Stability and Generalization of Decentralized Stochastic Gradient Descent

Tao Sun[1], **Dongsheng Li**[1*], **and Bao Wang**[2]
[1]College of Computer, National University of Defense Technology, Changsha, Hunan, China.
[2]Scientific Computing & Imaging Institute, University of Utah, USA.
nudtsuntao@163.com, dsli@nudt.edu.cn, wangbaonj@gmail.com

Abstract

The stability and generalization of stochastic gradient-based methods provide valuable insights into understanding the algorithmic performance of machine learning models. As the main workhorse for deep learning, stochastic gradient descent has received a considerable amount of studies. Nevertheless, the community paid little attention to its decentralized variants. In this paper, we provide a novel formulation of the decentralized stochastic gradient descent. Leveraging this formulation together with (non)convex optimization theory, we establish the first stability and generalization guarantees for the decentralized stochastic gradient descent. Our theoretical results are built on top of a few common and mild assumptions and reveal that the decentralization deteriorates the stability of SGD for the first time. We verify our theoretical findings by using a variety of decentralized settings and benchmark machine learning models.

Introduction

The great success of deep learning (LeCun, Bengio, and Hinton 2015) gives impetus to the development of stochastic gradient descent (SGD) (Robbins and Monro 1951) and its variants (Nemirovski et al. 2009; Duchi, Hazan, and Singer 2011; Rakhlin, Shamir, and Sridharan 2012; Kingma and Ba 2014; Wang et al. 2020). Although the convergence results of SGD are abundant, the effects caused by the training data is absent. To this end, the generalization error (Hardt, Recht, and Singer 2016; Lin, Camoriano, and Rosasco 2016; Bousquet and Elisseeff 2002; Bottou and Bousquet 2008) is developed as an alternative method to analyze SGD. The generalization bound reveals the performance of stochastic algorithms and characterizes how the training data and stochastic algorithm jointly affect the target machine learning model. To mathematically describe generalization, Hardt, Recht, and Singer (2016); Bousquet and Elisseeff (2002); Elisseeff, Evgeniou, and Pontil (2005) introduce the algorithmic stability for SGD, which mainly depends on the landscape of the underlying loss function, to study the generalization bound of SGD. The stability theory of SGD has been further developed (Charles and Papailiopoulos 2018; Kuzborskij and Lampert 2018; Lei and Ying 2020).

SGD has already been widely used in parallel and distributed settings (Agarwal and Duchi 2011; Dekel et al. 2012; Recht et al. 2011), e.g., the decentralized SGD (D-SGD) (Ram, Nedić, and Veeravalli 2010b; Lan, Lee, and Zhou 2020; Srivastava and Nedic 2011; Lian et al. 2017). D-SGD is implemented without a centralized parameter server, and all nodes are connected through an undirected graph. Compared to the centralized SGD, the decentralized one requires much less communication with the busiest node (Lian et al. 2017), accelerating the whole computational system.

From the theoretical viewpoint, although there exist plenty of convergence analysis of D-SGD (Sirb and Ye 2016; Lan, Lee, and Zhou 2020; Lian et al. 2017, 2018), the stability and generalization analysis of D-SGD remains rare.

Contributions

In this paper, we establish the first theoretical result on the stability and generalization of the D-SGD. We elaborate on our contributions below.

1. *Stability of D-SGD*: We provide the uniform stability of D-SGD in the general convex, strongly convex, and nonconvex cases. Our theory shows that besides the learning rate, data size, and iteration number, the stability and generalization of D-SGD are also dependent on the connected graph structure. To the best of our knowledge, our result is the first theoretical stability guarantee for D-SGD. In the general convex setting, we also present the stability of D-SGD in terms of the ergodic average instead of the last iteration for the excess generalization analysis.
2. *Computational errors for D-SGD with convexity and projection*: We consider more general schemes of D-SGD, that is, D-SGD with projection. In the previous work (Ram, Nedić, and Veeravalli 2010b), to get the convergence rate, the authors need to make additional assumptions on the graph ([Assumptions 2 and 3, (Ram, Nedić, and Veeravalli 2010b)]). In this paper, we remove these assumptions, and we present the computational errors of D-SGD with projections in the strongly convex setting.
3. *Generalization bounds for D-SGD with convexity*: We derive (excess) generalization bounds for convex D-SGD. The excess generalization is controlled by the computational error and the generalization bound, which can be directly obtained from the stability.

*Corresponding author.

4. *Numerical results*: We numerically verify our theoretical results by using various benchmark machine learning models, ranging from strongly convex and convex to non-convex settings, in different decentralized settings.

Prior Art

In this section, we briefly review two kinds of related works: decentralized optimization and stability and generalization analysis of SGD.

Decentralized and distributed optimization Decentralized algorithms arise in calculating the mean of data distributed over multiple sensors (Boyd et al. 2005; Olfati-Saber, Fax, and Murray 2007). The decentralized (sub)gradient descent (DGD) algorithms are propose and studied by (Nedic and Ozdaglar 2009; Yuan, Ling, and Yin 2016). Recently, DGD has been generalized to the stochastic settings. With a local Poisson clock assumption on each agent, Ram, Nedić, and Veeravalli (2010a) proposes an asynchronous gossip algorithm. The decentralized algorithm with a random communication graph is proposed in (Srivastava and Nedic 2011; Ram, Nedić, and Veeravalli 2010b). Sirb and Ye (2016); Lan, Lee, and Zhou (2020); Lian et al. (2017) consider the randomness caused by the stochastic gradients and proposed the decentralized SGD (D-SGD). The complexity analysis of D-SGD has been done in (Sirb and Ye 2016). In (Lan, Lee, and Zhou 2020), the authors propose another kind of D-SGD that leverages dual information, and provide the related computational complexity. In the paper (Lian et al. 2017), the authors show the advantage of D-SGD compared to the centralized SGD. In a recent paper (Lian et al. 2018), the authors developed asynchronous D-SGD with theoretical convergence guarantees. The biased decentralized SGD is proposed and studied by (Sun et al. 2019). In (Richards et al. 2020), the authors studied the stability for a non-fully decentralized training method, in which each node needs to communicate extra gradient information. Paper (Richards et al. 2020) is closed to ours, but we consider the DSGD, which is different from the algorithm investigated by (Richards et al. 2020) and more general. Further more, we studied the non-convex settings.

