{-1} \log(1/\epsilon)) \) in the centralized setting.

Remark 3: The above LocalUpdate routine corresponds to the so-called Option II (w.r.t. setting \( y_{k}^{t+c} \) as uniformly at random selected from previous updates) specified in [3]. Under similar assumptions, we also establish the convergence of D-SVRG using Option I, where the output \( y_{k}^{t,m} \) is set as \( y_{k}^{t,m} \). In addition, D-SVRG still converges linearly in the absence of Assumption 2. We leave these extensions in the supplementary materials.
Algorithm 2: LocalUpdate via SVRG/SARAH.
1: **Input:** local data $\mathcal{M}_k$, $\tilde{x}^t$, $\nabla f(\tilde{x}^t)$;
2: **Parameters:** step size $\eta$, number of iterations $m$;
3: Set $y^t_0 = \tilde{x}^t$, $v^t_0 = \nabla f(\tilde{x}^t)$;
4: for $s = 0, \ldots, m - 1$ do
5: Sample $z$ from $\mathcal{M}_k$ uniformly at random;
6: Compute
$$v^t_{s+1} = \begin{cases} \nabla \ell_z(y^t_{s+1}) - \nabla \ell_z(\tilde{x}^t) + \nabla f(\tilde{x}^t) & \text{SVRG} \\ \nabla \ell_z(y^t_{s+1}) - \nabla \ell_z(y^t_s) + v^t_s & \text{SARAH} \end{cases}$$
7: $y^t_{s+1} = y^t_s - \eta v^t_s$;
8: end for
9: Set $y^{t,1}_k$, $\ldots$, $y^{t,m}_k$ uniformly at random from $y^{t,1}_k$.

B. Distributed SARAH (D-SARAH)

The LocalUpdate of D-SARAH is also described in Algorithm 2, which is different from SVRG in the update of stochastic gradient $v^t_{s,a}$, by using a recursive formula proposed in [9]. Theorem 2 provides the convergence guarantee of D-SARAH as long as the distributed restricted smoothness parameter is small enough.

**Theorem 2 (D-SARAH):** Suppose that Assumptions 1, 2 and 3 hold, and Assumption 4a holds with $c_k \leq c < \sqrt{2c}/\sigma$. With $m = O(\kappa(1 - 2\sqrt{2c}/c))$ and proper step size $\eta$, the iterates of D-SARAH satisfy
$$E \left[ \| \nabla f(\tilde{x}^{t+1}) \|^2 \right] < \frac{1}{2 - 8c^2/\sigma^2} E \left[ \| \nabla f(\tilde{x}^t) \|^2 \right] .$$

The communication and runtime complexities of finding an $\epsilon$-optimal solution (in terms of gradient norm) are $O(\zeta^{-1} \log(1/\epsilon))$ and $O((N/n + \zeta^{-2}) \zeta^{-1} \log(1/\epsilon))$ respectively, where $\zeta = 1 - 2\sqrt{2c}/\sigma$.

Theorem 2 establishes the linear convergence of the gradient norm in expectation for D-SARAH, as long as the parameter $c$ is small enough. Similar to D-SVRG, a smaller $c$ leads to faster convergence of D-SARAH. When $c$ is set such that $c/\sigma$ is bounded by a constant smaller than $\sqrt{2}/4$, the runtime complexity becomes $O((N/n + \kappa) \log(1/\epsilon))$, which improves the counterpart of SARAH $O((N + \kappa) \log(1/\epsilon))$ in the centralized setting. In particular, Theorem 2 suggests that D-SARAH may allow a larger $c$, compared with D-SVRG, to guarantee convergence.

C. Distributed MiG (D-MiG)

The LocalUpdate of D-MiG is described in Algorithm 3, which is inspired by the inner loop of the MiG algorithm [11], a recently proposed accelerated variance-reduced algorithm. Compared with D-SVRG and D-SARAH, D-MiG uses additional information from the previous updates, according to Line 3-6 in Algorithm 3. Theorem 3 provides the convergence guarantee of D-MiG, as long as the distributed smoothness parameter is small enough.

**Theorem 3 (D-MiG):** Suppose that Assumptions 1, 2 and 3 hold, and Assumption 4b holds with $c_k \leq c < \sigma/8$. Let $w = (1 + \eta \sigma)/(1 + 3\eta c)$. With $m = O((N/n) + \kappa) \log(1/\epsilon))$, the iterates of D-MiG achieve an $\epsilon$-optimal solution within a communication complexity of $O((1 + \sqrt{2c}/\sigma) \log(1/\epsilon))$ and runtime complexity of $O((N/n + \sqrt{2c}/\sigma) \log(1/\epsilon))$.

IV. REGULARIZATION HELPS UNBALANCED DATA

So far, we have established the convergence when the distributed smoothness is not too large. While it may be reasonable in certain settings, e.g. in a data center where one has control over how to distribute data, it is increasingly harder to satisfy when the data are generated locally and heterogeneous across workers. However, when such conditions are violated, the algorithms might diverge. In this situation, adding a regularization term might ensure the convergence, at the cost of possibly slowing down the convergence rate. We consider regularizing the local gradient update of D-SVRG in Algorithm 2 as
$$v^{t,s+1}_k = \nabla \ell_z(y^{t,s+1}_k) - \nabla \ell_z(\tilde{x}^t) + \nabla f(\tilde{x}^t) + \mu_k (y^{t,s+1}_k - \tilde{x}^t),$$
where the last regularization term penalizes the proximity between the current iterates $y^{t,s+1}_k$ and the reference point $\tilde{x}^t$, where $\mu_k > 0$ is the regularization parameter employed at the $k$th worker. We have the following theorem.

**Theorem 4 (Distributed Regularized SVRG (D-RSVRG)):** Suppose that Assumptions 1, 2 and 3 hold, and Assumption 4a
holds with \( ck < (\sigma + \mu_k)/4 \). Let \( \mu = \min_{1<k<n} \mu_k \). With proper \( m \) and step size \( \eta \), there exists some constant \( 0 \leq \eta < 1 \) such that the iterates of D-\text{RSVRG} satisfy
\[
\mathbb{E} \left[ f(\hat{x}^{t+1}) - f^* \right] \leq \left( 1 - (1 - \nu) \frac{\sigma}{L + \mu} \right) \mathbb{E} \left[ f(\hat{x}^t) - f^* \right],
\]
and the runtime complexity of finding an \( \epsilon \)-optimal solution is bounded by
\[
\mathcal{O} \left( \left( N/n + \zeta^{-2} \rho \right) \zeta^{-1} \min \left\{ \kappa + \frac{\mu}{\sigma}, \frac{1}{\max\{1 - \mu/\sigma, 0\}} \right\} \log(1/\epsilon) \right),
\]
where \( \zeta = 1 - 4c/(\sigma + \mu) \) and \( \rho = (L + \mu)/(\sigma + \mu) \).