Stability and Generalization of SGD In (Shalev-Shwartz et al. 2010), on-average stability is proposed and further studied by Kuzborskij and Lampert (2018). The uniform stability of empirical risk minimization (ERM) under strongly convex objectives is considered by Bousquet and Elisseeff (2002). Extended results are proved with the pointwise-hypothesis assumption, which shows that a class of learning algorithms is convergent with global optimum (Charles and Papailiopoulos 2018). In order to prove uniform stability of SGD, Hardt, Recht, and Singer (2016) reformulate SGD as a contractive iteration. In (Lei and Ying 2020), a new stability notion is proposed to remove the bounded gradient assumptions. In (Bottou and Bousquet 2008), the authors establish a framework for the generalization performance of SGD. Hardt, Recht, and Singer (2016) connects the uniform stability with generalization error. The generalization errors with strong convexity are established in (Hardt, Recht, and Singer 2016; Lin, Camoriano, and Rosasco 2016). The stability and generalization are also studied for the Langevin dynamics (Li, Luo, and Qiao 2019; Mou et al. 2018).

Setup

This part contains preliminaries and mathematical descriptions of our problem. Analyzing the stability of D-SGD is more complicated than that of SGD due to the challenge arises from the mixing matrix in D-SGD. We cannot directly adapt the analysis for SGD to D-SGD. To this end, we reformulate D-SGD as an operator iteration with an error term, which is followed by bounding the error in each iteration.

Stability and Generalization

The population risk minimization is an important model in machine learning and statistics, whose mathematical formulation reads as

$$\min_{\mathbf{x}\in\mathbb{R}^d} R(\mathbf{x}) := \mathbb{E}_{\xi\sim\mathcal{D}} f(\mathbf{x};\xi),$$

where $f(\mathbf{x};\xi)$ denotes the loss of the model associated with data ξ and $\mathcal{D}$ is the data distribution. Due to the fact that $\mathcal{D}$ is usually unknown or very complicated, we consider the following surrogate ERM

$$\min_{\mathbf{x}\in\mathbb{R}^d} R_S(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x};\xi_i),$$

where $S := \{\xi_1, \xi_2, \ldots, \xi_N\}$ and $\xi_i \sim \mathcal{D}$ is a given data.

For a specific stochastic algorithm $\mathcal{A}$ act on S with output $\mathcal{A}(S)$, the *generalization error* of $\mathcal{A}$ is defined as $\epsilon_{\text{gen}} := \mathbb{E}_{S,\mathcal{A}}[R(\mathcal{A}(S)) - R_S(\mathcal{A}(S))]$. Here, the expectation is taken over the algorithm and the data. The generalization bound reflects the joint effects caused by the data S and the algorithm $\mathcal{A}$. We are also interested in the *excess generalization error*, which is defined as $\epsilon_{\text{ex-gen}} := \mathbb{E}_{S,\mathcal{A}}[R(\mathcal{A}(S)) - R(\mathbf{x}^*)]$, where $\mathbf{x}^*$ is the minimizer of R. Let $\overline{\mathbf{x}}$ be the minimizer of R_S. Due to the unbiased expectation of the data S, we have $\mathbb{E}_S[R_S(\mathbf{x}^*)] = \mathbb{E}[R(\mathbf{x}^*)]$. Thus, Bottou and Bousquet (2008) point out $\epsilon_{\text{ex-gen}}$ can be decomposed as follows

$$\mathbb{E}_{S,\mathcal{A}}[R(\mathcal{A}(S)) - R(\mathbf{x}^*)] = \underbrace{\mathbb{E}_{S,\mathcal{A}}[R(\mathcal{A}(S)) - R_S(\mathcal{A}(S))]}_{\text{generalization error}}$$
$$+ \underbrace{\mathbb{E}_{S,\mathcal{A}}[R_S(\mathcal{A}(S)) - R_S(\overline{\mathbf{x}})]}_{\text{optimization error}} + \underbrace{\mathbb{E}_{S,\mathcal{A}}[R_S(\overline{\mathbf{x}}) - R_S(\mathbf{x}^*)]}_{\text{test error}}.$$

Notice that $R_S(\overline{\mathbf{x}}) \le R_S(\mathbf{x}^*)$, therefore

$$\epsilon_{\text{ex-gen}} \le \epsilon_{\text{gen}} + \mathbb{E}_{S,\mathcal{A}}[R_S(\mathcal{A}(S)) - R_S(\overline{\mathbf{x}})].$$

The *uniform stability* is used to bound the generalization error of a given algorithm $\mathcal{A}$ (Hardt, Recht, and Singer 2016; Elisseeff, Evgeniou, and Pontil 2005).

Definition 1 *We say that the randomized algorithm $\mathcal{A}$ is ϵ-*uniformly stable *if for any two data sets S, S' with n samples that differ in one example, we have*

$$\sup_{\xi} \mathbb{E}_{\mathcal{A}}\left[f(\mathcal{A}(S);\xi) - f(\mathcal{A}(S');\xi)\right] \le \epsilon.$$

It has been proved that the uniform stability directly implies the generalization bound.

Lemma 1 ((Hardt, Recht, and Singer 2016)) *Let $\mathcal{A}$ be ϵ-*uniformly stable*, it follows $|\mathbb{E}_{S,\mathcal{A}}[R(\mathcal{A}(S)) - R_S(\mathcal{A}(S))]| \leq \epsilon$.*

Thus, to get the generalization bound of a random algorithm, we just need to compute the uniform stability bound ϵ.

Problem Formulation

Notation: We use the following notations throughout the paper. We denote the ℓ_2 norm of $\mathbf{x} \in \mathbb{R}^d$ as $\|\mathbf{x}\|$. For a matrix $\mathbf{A}$, $\mathbf{A}^\top$ denotes its transpose, we denote the spectral norm of $\mathbf{A}$ as $\|\mathbf{A}\|_{\text{op}}$. Given another matrix $\mathbf{B}$, $\mathbf{A} \succ \mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is positive define; and $\mathbf{A} \succeq \mathbf{B}$ means $\mathbf{A} - \mathbf{B}$ is positive semidefinite. The identity matrix is defined as $\mathbb{I}$. We use $\mathbb{E}[\cdot]$ to denote the expectation of $\cdot$ with respect to the underlying probability space. For two positive constants a and b, we denote $a = \mathcal{O}(b)$ if there exists $C > 0$ such that $a \leq Cb$, and $\widetilde{\mathcal{O}}(b)$ hides a logarithmic factor of b.

Let $\mathcal{D}_i = \{\xi_{l(i)}\}_{1 \leq l \leq n}$ $(1 \leq i \leq m)$ denote the data stored in the ith client, which follow the same distribution of $\mathcal{D}$ [1]. In this paper, we consider solving the objective function (1) by the DGD, where

$$f(\mathbf{x}) := \frac{1}{mn} \sum_{i=1}^{m} \sum_{l=1}^{n} f(\mathbf{x}; \xi_{l(i)}). \tag{1}$$

Note that (1) is a decentralized approximation to the following population risk function

$$F(\mathbf{x}) := \mathbb{E}_{\xi \sim \mathcal{D}} f(\mathbf{x}; \xi). \tag{2}$$

To distinguish from the objective functions in the last subsection, we use f rather than R_S here. The decentralized optimization is usually associated with a mixing matrix, which is designed by the users according to a given graph structure. In particular, we consider the connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, ..., M\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ with edge $(i, l) \in \mathcal{E}$ represents the communication link between nodes i and l. Before proceeding, let us recall the definition of the mixing matrix.

Definition 2 (Mixing matrix) *For any given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the mixing matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{M \times M}$ is defined on the edge set $\mathcal{V}$ that satisfies: (1) If $i \neq j$ and $(i, j) \notin \mathcal{E}$, then $w_{ij} = 0$; otherwise, $w_{ij} > 0$; (2) $\mathbf{W} = \mathbf{W}^\top$; (3) $\text{null}\{\mathbb{I} - \mathbf{W}\} = \text{span}\{\mathbf{1}\}$; (4) $\mathbb{I} \succeq \mathbf{W} \succ -\mathbb{I}$.*

Note that $\mathbf{W}$ is a doubly stochastic matrix (Marshall, Olkin, and Arnold 1979), and the mixing matrix is non-unique for a given graph. Several common examples for $\mathbf{W}$ include the Laplacian matrix and the maximum-degree matrix (Boyd, Diaconis, and Xiao 2004). A crucial constant that characterizes the mixing matrix is

$$\lambda := \max\{|\lambda_2|, |\lambda_m(\mathbf{W})|\},$$

[1] For simplicity, we assume all clients have the same amount of samples.

Algorithm 1 Decentralized Stochastic Gradient Descent (D-SGD)

Require: $(\alpha_t > 0)_{t \geq 0}$, initialization $\mathbf{x}^0$
  for node $i = 1, 2, \ldots, m$
    for $t = 1, 2, \ldots$
      updates local parameter as (3) and (4)
      $\mathbf{x}^t = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}^t(i)$
    end for
  end for

where λ_i denotes the ith largest eigenvalue of $\mathbf{W} \in \mathbb{R}^{m \times m}$. The definition of the mixing matrix implies that $0 \leq \lambda < 1$. In [Corollary 1.14., (Montenegro and Tetali 2006)], the authors proved the following result.

Lemma 2 *Let $\mathbf{P} \in \mathbb{R}^{m \times m}$ be the matrix whose elements are all $1/m$. Given any $k \in \mathbb{Z}^+$, the mixing matrix $\mathbf{W} \in \mathbb{R}^{m \times m}$ satisfies*

$$\|\mathbf{W}^k - \mathbf{P}\|_{\text{op}} \leq \lambda^k.$$

Note the fact that the stationary distribution of an irreducible aperiodic finite Markov chain is uniform if and only if its transition matrix is doubly stochastic. Thus, $\mathbf{W}$ corresponds to some Markov chain's transition matrix, and the parameter $0 \leq \lambda < 1$ characterizes the speed of convergence to the stationary state.

We consider a general decentralized stochastic gradient descent with projection, which carries out in the following manner: in the t-th iteration, 1) client i applies an approximate copy $\mathbf{x}^t(i) \in \mathbb{R}^d$ to calculate a unbiased gradient estimate $\nabla f(\mathbf{x}^t(i); \xi_{j_t(i)})$, where $j_t(i) \in \mathbb{Z}^+$ is the local random index; 2) client i replaces its local parameters with the weighted average of its neighbors, i.e.,

$$\tilde{\mathbf{x}}^t(i) = \sum_{l \in \mathcal{N}(i)} w_{i,l} \mathbf{x}^t(l); \tag{3}$$

3) client i updates its parameters as

$$\mathbf{x}^{t+1}(i) = \mathbf{Proj}_V\Big(\tilde{\mathbf{x}}^t(i) - \alpha_t \nabla f(\mathbf{x}^t(i); \xi_{j_t(i)})\Big) \tag{4}$$

with learning rate $\alpha_t > 0$, and $\mathbf{Proj}_V(\cdot)$ stands for projecting the quantity $\cdot$ into the space V. We stress that, in practice, we do not need to compute the average $\mathbf{x}^t = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}^t(i)$ in each iteration, and we take the average only in the last iteration.