Compared with Theorem 1, Theorem 4 relaxes the assumption \( ck < \sigma/4 \) to \( ck < (\sigma + \mu_k)/4 \), which means that by inserting a larger regularization \( \mu_k \) to local workers that are not distributed smooth, i.e., those with large \( ck \), one can still guarantee the convergence of D-\text{RSVRG}. However, increasing \( \mu \) leads to a slower convergence rate: a large \( \mu = 8L \) leads to an iteration complexity \( \mathcal{O}(\kappa \log(1/\epsilon)) \), similar to gradient descent. Compared with SCOPE [12] which requires a uniform regularization \( \mu > L - \sigma \), our analysis applies tailored regularization to local workers, and potentially allows much smaller regularization to guarantee the convergence, since \( ck \)'s can be much smaller than the smoothness parameter \( L \).

V. CONVERGENCE IN THE NONCONVEX CASE

In this section, we extend the convergence analysis of D-SARAH to handle nonconvex loss functions, since SARAH-type algorithms are recently shown to achieve near-optimal performances for nonconvex problems [10], [28], [29]. As a modification that eases the analysis, we make every worker return \( y_k^+ = y_k^{t,m} \) in line 9 of Algorithm 2. Our result is summarized in the theorem below.

Theorem 5 (D-SARAH for non-convex losses): Suppose that Assumption 1 and Assumption 4b hold with \( ck \leq c \). With the step size \( \eta = \frac{2}{(L + \sqrt{1 + 6m + 4m(m-1)^2})^2} \), D-SARAH satisfies
\[
\frac{1}{Tm} \sum_{t=0}^{T-1} \sum_{s=0}^{m-1} \mathbb{E} \left[ \left\| \nabla f(\hat{x}^{t,s}_{k(t)}) \right\|^2 \right] \leq \frac{2}{\eta Tm} (f(\hat{x}^{0}) - f^*),
\]
where \( k(t) \) is the agent index selected in the \( t \)-th round for parameter update, i.e., \( \hat{x}^{t+1} = y_k^{t+1} \) (c.f. line 8 of Algorithm 1). To find an \( \epsilon \)-optimal solution, the communication complexity is \( \mathcal{O}(1 + (\sqrt{N/n} + c/L) L/\epsilon) \), and the runtime complexity is \( \mathcal{O}(N/n + \sqrt{(N/n)N/n} + N/n \cdot c/L) L/\epsilon \) by setting \( m = \mathcal{O}(N/n) \).

Theorem 5 suggests that D-SARAH converges as long as the step size is small enough. Furthermore, a smaller \( c \) allows a larger step size \( \eta \), and hence faster convergence. To gain further insights, assuming i.i.d. data at each worker, by concentration inequalities it is known that \( c/L = \mathcal{O}(\sqrt{\log(N/n)/N/n}) \) under mild conditions [42], and consequently, the runtime complexity of finding an \( \epsilon \)-accurate solution using D-SARAH is \( \mathcal{O}(N/n + L \sqrt{\log(N/n)/N/n}) \). This is comparable to the best known result \( \mathcal{O}(N + L \sqrt{N/n}) \) for the centralized SARAH-type algorithms in the nonconvex setting [10], [28], [29] up to logarithmic factors, where the data size is replaced from \( N \) to the size of local data \( N/n \) – demonstrating again the benefit of data distribution.

VI. PROOFS OF MAIN THEOREMS

In this section, we outline the convergence proofs of D-SVRG (Theorem 1), D-\text{RSVRG} (Theorem 4), D-SARAH in the strongly convex (Theorem 2) and nonconvex (Theorem 5) settings, while leaving the details to Appendix D. The convergence proof of D-MiG (Theorem 3) is delegated to the supplemental materials due to space limits. Throughout this section, we simplify the notations \( y_k^s \) and \( V_k^s \) by dropping the \( t \) superscript and the \( k \) subscript, whenever the meaning is clear, since it is often sufficient to analyze the convergence of a specific worker \( k \) during a single round.

A. D-SVRG (Theorem 1)

We generalize the analysis of SVRG using the dissipativity theory in [43] to the analysis of D-SVRG, which might be of independent interest. Setting \( \xi^t = y^t - x^t \), we can write the local update of D-SVRG via the following linear time-invariant system [43]:
\[
\xi^{t+1} = \xi^t - \eta \left[ \nabla \ell_z(y^t) - \nabla \ell_z(y^0) + \nabla f(y^0) \right],
\]
where \( z \) is selected uniformly at random from local data points \( \mathcal{M}_k \), or equivalently,
\[
\xi^{t+1} = A \xi^t + B w^s,
\]
with \( A = I_d, B = [-\eta I_d, -\eta I_d] \), and
\[
w^s = \left[ \nabla \ell_z(y^t) - \nabla \ell_z(x^t) \nabla \ell_z(x^s) - \nabla \ell_z(y^0) + \nabla f(y^0) \right].
\]

Here, \( I_d \) is the identity matrix of dimension \( d \).

Dissipativity theory characterizes how “the inputs” \( w^s \), \( s = 0, 1, 2, \ldots \) drive the internal energy stored in the “states” \( \xi^t \), \( s = 0, 1, 2, \ldots \) via an energy function \( V : \mathbb{R}^d \rightarrow \mathbb{R}_+ \) and a supply rate \( S : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}^d \). For our purpose, it is sufficient to choose the energy function as \( V(\xi) = \|\xi\|^2/2 \). By setting \( S(\xi, w) = \sum_{j=1}^J \lambda_j S_j(\xi, w) \) as the supply rate, where
\[
S_j(\xi, w) = [\xi^+, w^+] J \xi_j [\xi^+, w^+]^T,
\]
we have
\[
V(\xi^{t+1}) \leq \rho^2 V(\xi^k) + \sum_{j=1}^J \lambda_j S_j(\xi^t, w^s)
\]
for some \( \rho \in (0, 1] \) as long as there exist non-negative scalars \( \lambda_j \) such that
\[
[\lambda J A \rho^2 I_d \lambda J B \rho^2 B^T] - J \lambda_j X_j \preceq 0.
\]
In fact, by left multiplying \( [\xi^+, w^+] \) and right multiplying \( [\xi^+_k, w^+_k]^T \) to (8), we recover (7).
To invoke the dissipativity theory for convergence analysis, we take two steps. First, we capture properties of the objective function such as strong convexity and co-coercivity using the supply rate functions by selecting proper matrices $X_j$ (c.f. Lemma 1). Next, we invoke (8) to determine a valid combination of supply rate functions to guarantee the dissipativity inequality (c.f. Lemma 2), which entails the rate of convergence.