In the following, we draw necessary assumptions, which are all common and widely used in the nonconvex analysis community.

Assumption 1 *The loss function $f(\mathbf{x}; \xi)$ is nonnegative and differentiable with respect to $\mathbf{x}$, and $\nabla f(\mathbf{x}; \xi)$ is bounded by the constant B over V, i.e., $\max_{\mathbf{x} \in V, \xi \sim \mathcal{D}} \|\nabla f(\mathbf{x}; \xi)\| \leq B$.*

Assumption 1 implies that $|f(\mathbf{x}; \xi) - f(\mathbf{y}; \xi)| \leq B\|\mathbf{x} - \mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in V$ and any $\xi \sim \mathcal{D}$.

Assumption 2 *The gradient of $f(\mathbf{x};\xi)$ with respect to $\mathbf{x}$ is L-Lipschitz, i.e., $\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{y};\xi)\| \leq L\|\mathbf{x}-\mathbf{y}\|$, for all $\mathbf{x},\mathbf{y} \in V$ and any $\xi \sim \mathcal{D}$.*

Assumption 3 *The set V forms a closed ball in $\mathbb{R}^d$.*

Compared with the scheme presented in (Lian et al. 2017), our algorithm accommodates a projection after each update in each client. When $\nabla f(\mathbf{x};\xi)$ is non-strongly convex, V can be set as the full space and Algorithm 1 reduces to the scheme given in (Lian et al. 2017), whose convergence has been well studied. Such a projection is more general and is necessary for the strongly convex analysis; we explain this necessary claim as follows: if f is ν-strongly convex, then $\|\nabla f(\mathbf{x})\|^2 \geq \nu\|\mathbf{x}-\mathbf{x}^*\|^2$ with $\mathbf{x}^*$ being the minimizer of $f(\mathbf{x})$ (Karimi, Nutini, and Schmidt 2016). Thus, when $\mathbf{x}$ is far from $\mathbf{x}^*$, the gradient is unbounded, which breaks Assumption 1. However, with the projection procedure, D-SGD (Algorithm 1) actually minimizes function (1) over the set V. The strong convexity gives us $f(\mathbf{x})-f(\mathbf{x}^*) \geq \frac{\nu}{2}\|\mathbf{x}-\mathbf{x}^*\|^2$, which indicates $\mathbf{x}^* \in \mathbf{B}(\mathbf{0}, \sqrt{2(f(\mathbf{x}^0)-\min f)/\nu})$. Thus, when the radius of V is large enough, the projection does not change the output of D-SGD.

Stability of D-SGD

In this section, we prove the stability theory for D-SGD in strongly convex, convex, and nonconvex settings.

General Convexity

This part contains the stability result of D-SGD when $f(\cdot;\xi)$ is generally convex.

Theorem 1 *Let $f(\cdot;\xi)$ be convex and Assumptions 1, 2, 3 hold. If the step size $\alpha_t \leq 2/L$, then D-SGD satisfies the uniform stability with*

$$\epsilon_{\text{stab}} \leq \frac{2B^2\sum_{t=1}^{T-1}\alpha_t}{mn} + 4B^2\sum_{t=1}^{T-1}\Big[(1+\alpha_t B)\sum_{j=0}^{t-1}\alpha_j\lambda^{t-1-j}\Big].$$

Compared to the results of minimizing (1) by using centralized SGD with step sizes $(\alpha_t)_{t\geq 1}$ [Theorem 3.8, (Hardt, Recht, and Singer 2016)], which yields the uniformly stable bound as $2B^2\sum_{t=1}^{T-1}\alpha_t/(mn)$. Theorem 1 shows that D-SGD suffers from an additional term $4B^2\sum_{t=1}^{T-1}(1+\alpha_t B)\sum_{j=0}^{t-1}\alpha_j\lambda^{t-1-j}$, which does not vanish when $\lambda > 0$.

If we set $\alpha_t = 1/(t+1)$, with Lemma 3, it is easy to check that $\epsilon_{\text{stab}} = \mathcal{O}(\frac{\ln T}{mn} + C_\lambda \ln T)$; However, if we use a constant learning rate, (i.e., $\alpha_t \equiv \alpha$), when $0 < \lambda < 1$, we have $4B^2\sum_{t=1}^{T-1}(1+\alpha_t B)\sum_{j=0}^{t-1}\alpha_j\lambda^{t-1-j} = \mathcal{O}(\frac{\alpha T}{1-\lambda})$ and $\epsilon_{\text{stab}} = \mathcal{O}(\frac{\alpha T}{1-\lambda} + \frac{\alpha T}{mn})$. The result indicates that although decentralization reduces the busiest node's communication, it hurts the stability.

Theorem 1 provides the uniform stability for the last-iterate of D-SGD. However, the computational error of D-SGD in general convexity case uses the following average

$$\text{ave}(\mathbf{x}^T) := \frac{\sum_{t=1}^{T-1}\alpha_t\mathbf{x}^t}{\sum_{t=1}^{T-1}\alpha_t}. \tag{5}$$

Such a mismatch leads to the difficulty in characterizing the excess generalization bound. It is thus necessary describe to the uniform stability in terms of $\text{ave}(\mathbf{x}^T)$. To this end, we consider that D-SGD outputs $\text{ave}(\mathbf{x}^t)$ instead of $\mathbf{x}^t$ in the t-th iteration. The uniform stability, in this case, is defined as $\epsilon_{\text{ave-stab}}$, and we have the following result.