To proceed, we start by defining the supply rate functions in (6). Since $\xi^* \in \mathbb{R}^d$ and $w^* \in \mathbb{R}^{2d}$, we will write $X_j$ as $X_j = \tilde{X}_j \otimes I_k$, where $\tilde{X}_j \in \mathbb{R}^{3 \times 3}$. Following [43], we consider the supply rates characterized by the following matrices:

$$
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}, \quad
\begin{bmatrix}
0 & -1 & -1 \\
-1 & 0 & 0 \\
-1 & 0 & 0
\end{bmatrix},
$$

which correspond to the supply rates:

$$
\begin{align*}
E[S_1] &= E \left[ \| \nabla \ell_z(y^s) - \nabla \ell_z(x^*) \|^2 \right], \\
E[S_2] &= E \left[ \| \nabla \ell_z(x^s) - \nabla \ell_z(y^0) - \nabla f(y^0) \|^2 \right], \\
E[S_3] &= -2E \left[ \langle y^s - x^s, \nabla \ell_z(y^s) - \nabla \ell_z(y^0) + \nabla f(y^0) \rangle \right].
\end{align*}
$$

(9)

The following lemma, proved in Appendix A, bounds the supply rates when $z$ is drawn from the local data $\mathcal{M}_k$.

**Lemma 1:** Suppose that Assumption 1, 2, 3 and 4a hold with $c_k \leq c$. For the supply rates defined in (9), we have

$$
\begin{align*}
E[S_1] &\leq 2L\left( \| y^s - x^s \|^2 + 2cL \| y^s - x^s \|^2 \right), \\
E[S_2] &\leq 4L\left( \| y^s - x^s \|^2 + c(4L + 2c) \| y^s - x^s \|^2 \right), \\
E[S_3] &\leq -2E \left[ \langle y^s - x^s, \nabla \ell_z(y^s) - \nabla \ell_z(y^0) + \nabla f(y^0) \rangle \right] + cE \left[ \| y^s - x^s \|^2 \right].
\end{align*}
$$

The next lemma, proved in Appendix B, details the dissipativity inequality when (8) holds when setting $\rho = 1$.

**Lemma 2:** Suppose that Assumption 1, 2, 3 and 4a hold with $c_k \leq c$. If there exist non-negative scalars $\lambda_j$, $j = 1, 2, 3$, such that

$$
\lambda_3 - L\lambda_1 - (2L\lambda_1 + 3\lambda_3)c/\sigma > 0
$$

(10)

and

$$
\begin{bmatrix}
0 & \lambda_3 - \eta & 0 \\
\lambda_3 - \eta & -\lambda_1 & \eta^2 \\
\lambda_1 & \eta^2 & -\lambda_2
\end{bmatrix} \preceq 0
$$

(11)

hold. Then D-SVRG satisfies

$$
E[f(y^{s+1}) - f^*] \leq \frac{1}{\sigma m} \left[ \frac{1}{\sigma m} + \frac{c}{\sigma} \left( (4L + 2c)\lambda_2 + \lambda_3 \right) + 2L\lambda_2 \right] E\left[ f(y^0) - f^* \right],
$$

(12)

where the final output $y^+$ is selected from $y^1, \ldots, y^m$ uniformly at random.

We are now ready to prove Theorem 1. Set $\lambda_1 = \lambda_2 = 2\eta^2$, and $\lambda_3 = \eta$. We have (10) holds with the step size $\eta < \sigma - 3c/L(2\sigma + 4c)$, and (11) holds since

$$
\begin{bmatrix}
0 & 0 & 0 \\
0 & -\eta^2 & \eta^2 \\
\eta^2 & -\eta^2
\end{bmatrix} \preceq 0.
$$

Applying Lemma 2 with the above choice of parameters, (12) can be written as

$$
E[f(y^{s+1}) - f^*] \leq \frac{1}{\sigma m} \left[ \frac{1}{\sigma m} + c((8L + 4c)\eta + 1) + 4L\eta \right] \frac{1}{\sigma(1 - 2L\eta) - c(4L\eta + 3)} E[f(y^0) - f^*].
$$

When $c < \sigma/4$, by choosing $\eta = (1 - 4c/\sigma)(40L)^{-1}$, $m = 160\kappa(1 - 4c/\sigma)^{-2}$, a convergence rate no more than $\nu := 1 - \frac{1}{2} \cdot \frac{\sigma - 4c}{\sigma + 4c}$ is obtained. Therefore, after line 8 of Algorithm 1, we have for D-SVRG,

$$
E[f(\tilde{x}^{t+1}) - f^*] \leq \frac{1}{n} \sum_{k=1}^{n} E[f(y_k^{t+1}) - f^*] \\
\leq \left( 1 - 2 \cdot \frac{\sigma - 4c}{\sigma + 4c} \right) E[f(\tilde{x}^t) - f^*].
$$

B. D-RSVRG (Theorem 4)

Consider an auxiliary sample function at the $t^{th}$ round and the $k^{th}$ worker as

$$
\ell_{\mu_k}(x; z) = \ell(x; z) + \frac{\mu_k}{2} \| x - \tilde{x}^t \|^2.
$$

This leads to the auxiliary local and global loss functions, respectively,

$$
f_{\ell}^t(x) = \frac{1}{|M_i|} \sum_{z \in M_i} \ell_{\mu_k}(x; z) = f_i(x) + \frac{\mu_k}{2} \| x - \tilde{x}^t \|^2,
$$

for $1 \leq i \leq n$, and

$$
f_{\ell}^t(x) = \frac{1}{n} \sum_{i=1}^{n} f_{\ell}^t(x) = f(x) + \frac{\mu_k}{2} \| x - \tilde{x}^t \|^2.
$$

(13)

Moreover, we have

$$
\nabla \ell_{\mu_k}(y_k^{t+1}; z) - \nabla \ell_{\mu_k}(\tilde{x}^t; z) + \nabla f(\tilde{x}^t) = \nabla \ell(y_k^{t+1}; z) - \nabla \ell(\tilde{x}^t; z) + \nabla f(\tilde{x}^t) + \mu_k(y_k^{t+1} - \tilde{x}^t),
$$

which means that D-RSVRG performs in exactly the same way as the unregularized D-SVRG with the auxiliary loss functions $\ell_{\mu_k}$ in the $t^{th}$ round. Note that $\ell_{\mu_k}$ is $(\mu_k + L)$-smooth and that $f^t$ is $(\mu_k + \sigma)$-strongly-convex, while the restricted smoothness of
the $k$th worker, $f^t - f^t_k = f - f_k$ remains unchanged. Applying Theorem 1, we have
\[\mathbb{E} [f^t(\tilde{x}^{t+1}) - f^*] \leq \frac{2}{\eta} \mathbb{E} [f(y^0) - f(y^m)] + \frac{1}{\eta} \sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s) - v^s\| \right] - (1 - L\eta) \sum_{s=0}^{m-1} \mathbb{E} \left[ \|v^s\|^2 \right].\]