Proposition 1 *Let $f(\cdot;\xi)$ be convex and Assumptions 1, 2, 3 hold. If the step size $\alpha_t \equiv \alpha \leq 2/L$, the uniform stability $\epsilon_{\text{ave-stab}}$, in terms of ave$(\mathbf{x}^t)$, satisfies*

$$\epsilon_{\text{ave-stab}} \leq \frac{2B^2\alpha(t-1)}{mn} + \frac{4\alpha B^2(1+\alpha B)(t-1)}{1-\lambda}1_{\lambda\neq 1}.$$

Furthermore, if the step size is chosen as $\alpha_t = 1/(t+1)$, we have

$$\epsilon_{\text{ave-stab}} \leq \frac{B^2\ln T}{mn} + \frac{4B^2(1+B)}{\ln(T+1)}1_{\lambda\neq 1}.$$

Unlike the uniform stability for $\mathbf{x}^T$, the average turns out to be a very complicated one. We thus just present two classical kinds of step size.

Strong Convexity

In the convex setting, for a fixed iteration number T, as the data size mn increases and λ decreases, ϵ_{stab} gets smaller for both diminishing and constant learning rates. However, similar to SGD, D-SGD also fails to have ϵ_{stab} under control when T increases. This drawback does not exist in the strongly convex setting.

Strongly convex loss functions appear in the ℓ_2 regularized machine learning models. As mentioned in Section 2, to guarantee the bounded gradient, the set V should be restricted to a closed ball. We formulate the uniform stability results in this case in Theorem 2.

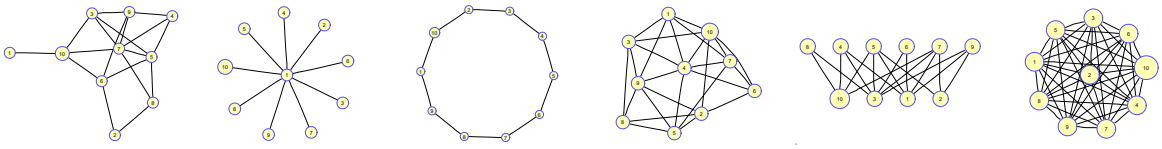
Theorem 2 *Let $f(\cdot;\xi)$ be ν-strongly convex and Assumptions 1, 2, 3 hold. If the step size $\alpha_t \equiv \alpha \leq 1/L$, then D-SGD satisfies the uniform stability with*

$$\epsilon_{\text{stab}} \leq \frac{2B^2}{mn\nu} + \frac{4(1+\alpha B)B^2}{\nu}\frac{1_{\lambda\neq 0}}{1-\lambda}.$$

Furthermore, if the step size $\alpha_t = \frac{1}{\nu(t+1)}$, it holds that

$$\epsilon_{\text{stab}} \leq \frac{2B^2}{mn\nu} + 4(1+\frac{B}{\nu})\frac{B^2}{\nu}\frac{1_{\lambda\neq 0}}{1-\lambda}.$$

The uniformly stability bound for SGD with strong convexity is $2B^2/(mn\nu)$ (Hardt, Recht, and Singer 2016), which is smaller than the one of D-SGD. From Theorem 2, we see that the uniform stability bound of D-SGD is independent on the iterative number T. Moreover, D-SGD enjoys a smaller uniformly stable bound when the data size mn is larger and λ is smaller.



(a) Random (b) Star (c) Cycle (d) k-NNG (e) Bipartite (f) Complete

Figure 1: Structures of the connected graph used in our numerical tests.

Nonconvexity

We now present the stability result for nonconvex loss functions.

Theorem 3 *Suppose Assumptions 1, 2, 3 hold and* $\sup_{\mathbf{x}\in V,\xi} f(\mathbf{x};\xi) \leq 1$. *For any* T, *if the step size* $\alpha_t \leq c/(t+1)$ *and* c *is small enough, then D-SGD satisfies the uniform stability with* $\epsilon_{\text{stab}} \leq \frac{c^{\frac{1}{1+cL}} T^{\frac{cL}{1+cL}}}{mn} + c^{\frac{1}{1+cL}} \left[\frac{2B^2 cL}{mn} + 4(1+cB)B^2 LC_\lambda\right] T^{\frac{cL}{1+cL}}$.

Without the convexity assumption, the uniform stable bound of D-SGD deteriorated. Theorem 3 shows that $\epsilon_{\text{stab}} = \mathcal{O}((1+C_\lambda)T^{\frac{cL}{1+cL}}/(mn))$, which is much larger than the bounds in the convex case ($\mathcal{O}(\ln T/(mn) + C_\lambda \ln T)$).

Excess Generalization for Convex Problems

In the nonconvex case, the optimization error of the function value is unclear. Thus, the excess generalization error is absent. We are also interested in the excess generalization associated with the computational optimization error. The existing computational errors of Algorithm 1 require extra assumptions on the graph for projections. However, these assumptions may fail to hold in many applications. Thus, we first present the optimization error of D-SGD when $f(\mathbf{x};\xi)$ is convex without extra assumptions.

Optimization Error of Convex D-SGD

This part consists of optimization errors of D-SGD for convex and strongly convex settings. Assume $\mathbf{x}^*$ is the minimizer of $f(\mathbf{x})$ over the set V, i.e., $f(\mathbf{x}^*) = \min_{\mathbf{x}\in V} f(\mathbf{x})$.