C. D-SARAH in the Strongly Convex Case (Theorem 2)

We motivate the convergence analysis by citing the following lemma from [9], which bounds the sum of gradient norm of iterations on a specific worker during $t$th round:

**Lemma 3:** [9] Suppose that Assumption 1 holds, then
\[\sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s)\|^2 \right] \leq \frac{2}{\eta} \mathbb{E} [f(y^0) - f(y^m)] + \frac{1}{\eta} \sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s) - v^s\| \right] - (1 - L\eta) \sum_{s=0}^{m-1} \mathbb{E} \left[ \|v^s\|^2 \right].\]

We note that the first term on RHS can be neglected when $m$ is large; the second term measures the effect of biased gradient estimator and can thus be controlled with the step size $\eta$ and distributed smoothness; the third term can be dropped when $\eta \leq 1/L$. A careful analysis leads to the following theorem, whose proof can be found in Appendix C.

**Theorem 6:** Suppose that Assumption 1, 2, 3 and 4a hold with $c_k \leq c$. Then D-SARAH satisfies
\[\left( 1 - \frac{4c^2}{\sigma^2} \right) \mathbb{E} \left[ \|\nabla f(\tilde{x}^{t+1})\|^2 \right] \leq \left( \frac{1}{\sigma m} + \frac{4c^2}{\sigma^2} + \frac{2\eta L}{2 - \eta L} \right) \mathbb{E} \left[ \|\nabla f(\tilde{x}^t)\|^2 \right].\]

When $c < \frac{\sqrt{2}}{\sigma}$, we can choose $\eta = \frac{2(1 - 8c^2/\sigma^2)}{(9 - 8c^2/\sigma^2)L}$ and $m = 2\kappa \frac{9 - 8c^2/\sigma^2}{(1 - 8c^2/\sigma^2)^2}$ in Theorem 6, leading to the following convergence rate:
\[\mathbb{E} \left[ \|\nabla f(\tilde{x}^{t+1})\|^2 \right] \leq \frac{1}{2 - 8c^2/\sigma^2} \mathbb{E} \left[ \|\nabla f(\tilde{x}^t)\|^2 \right].\]

Consequently, following similar discussions as D-SVRG, the communication complexity of finding an $\epsilon$-optimal solution is $\mathcal{O}(\zeta^{-1} \log(1/\epsilon))$, and the runtime complexity is $\mathcal{O}((N/n + \zeta^{-2}k)\zeta^{-1} \log(1/\epsilon))$, where $\zeta = 1 - 2\sqrt{2}c/\sigma$.

D. D-SARAH in the Nonconvex Case (Theorem 5)

The convergence analysis in the nonconvex case is also based upon Lemma 3. Due to lack of convexity, the tighter bound of the second term on RHS adopted in the analysis of Theorem 6 is not available, so a smaller step size $\eta$ is needed to cancel out the second term and the third term, i.e., to make
\[\sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s) - v^s\|^2 \right] - (1 - L\eta) \sum_{s=0}^{m-1} \mathbb{E} \left[ \|v^s\|^2 \right] \leq 0.\]

Formally speaking, we have the following result, proved in Appendix D.

**Theorem 7:** Suppose that Assumption 1 and 4b hold with $c_k \leq c$. By setting the step size
\[\eta \leq \frac{2}{L \left( 1 + \sqrt{1 + 8(m - 1) + 4m(m - 1)c^2/L^2} \right)},\]

For a single outer loop of D-SARAH, it satisfies:
\[\sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s)\|^2 \right] \leq \frac{2}{\eta} \mathbb{E} [f(y^0) - f(y^m)].\]

By setting $\tilde{x}^{t+1} = y^m$, from the above theorem, we have
\[\sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^s)\|^2 \right] \leq \frac{2}{\eta} \mathbb{E} [f(\tilde{x}^t) - f(\tilde{x}^{t+1})].\]

Hence, with $T$ outer loops, we have
\[\frac{1}{Tm} \sum_{t=0}^{T-1} \sum_{s=0}^{m-1} \mathbb{E} \left[ \|\nabla f(y^{k(t)}_t)\|^2 \right] \leq \frac{2}{\eta Tm} (f(\tilde{x}^0) - f^*),\]

where $k(t)$ is the worker index selected in the $t$th round for parameter update. The communication complexity to achieve an $\epsilon$-optimal solution is
\[T = \mathcal{O} \left( 1 + \frac{1}{\eta mc} \right) = \mathcal{O} \left( 1 + \sqrt{m + m^2/c^2/L^2} \cdot \frac{L}{\epsilon} \right),\]

with the choice $\eta = \frac{1}{L \sqrt{m + m^2/c^2/L^2}}$. Per round, the runtime complexity at each worker is $\mathcal{O}(N/n + m)$. By choosing $m = \mathcal{O}(N/n)$, we achieve the runtime complexity
\[\mathcal{O} \left( N/n + \sqrt{N/n + N/n \cdot \frac{c}{L}} \cdot \frac{L}{\epsilon} \right).\]
VII. NUMERICAL EXPERIMENTS

Though the focus of this paper is theoretical, we illustrate the performance of the proposed distributed stochastic variance reduced algorithms in various settings as a proof-of-concept.

A. Logistic Regression in the Strongly Convex Setting

Consider $\ell_2$-regularized logistic regression, where the sample loss is defined as

$$
\ell(x; z_i) = \log (1 + \exp (-b_i a_i^T x)) + \frac{\lambda}{2} \|x\|^2,
$$

with the data $z_i = (a_i, b_i) \in \mathbb{R}^d \times \{\pm 1\}$. We evaluate the performance on the gisette dataset [44] and the rcv1 dataset [45] by splitting the data equally to all workers. We scale the data according to $\max_{i \in [N]} \|a_i\|^2 = 1$, so that the smoothness parameter is estimated as $L = 1/4 + \lambda$. We choose $\lambda = N^{-0.5}$, $N^{-0.75}$ and $N^{-1}$ respectively to illustrate the performance under different condition numbers. We use the optimality gap, defined as $f(\tilde{x}_t) - f^*$, to illustrate the convergence behavior.