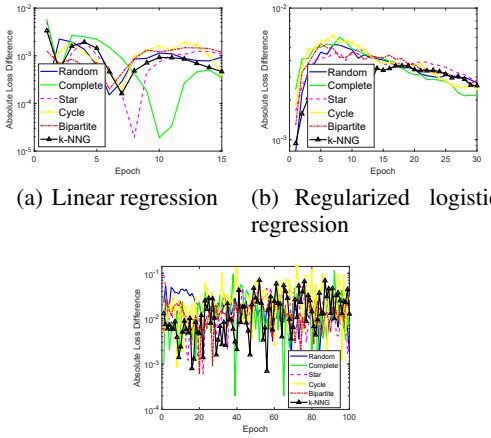
Lemma 3 *Let* $f(\cdot;\xi)$ *be convex and Assumptions 1, 2 hold, and let* $(\mathbf{x}^t)_{1\leq t\leq T}$ *be the sequence generated by D-SGD. Then*

$$\mathbb{E}(f(\text{ave}(\mathbf{x}^T)) - f(\mathbf{x}^*)) \leq \frac{\|\mathbf{x}^1 - \mathbf{x}^*\|^2}{\sum_{t=1}^{T-1}\alpha_t} + \frac{2B^2\sum_{t=1}^{T-1}\alpha_t^2}{m\sum_{t=1}^{T-1}\alpha_t} + 8LrBM(T) + 2\lambda^2 B^2 M(T)^2,$$

where $M(T) := \max_{1\leq t\leq T-1}\{\sum_{j=0}^{t-1}\alpha_j\lambda^{t-1-j}\}$ *and* r *is the radius of* V.

It is worth mentioning that the optimization error is established on the average point $\text{ave}(\mathbf{x}^T)$ for technical reasons.

In the following, we provide the results for the strongly convex setting.



(a) Linear regression (b) Regularized logistic regression

(c) ResNet-20 classification on Cifar-10

Figure 2: Comparison of the absolute loss difference under different graphs. (a), (b) and (c) correspond to the general convex, strongly convex, and nonconvex cases, respectively. In the strongly convex case, the curves become stable after enough iterations for all graphs. In the general convex case, the absolute loss difference oscillates and inferior to the strongly convex case. D-SGD performs worst in the nonconvex tests in terms of stability.

Lemma 4 *Let* $f(\cdot;\xi)$ *be* ν*-strongly convex and Assumptions 1, 2, 3 hold, and let* V *be a closed ball with radius* $r > 0$, *and let* $(\mathbf{x}^t)_{1\leq t\leq T}$ *be the sequence generated by D-SGD. When* $\alpha_t \equiv \alpha > 0$, *then*

$$\mathbb{E}\|\mathbf{x}^T - \mathbf{x}^*\|^2 \leq (1-2\alpha\nu)^{T-1}\|\mathbf{x}^1 - \mathbf{x}^*\|^2 + \left(\frac{4\alpha LrB}{(1-\lambda)\nu} + \frac{\lambda^2 B^2\alpha}{m(1-\lambda)^2\nu}\right)1_{\lambda\neq 0},$$

where $1_{\lambda\neq 0} = 1$ *when* $\lambda \neq 0$, *and* $1_{\lambda\neq 0} = 0$ *when* $\lambda = 0$. *When* $\alpha_t = 1/(2\nu(t+1))$, *it then follows*

$$\mathbb{E}\|\mathbf{x}^T - \mathbf{x}^*\|^2 \leq \frac{\|\mathbf{x}^1 - \mathbf{x}^*\|^2}{T-1} + \frac{D_\lambda \ln T}{T-1},$$

where $D_\lambda := \frac{B^2}{2\nu^2} + \frac{\lambda^2 B^2 C_\lambda^2}{2\nu^2 m} + \frac{2LrBC_\lambda}{\nu^2}$ *and*

$$C_\lambda := \begin{cases} \ln\frac{1}{\lambda}\frac{\lambda^{\ln\frac{1}{\lambda}}}{\lambda} + \frac{\ln^2\frac{1}{\lambda}}{16\lambda}\lambda^{\frac{\ln\frac{1}{\lambda}}{8}} + \frac{2}{\lambda\ln\frac{1}{\lambda}} & \lambda \neq 0, \\ 0, & \lambda = 0. \end{cases}$$

The result shows that D-SGD with projection converges sublinearly in the strongly convex case. To reach an $\epsilon > 0$

error, we shall set the iteration number as $\widetilde{\mathcal{O}}(1/\epsilon)$. Our result coincides with the existing convergence results of S-GD with strong convexity (Rakhlin, Shamir, and Sridharan 2012). What is different is that D-SGD is affected by the parameter λ, which is determined by the structure of the connected graph[2].

General Convexity

Notice that the computational error of D-SGD, in this case, is described by $\text{ave}(\mathbf{x}^T)$. Thus, we need to estimate the generalization bound about $\text{ave}(\mathbf{x}^T)$.

Theorem 4 *Let $f(\cdot;\xi)$ be convex and Assumptions 1, 2, 3 hold. If the step size $\alpha_t \equiv \alpha \leq 2/L$, then the average output* (5) *obeys the following generalization bound*

$$\epsilon_{\text{ex-gen}} \leq \frac{2B^2\alpha(t-1)}{mn} + \frac{4\alpha B^2(1+\alpha B)(t-1)}{1-\lambda}1_{\lambda\neq 1} + \frac{4r^2}{(T-1)\alpha} + \frac{2B^2\alpha}{m} + \frac{8LrB\alpha}{1-\lambda}1_{\lambda\neq 1} + \frac{2\lambda^2B^2\alpha^2}{(1-\lambda)^2}.$$

Furthermore, if the step size is chosen as $\alpha_t = 1/(t+1)$, we have

$$\epsilon_{\text{ex-gen}} \leq \frac{B^2\ln T}{mn} + \frac{4B^2(1+B)}{\ln(T+1)}1_{\lambda\neq 1} + 2\lambda^2B^2C_\lambda^2 + \frac{4r^2}{\ln(T+1)} + \frac{4B^2}{m\ln(T+1)} + 8LrBC_\lambda 1_{\lambda\neq 1}.$$

Strong Convexity

Now, we present the excess generalization of D-SGD under strong convexity.