For D-SVRG and D-SARAH, the step size is set as $\eta = 1/(2L)$. For D-MiG, although the choice of $w$ in the theory requires knowledge of $c$, we simply ignore it and set $w = 1 + \eta \sigma$, $\theta = 1/2$ and the step size $\eta = 1/(30L)$ to reflect the robustness of the practical performance to parameters. We further use $\tilde{x}_t^{+1} = \frac{1}{n} \sum_{k=1}^n y_k x_t^k$ at the PS, which provides better empirical performance than the random selection rule in Algorithm 1, as seen in Fig. 3. For D-AGD, the step size is set as $\eta = 1/L$ and the momentum parameter is set as $\frac{\sqrt{\pi - 1}}{\sqrt{\pi + 1}}$. Following [3], which sets the number of inner loop iterations as $m = 2N$, we set $m \approx 2N/n$ to ensure the same number of total inner iterations.

We note that such parameters can be further tuned to achieve better trade-off between communication cost and computation cost in practice.

Fig. 1 illustrates the optimality gap of various algorithms with respect to the number of communication rounds with 4 local workers under different conditioning, and Fig. 2 shows the corresponding result with different numbers of local workers when $\lambda = N^{-1}$. The distributed stochastic variance-reduced algorithms outperform distributed AGD significantly. In addition, D-MiG outperforms D-SVRG and D-SARAH when the condition number is large. Fig. 3 shows the optimality gap after a fixed 20 communication rounds with respect to the number of agents on the gisette dataset with $\lambda = N^{-1}$. We observe that the performance of distributed variance-reduced theorems degenerates as the number of agents grows, due to the shrinking size of local dataset, which leads to a larger distributed smoothness parameter $c$.

B. Dealing With Unbalanced Data

We justify the benefit of regularization by evaluating the proposed algorithms under unbalanced data allocation. We assign 50%, 30%, 19.9%, 0.1% percent of data to four workers, respectively, and set $\lambda = N^{-1}$ in the logistic regression loss (15). To deal with unbalanced data, we perform the regularized update, given in (3), on the worker with the least amount of data, and keep the update on the rest of the workers unchanged. A similar regularized update can be conceived for D-SARAH and D-MiG, resulting in regularized variants, D-RSARAH and
D-RMiG. While our theory does not cover them, we still evaluate their numerical performance. We properly set $\mu$ according to the amount of data on this worker as $\mu = 0.1/(0.1\% \cdot N)^{0.5}$. We set the number of iterations at workers $m = 2N$ on all agents. Fig. 4 shows the optimality gap with respect to the number of communication rounds for all algorithms. It can be seen that all unregularized methods fail to converge, and the regularized algorithms still converge, verifying the role of regularization in addressing unbalanced data. It is also worth mentioning that the regularization can be flexibly imposed depending on the local data size, rather than homogeneously across all workers.

C. Performance of D-SARAH in the Nonconvex Setting

We follow the same setting as [29] to evaluate D-SARAH and Distributed Gradient Descent (D-GD) on the gisette dataset with a nonconvex sample loss function:

$$\ell_{\text{ncvx}}(x; z_i) = \log(1 + \exp(-b_i a_i^\top x)) + \lambda \sum_{j=1}^d x_j^2/(1 + x_j^2),$$

which consists of the logistic loss and a non-convex regularizer, where $x_j$ is the $j$th entry of $x$. The smoothness parameter of $\ell_{\text{ncvx}}(x; z_i)$ can be estimated as $L = 1/4 + 2\lambda$. Fig. 5 plots the squared norm the gradient $\|\nabla f(\bar{x}^t)\|^2$ of D-SARAH and D-GD...
with respect to the number of communication rounds. It can be seen that D-SARAH achieves a much lower gradient norm than D-GD with the same number of communication rounds.

VIII. CONCLUSION

In this paper, we have developed a convergence theory for a family of distributed stochastic variance reduced methods without sampling extra data, under a mild distributed smoothness assumption that measures the discrepancy between the local and global loss functions. Convergence guarantees are obtained for distributed stochastic variance reduced methods using accelerations and recursive gradient updates, and for minimizing both strongly convex and nonconvex losses. We also suggest regularization as a means of ensuring convergence when the local data are unbalanced and heterogeneous. We believe the analysis framework is useful for studying distributed variants of other stochastic variance-reduced methods such as Katyusha [26], and proximal variants such as [46].

APPENDIX

We first establish a lemma which will be useful later.  

Lemma 4: When Assumptions 1 and one of the distributed smoothness (Assumption 4a or 4b) hold, we have

$$\mathbb{E}[\|\nabla \ell_z(x_1) - \nabla \ell_z(x_2)\|^2] \leq 2LDf(x_1, x_2)$$

$$\quad + \left\{ \begin{array}{ll} 2cL \left( \|x_1 - x^*\|^2 + \|x_2 - x^*\|^2 \right) & \text{Assumption 4a} \\ cL \|x_1 - x_2\|^2 & \text{Assumption 4b} \end{array} \right.$$ 

for any $\hat{x}$, where the expectation is evaluated over $z$, and $D_f(x_1, x_2) = f(x_1) - f(x_2) - \langle \nabla f(x_2), x_1 - x_2 \rangle$.

Proof: Given $f$ is $L$-smooth and convex, the Bregman divergence $D_f(x_1, x_2)$ is $L$-smooth and convex as a function of $x_1$. When Assumptions 1 and 2 hold, we have

$$0 = D_{\ell_z}(x_2, x_2)$$

$$\leq D_{\ell_z}(x_1, x_2) - \frac{1}{2L} \|\nabla x_1 D_{\ell_z}(x_1, x_2)\|^2$$

$$= D_{\ell_z}(x_1, x_2) - \frac{1}{2L} \|\nabla \ell_z(x_1) - \nabla \ell_z(x_2)\|^2.$$ 

Averaging the above inequality over $z \in \mathcal{M}_k$ gives

$$2L \cdot D_{f_k}(x_1, x_2) \leq \mathbb{E}_z \left[ \|\nabla \ell_z(x_1) - \nabla \ell_z(x_2)\|^2 \right]. \quad (16)$$

To further bound the left-hand side, Assumption 4a allows us to compare $D_f$ and $D_{f_k}$:

$$|D_{f_k}(x_1, x_2) - D_f(x_1, x_2)|$$

$$= \left| D_{f-f_k}(x_1, x^*) + D_{f-f_k}(x^*, x_2) \right.$$ 

$$+ \left( \langle \nabla (f - f_k)(x^* - x_2), x_1 - x^* \rangle \right)$$

$$\leq c \left( \|x_1 - x^*\|^2 + \|x^* - x_2\|^2 + c \|x^* - x_2\| \|x_1 - x^*\| \right)$$

$$\leq c \left( \|x_1 - x^*\|^2 + \|x^* - x_2\|^2 \right).$$

Following similar arguments, using Assumption 4b we obtain a tighter bound by replacing $x^*$ with any $\hat{x}$. In particular, setting $\hat{x} = (x_1 + x_2)/2$ we have

$$|D_{f_k}(x_1, x_2) - D_f(x_1, x_2)| \leq c \|x_1 - x_2\|^2/2.$$

Combining the above estimates with (16) proves the lemma. ■

PROOF FOR D-SVRG

A. Proof of Lemma 1

For $\mathbb{E}[S_1]$, we apply Lemma 4 directly:

$$\mathbb{E}[S_1] = \mathbb{E}\left[ \|\nabla \ell_z(y^*) - \nabla \ell_z(x^*)\|^2 \right]$$

$$\leq 2L \mathbb{E} \left[ |f(y^*) - f(x^*)| + 2cL \mathbb{E} \left[ \|y^* - x^*\|^2 \right] \right],$$

where the inequality follows from the definition of $D_f(y^*, x^*)$, and $\nabla f(x^*) = 0$. For $\mathbb{E}[S_2]$, we have

$$\mathbb{E}[S_2] = \mathbb{E}\left[ \|\nabla \ell_z(x^*) - \nabla \ell_z(y^0) + \nabla f(y^0)\|^2 \right]$$

$$\leq 2E \left[ \|\nabla \ell_z(x^*) - \nabla \ell_z(y^0) - \nabla f_k(y^0)\|^2 \right]$$

$$+ 2E \left[ \|\nabla f_k(x^*) - \nabla f(y^0)\|^2 \right]$$

$$\leq 2E \left[ \|\nabla \ell_z(x^*) - \nabla \ell_z(y^0)\|^2 \right] + 2c^2E \left[ \|y^0 - x^*\|^2 \right]$$

$$\leq 4L \left[ f(y^0) - f(x^*) \right] + c(4L + 2c) \left[ \|y^0 - x^*\|^2 \right].$$

where the first inequality is due to $\|a + b\|^2 \leq 2(a^2 + 2b^2)$, the second inequality follows from evaluating the expectation and Assumption 4a, and the last step uses Lemma 4 again.

For $\mathbb{E}[S_3]$, we have

$$\mathbb{E}[S_3] = -2E \left[ (y^* - x^*, \nabla \ell_z(y^*) - \nabla \ell_z(y^0) + \nabla f(y^0)) \right]$$

$$= 2E \left[ - (y^* - x^*, \nabla f(y^0)) \right]$$

$$- 2E \left[ (y^* - x^*, \nabla (f - f_k)(y^0) - \nabla (f - f_k)(y^0)) \right]$$

$$\leq -2E \left[ f(y^0) - f(x^*) \right]$$

$$+ 2cE \left[ \|y^* - x^*\| \left( \|y^* - x^*\| + \|y^0 - x^*\| \right) \right].$$
Suppose Assumption 1 and 2 hold and 
\[ 2 \nabla x > 3 \nabla y > 4 \]
in (8), we have that it becomes equivalent to 
\[ 2 \eta L y \leq \gamma \nabla x \leq \gamma \nabla y \]
implies 
\[ 2 \gamma \leq 1 \nabla x \leq \nabla y \].
Therefore, combining (17) and (18), we have
\[ \| \nabla f(y) - \nabla f_k(y) \|^2 \leq \| \nabla(y - f_k(y)) \|^2 \]
\[ \leq 2c_2 \| y - x^* \|^2 + 2c_2 \| y^0 - x^* \|^2. \]

Lemma 6: The update rule of D-SARAH satisfies
\[ \mathbb{E} \left[ \| \nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^0) - \nabla f(x^*) \|^2 \right] = \sum_{j=1}^{s} \mathbb{E} \left[ \| v^j - v^{j-1} \|^2 \right] - \sum_{j=1}^{s} \mathbb{E} \left[ \| \nabla f_k(y^j) - \nabla f_k(y^{j-1}) \|^2 \right]. \]

Proof of Theorem 6: By combining Lemmas 5 and 6, we have
\[ \mathbb{E} \left[ \| \nabla f(y) - \nabla f_k(y) + \nabla f_k(y) - \nabla f(x^*) \|^2 \right] \leq \frac{\eta L}{2 - \eta L} \mathbb{E} \left[ \| v^0 \|^2 \right]. \] (17)

By Assumption 4a, we have
\[ \| \nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^0) - \nabla f(x^*) \|^2 \]
\[ \leq 2c_2 \| y^0 - x^* \|^2 + 2c_2 \| y^0 - x^* \|^2. \] (18)

Therefore, combining (17) and (18), we have
\[ \mathbb{E} \left[ \| \nabla f(y^0) - v^0 \|^2 \right] \leq 2 \mathbb{E} \left[ \| \nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^0) - \nabla f(x^*) \|^2 \right] + 2 \mathbb{E} \left[ \| \nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^0) - v^0 \|^2 \right] \]
\[ \leq 4c_2 \mathbb{E} \left[ \| y^0 - x^* \|^2 \right] + 4c_2 \mathbb{E} \left[ \| y^0 - x^* \|^2 \right] + \frac{2\eta L}{2 - \eta L} \mathbb{E} \left[ \| v^0 \|^2 \right]. \]

Substituting it into Lemma 3 gives
\[ \sum_{s=0}^{m-1} \mathbb{E} \left[ \| \nabla f(y^s) \|^2 \right] \leq \frac{2}{\eta} \mathbb{E} \left[ f(y^0) - f(x^*) \right] + \sum_{s=0}^{m-1} \mathbb{E} \left[ \| \nabla f(y^s) - v^s \|^2 \right] \]
\[ \leq \frac{2}{\eta} \mathbb{E} \left[ f(y^0) - f(x^*) \right] + \sum_{s=0}^{m-1} \left( 4c_2 \mathbb{E} \left[ \| y^s - x^* \|^2 \right] + 4c_2 \mathbb{E} \left[ \| y^s - x^* \|^2 \right] + \frac{2\eta L}{2 - \eta L} \mathbb{E} \left[ \| v^0 \|^2 \right] \right). \]

Since \( f \) is \( \sigma \)-strongly convex, we have
\[ 4c_2 \| y^s - x^* \|^2 \leq \frac{4c_2^2}{\sigma^2} \| \nabla f(y^s) \|^2. \] Denote \( y^+ \) as the local update which is selected.
from $y^0, \ldots, y^{m-1}$ uniformly at random. We have
\[
\begin{aligned}
&\left(1 - \frac{4c^2}{\sigma^2}\right) \mathbb{E} \left[\|\nabla f(y^*)\|^2\right] \\
&= \left(1 - \frac{4c^2}{\sigma^2}\right) \frac{1}{m} \sum_{s=0}^{m-1} \mathbb{E} \left[\|\nabla f(y^s)\|^2\right] \\
&\leq \frac{2}{\eta m} \mathbb{E} \left[f(y^0) - f(x^*)\right] + 4c^2 \mathbb{E} \left[\|y^0 - x^*\|^2\right] \\
&\quad + \frac{2\eta L}{2 - \eta L} \mathbb{E} \left[\|\nabla f(y^s)\|^2\right] \\
&\leq \left(\frac{1}{\sigma \eta m} + \frac{4c^2}{\sigma^2} + \frac{2\eta L}{2 - \eta L}\right) \mathbb{E} \left[\|\nabla f(y^s)\|^2\right].
\end{aligned}
\]
Since $\tilde{x}^{k+1}$ is randomly chosen from the local outputs $\{y^k, \ldots, y^n\}$, we have
\[
\begin{aligned}
&\left(1 - \frac{4c^2}{\sigma^2}\right) \mathbb{E} \left[\|\nabla f(\tilde{x}^{k+1})\|^2\right] \\
&\leq \left(\frac{1}{\sigma \eta m} + \frac{4c^2}{\sigma^2} + \frac{2\eta L}{2 - \eta L}\right) \mathbb{E} \left[\|\nabla f(x^k)\|^2\right].
\end{aligned}
\]