Theorem 5 *Let $f(\cdot;\xi)$ be ν-strongly convex and Assumptions 1, 2, 3 hold. If the step size $\alpha_t \equiv \alpha \leq 1/L$, the excess generalization bound is*

$$\epsilon_{\text{ex-gen}} \leq \frac{2B^2}{mn\nu} + \frac{4(1+\alpha B)B^2}{\nu}\frac{1_{\lambda\neq 0}}{1-\lambda} + B\sqrt{(1-2\alpha\nu)^{T-1}4r^2 + (\frac{4\alpha LrB}{(1-\lambda)\nu} + \frac{\lambda^2B^2\alpha}{m(1-\lambda)^2\nu})1_{\lambda\neq 0}}.$$

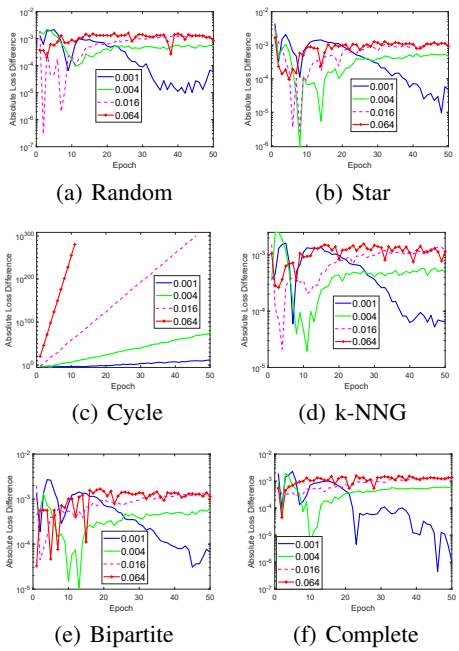
Furthermore, if the step size $\alpha_t = 1/(\nu(t+1))$, the excess generalization bound is

$$\epsilon_{\text{ex-gen}} \leq \frac{2B^2}{mn\nu} + \frac{4(\nu+B)B^2}{\nu^2}\frac{1_{\lambda\neq 0}}{1-\lambda} + B\sqrt{\frac{4r^2}{T-1} + \frac{D_\lambda\ln T}{T-1}}.$$

Numerical Results

We numerically verify our theoretical findings in this section, with a focus on testing three kinds of models, namely, strongly convex, convex, and nonconvex. For all the above three scenarios, we set the number of nodes m to 10 and conduct two kinds of experiments: the first kind of experiments is to verify the stability and generalization results. Given a fixed graph, we use two sets of samples that are of the same amount, and the entries are differing by a small portion. We compare the training loss and training accuracy of D-SGD on these two datasets; the second kind is to demonstrate the effects due to the structure of the connected graph. We run our experiments on different types of connected graphs with the same dataset. In particular, we test six different connected graphs, as shown in Figure 1.

[2]For the strongly convex case, we avoid showing the result under general step size due to the complicated form.

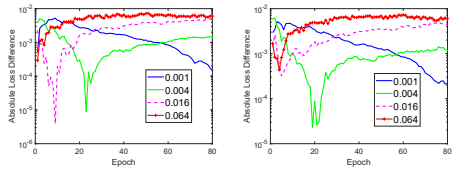


(a) Random (b) Star (c) Cycle (d) k-NNG (e) Bipartite (f) Complete

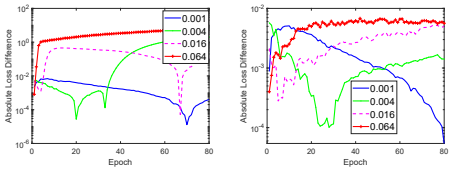
Figure 3: Absolute loss difference versus epochs for linear regression task on Body Fat dataset. With current learning rates, D-SGD diverges for the Cycle graph. With enough iterations, the smaller learning rate can achieves a smaller difference for all graph.

Convex Case

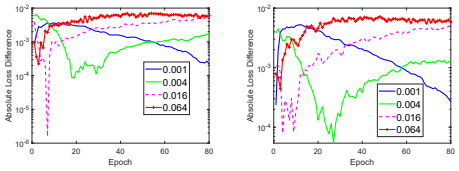
We consider the following optimization problem $\min_{\mathbf{x}\in\mathbb{R}^{14}}\Phi(\mathbf{x}) := \frac{1}{504}\sum_{i=1}^{252}\|\xi_i^\top\mathbf{x} - \mathbf{y}_i\|^2$, which arises from a simple regression problem. Here, we use the Body Fat dataset (Johnson 1996) which contains 252 samples. We run D-SGD on two subsets of the Body Fat dataset, and both of size 200. Let $\mathbf{x}^k$ and $\widehat{\mathbf{x}}^k$ be the outputs of the D-SGD on the two different subsets. We define the absolute loss difference as $|\Phi(\mathbf{x}^k) - \Phi(\widehat{\mathbf{x}}^k)|$. For the above six graphs, we record the absolute difference in the value of function Φ for a set of learning rate, namely, $\{0.001, 0.004, 0.016, 0.064\}$ in Figure 3. In the second test, we use the learning rate 0.001 and compare the absolute loss difference with different graphs in Figure 2 (a). Our results show that the smaller learning rate usually yields a smaller loss difference, and the complete graph can achieve the smallest bound. These observations are consistent with our theoretical results for the convex D-SGD.