D. Proof of Theorem 7

Recall Lemma 3. The theorem follows if
\[
\sum_{s=0}^{m-1} \mathbb{E} \left[\|\nabla f(y^*) - v^s\|^2\right] - (1 - \eta m) \sum_{s=0}^{m-1} \mathbb{E} \left[\|v^s\|^2\right] \leq 0.
\]
The rest of this proof is thus dedicated to show the above inequality.

Note that
\[
\begin{aligned}
&\mathbb{E} \left[\|\nabla f(y^s) - v^s\|^2\right] \\
&\leq 2 \mathbb{E} \left[\|\nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^s) - \nabla f(y^s)\|^2\right] \\
&\quad + 2 \mathbb{E} \left[\|\nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^s) - v^s\|^2\right] \\
&\leq 2c^2 \mathbb{E} \left[\|y^0 - y^s\|^2\right] + 2 \sum_{j=1}^{s} \mathbb{E} \left[\|v^j - v^{j-1}\|^2\right] \\
&\leq 2c^2 s \sum_{j=1}^{s} \mathbb{E} \left[\|y^j - y^{j-1}\|^2\right] + 2 \sum_{j=1}^{s} \mathbb{E} \left[\|v^j - v^{j-1}\|^2\right],
\end{aligned}
\]
where the second inequality follows from Lemma 6 and Assumption 4b, and the third inequality follows from $y^0 - y^s = \sum_{j=1}^{s} (y^j - y^{j-1})$, and the last line follows from the definition. The $L$-smoothness of $\ell_z$ implies that
\[
\|v^j - v^{j-1}\|^2 = \|\nabla \ell_z(y^j) - \nabla \ell_z(y^{j-1})\|^2 \\
\leq L^2 (y^j - y^{j-1})^2 = L^2 \eta^2 (v^{j-1})^2.
\]
So we have
\[
\begin{aligned}
&\sum_{s=0}^{m-1} \mathbb{E} \left[\|\nabla f(y^*) - v^s\|^2\right] - (1 - \eta m) \sum_{s=0}^{m-1} \mathbb{E} \left[\|v^s\|^2\right] \\
&\leq \sum_{s=1}^{m-1} \left(2c^2 s + 2L^2\eta^2\right) \sum_{j=1}^{s} \mathbb{E} \left[\|v^{j-1}\|^2\right] \\
&\quad - (1 - \eta m) \sum_{s=0}^{m-1} \mathbb{E} \left[\|v^s\|^2\right] \\
&\leq \sum_{s=0}^{m-1} \left(m(m-1)c^2\eta^2 + 2L^2\eta^2(m-1) - (1 - \eta)\right) \mathbb{E} \left[\|v^s\|^2\right].
\end{aligned}
\]
Therefore, with $0 < \eta \leq \frac{2}{L(1+\sqrt{1+8(m-1)+4m(m-1)c^2}/L^2)}$, we have $m(m-1)c^2\eta^2 + 2L^2\eta^2(m-1) - (1 - \eta) \leq 0$ and the proof is finished.

E. Proof of Lemma 6

First, we write
\[
\begin{aligned}
&\nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^s) - v^s \\
&= \nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^{s-1}) - v^{s-1} \\
&\quad + [\nabla f_k(y^s) - \nabla f_k(y^{s-1})] - [v^s - v^{s-1}].
\end{aligned}
\]
Let $\mathcal{F}_s$ denote the $\sigma$-algebra generated by all random sample selections in sub-iteration $0, \ldots, s - 1$. We have
\[
\begin{aligned}
&\mathbb{E} \left[\|\nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^s) - v^s\|^2 \mid \mathcal{F}_s\right] \\
&\leq \|\nabla f(y^0) - \nabla f_k(y^0)\|^2 + \|\nabla f_k(y^s) - \nabla f_k(y^{s-1})\|^2 + \mathbb{E} \left[\|v^s - v^{s-1}\|^2 \mid \mathcal{F}_s\right] \\
&\quad + 2 \langle \nabla f(y^0) - \nabla f_k(y^0), \nabla f_k(y^{s-1}) - v^{s-1} \rangle \\
&\quad - 2 \langle \nabla f(y^0) - \nabla f_k(y^0), \nabla f_k(y^{s-1}) - v^{s-1} \rangle \\
&\quad - 2 \langle \nabla f_k(y^s) - \nabla f_k(y^{s-1}), \mathbb{E} \left[v^s - v^{s-1} \mid \mathcal{F}_s\right]\rangle \\
&\quad - 2 \langle \nabla f_k(y^s) - \nabla f_k(y^{s-1}), \mathbb{E} \left[v^s - v^{s-1} \mid \mathcal{F}_s\right]\rangle \\
&\quad + \|\nabla f_k(y^s) - \nabla f_k(y^{s-1})\|^2 + \mathbb{E} \left[\|v^s - v^{s-1}\|^2 \mid \mathcal{F}_s\right],
\end{aligned}
\]
where the second equality follows from Lemma 6 and Assumption 4b, and the third inequality follows from $y^0 - y^s = \sum_{j=1}^{s} y^j - y^{j-1}$, and the last line follows from the definition. The $L$-smoothness of $\ell_z$ implies that
\[
\|v^j - v^{j-1}\|^2 = \|\nabla \ell_z(y^j) - \nabla \ell_z(y^{j-1})\|^2 \\
\leq L^2 (y^j - y^{j-1})^2 = L^2 \eta^2 (v^{j-1})^2.
\]
Taking expectation over $\mathcal{F}_s$ gives
\[
\begin{aligned}
&\mathbb{E} \left[\|\nabla f(y^0) - \nabla f_k(y^0) + \nabla f_k(y^s) - v^s\|^2 \mid \mathcal{F}_s\right] \\
&\quad - \|\nabla f_k(y^s) - \nabla f_k(y^{s-1})\|^2 + \mathbb{E} \left[\|v^s - v^{s-1}\|^2 \mid \mathcal{F}_s\right] \\
&\quad - \|\nabla f_k(y^s) - \nabla f_k(y^{s-1})\|^2 + \mathbb{E} \left[\|v^s - v^{s-1}\|^2 \mid \mathcal{F}_s\right].
\end{aligned}
\]
Telescoping the above equality we obtain the claimed result.

REFERENCES

[1] J. Konečný, B. McMahan, and D. Ramage, “Federated optimization: Distributed optimization beyond the datacenter,” 2015, arXiv:1511.03575.

[2] B. Recht, C. Re, S. Wright, and F. Niu, “Hogwild: A lock-free approach to parallelizing stochastic gradient descent,” in Proc. Advances Neural Inf. Process. Syst., 2011, pp. 693–701.

[3] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in Proc. Advances Neural Inf. Process. Syst., 2013, pp. 315–323.

[4] S. De and T. Goldstein, “Efficient distributed sgd with variance reduction,” in Proc. 31st Conf. Data Mining, 2016, pp. 111–120.

[5] S. J. Reddi, J. Konečný, P. Richtárik, B. Poczos, and A. Smola, “Aide: Fast and communication efficient distributed optimization,” 2016, arXiv:1608.08879.

[6] J. D. Lee, Q. Lin, T. Ma, and T. Yang, “Distributed stochastic variance reduced gradient methods by sampling extra data with replacement,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 4404–4446, 2017.

[7] O. Shamir, “Without-replacement sampling for stochastic gradient methods,” in Proc. Advances Neural Inf. Process. Syst., 2016, pp. 46–54.

[8] J. Wang, W. Wang, and N. Srebro, “Memory and communication efficient distributed stochastic optimization with minibatch prox,” in Proc. Conf. Learn. Theory, 2017, pp. 1882–1919.

[9] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, “Sarah: A novel method for machine learning problems using stochastic recursive gradient,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 2613–2621.

[10] S.-Y. Zhao, R. Xiang, Y.-H. Shi, P. Gao, and W.-J. Li, “SCOPE: Scalable composite optimization for learning on spark,” in Proc. 33rd AAAI Conf. Artif. Intell., 2017, pp. 2928–2934.

[11] C. Renggli, “The convergence of sparsified gradient methods,” in Proc. Advances Neural Inf. Process. Syst., 2017, pp. 1200–1205.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2011.

[13] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[14] O. Shamir, N. Srebro, and T. Zhang, “Communication-efficient distributed optimization using an approximate Newton-type method,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1000–1008.

[15] Y. Zhang and X. Liu, “DSCO: Distributed optimization for self-concordant empirical loss,” Int. Conf. Mach. Learn., pp. 362–370, 2015.

[16] S. Wang, F. Roosta-Khorasani, P. Xu, and M. W. Mahoney, “Giant: Globally improved approximate newton method for distributed optimization,” in Proc. Advances Neural Inf. Process. Syst., 2018, pp. 2338–2348.

[17] Y. Smith, S. Forte, M. Chenxin, M. Takač, M. I. Jordan, and M. Jaggi, “CoCoA: A general framework for communication-efficient distributed optimization,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 8590–8638, 2017.

[18] J. Fan, Y. Guo, and K. Wang, “Communication-efficient accurate statistical estimation,” 2019, arXiv:1906.04870.

[19] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, “Parallelized stochastic gradient descent,” in Proc. Advances Neural Inf. Process. Syst., 2010, pp. 2595–2603.

[20] Y. Zhang, M. J. Wainwright, and J. C. Duchi, “Communication-efficient algorithms for statistical optimization,” in Proc. Advances Neural Inf. Process. Syst., 2012, pp. 1502–1510.

[21] M. Schmidt, N. Le Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” Math. Program., vol. 162, no. 1-2, pp. 83–112, 2017.

[22] A. Defazio, F. Bach, and S. Lacoste-Julien, “Saga: A fast incremental gradient method with support for non-strongly convex composite objectives,” in Proc. Advances Neural Inf. Process. Syst., 2014, pp. 1646–1654.

[23] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” J. Mach. Learn. Res., vol. 14, pp. 567–599, 2013.

[24] Z. Allen-Zhu, “Katyusha: The first direct acceleration of stochastic gradient methods,” in Proc. 49th Annu. ACM SIGACT Symp. Theory Comput., 2017, pp. 1200–1205.

[25] H. Lin, J. Mairal, and Z. Harchaoui, “A universal catalyst for first-order optimization,” in Proc. Advances Neural Inf. Process. Syst., 2015, pp. 3384–3392.

[26] C. Fang, C. J. Li, Z. Lin, and T. Zhang, “Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator,” in Proc. Advances Neural Inf. Process. Syst., 2018, pp. 687–697.

[27] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, “SpiderBoost and momentum: Faster variance reduction algorithms,” in Proc. Advances Neural Inf. Process. Syst., 2019, pp. 2406–2416.

[28] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via gradient quantization and encoding,” in Proc. Advances Neural Inf. Process. Syst., 2017, pp. 1709–1720.

[29] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, “SignSGD: Compressed optimisation for non-convex problems,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 559–568.

[30] S. De and T. Goldstein, “Efficient distributed sgd with variance reduction,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 111–120.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2011.

[32] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[33] O. Shamir, N. Srebro, and T. Zhang, “Communication-efficient distributed optimization using an approximate Newton-type method,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1000–1008.

[34] Y. Zhang and X. Liu, “DSCO: Distributed optimization for self-concordant empirical loss,” Int. Conf. Mach. Learn., pp. 362–370, 2015.

[35] S. Wang, F. Roosta-Khorasani, P. Xu, and M. W. Mahoney, “Giant: Globally improved approximate newton method for distributed optimization,” in Proc. Advances Neural Inf. Process. Syst., 2018, pp. 2338–2348.

[36] Y. Smith, S. Forte, M. Chenxin, M. Takač, M. I. Jordan, and M. Jaggi, “CoCoA: A general framework for communication-efficient distributed optimization,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 8590–8638, 2017.

[37] J. Fan, Y. Guo, and K. Wang, “Communication-efficient accurate statistical estimation,” 2019, arXiv:1906.04870.

[38] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, “Parallelized stochastic gradient descent,” in Proc. Advances Neural Inf. Process. Syst., 2010, pp. 2595–2603.

[39] Y. Zhang, M. J. Wainwright, and J. C. Duchi, “Communication-efficient algorithms for statistical optimization,” in Proc. Advances Neural Inf. Process. Syst., 2012, pp. 1502–1510.

[40] M. Schmidt, N. Le Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” Math. Program., vol. 162, no. 1-2, pp. 83–112, 2017.