(a) Random (b) Star

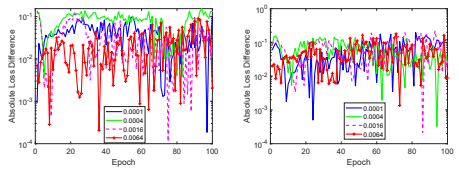


(c) Cycle (d) k-NNG

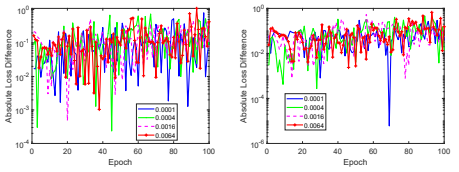


(e) Bipartite (f) Complete

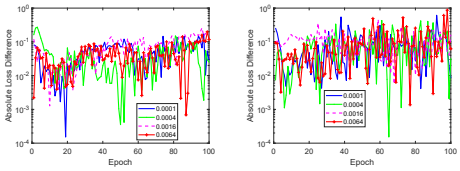
Figure 4: Absolute loss difference versus epochs for the ℓ_2 regularized logistic regression task on ijcnn1. The strongly convex tests are similar to general convex ones but more smooth. With strong convexity, D-SGD converges over the Cycle graph. In particular, when a smaller learning rate is used.



(a) Random (b) Star



(c) Cycle (d) k-NNG



(e) Bipartite (f) Complete

Figure 5: Absolute loss difference versus epochs for training nonconvex machine learning model, i.e., ResNet20. Unlike the convex cases, the absolute loss difference oscillates chaotically, which implies worse stability.

Strongly Convex Case

To verify our theory on the strongly convex case, we consider the regularized logistic regression model as follows

$$\min_{\mathbf{x}\in\mathbb{R}^{22}} \left\{ \frac{1}{9000} \sum_{i=1}^{9000} \left(\log(1+\exp(-b_i \mathbf{a}_i^\top \mathbf{x})) + \frac{\lambda}{2}\|\mathbf{x}\|^2 \right) \right\}.$$

We use the benchmark ijcnn1 dataset(Rennie and Rifkin 2001) and set $\lambda = 10^{-4}$. Two 8000-sample sub-datasets with 1000 different samples are used as the test set. We conduct experiments on the two datasets with the same set of learning rates that are used in the last subsection. The absolute loss difference under different learning rates is plotted in Figure 4, and the performance under different graphs is reported in Figure 2 (b). The results of D-SGD in the strongly convex case is similar to the convex case. Also, note that the absolute loss difference increases as the learning rates grow.

Nonconvex Case

We test ResNet-20 (He et al. 2016) for CIFAR10 classification (Krizhevsky 2009). We adopt two different 40000-sample subsets. The loss values are built on the test set. The absolute loss difference with the learning rate set $\{0.0001, 0.0004, 0.0016, 0.0064\}$ versus the epochs is presented in Figure 5, and the absolute loss difference with different graphs are shown in Figure 5 (c). 100 epochs are used in the nonconvex test. The results show that the nonconvex D-SGD is much more unstable than the convex ones, which matches our theoretical findings.

Conclusion

In this paper, we develop the stability and generalization error for the (projected) decentralized stochastic gradient descent (D-SGD) in strongly convex, convex, and nonconvex settings. In contrast to the previous works on the analysis of the projected decentralized gradient descent, our theories are built on much more relaxed assumptions. Our theoretical results show that the stability and generalization of D-SGD depend on the learning rate and the structure of the connected graph. Furthermore, we prove that decentralization deteriorates the stability of D-SGD. Our theoretical results are empirically supported by experiments on training different machine learning models in different decentralization settings. There are numerous avenues for future work: 1) deriving the improved stability and generalization bounds of D-SGD in the general convex and nonconvex cases, 2) proving the high probability bounds, 3) studying the stability and generalization bound of the moment variance of D-SGD.

Acknowledgments

The authors are indebted to anonymous reviewers for their useful suggestions. This work is sponsored in part by National Key R&D Program of China (2018YFB0204300), and the National Science Foundation of China (No. 61932001 and 61906200).

References

Agarwal, A.; and Duchi, J. C. 2011. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, 873–881.

Bottou, L.; and Bousquet, O. 2008. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, 161–168.

Bousquet, O.; and Elisseeff, A. 2002. Stability and generalization. *Journal of Machine Learning Research* 2(Mar): 499–526.

Boyd, S.; Diaconis, P.; and Xiao, L. 2004. Fastest mixing Markov chain on a graph. *SIAM review* 46(4): 667–689.

Boyd, S.; Ghosh, A.; Prabhakar, B.; and Shah, D. 2005. Gossip algorithms: Design, analysis and applications. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, 1653–1664. IEEE.

Charles, Z.; and Papailiopoulos, D. 2018. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learning*, 745–754.

Dekel, O.; Gilad-Bachrach, R.; Shamir, O.; and Xiao, L. 2012. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research* 13: 165–202.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(7).

Elisseeff, A.; Evgeniou, T.; and Pontil, M. 2005. Stability of randomized learning algorithms. *Journal of Machine Learning Research* 6(Jan): 55–79.

Hardt, M.; Recht, B.; and Singer, Y. 2016. Train faster, generalize better: Stability of stochastic gradient descent. *ICML 2016* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Johnson, R. W. 1996. Fitting Percentage of Body Fat to Simple Body Measurements. *Journal of Statistics Education* 4(1).

Karimi, H.; Nutini, J.; and Schmidt, M. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 795–811. Springer.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *ICLR 2014* .

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. *Master's Dissertation, University of Toronto* .

Kuzborskij, I.; and Lampert, C. 2018. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning*, 2815–2824.

Lan, G.; Lee, S.; and Zhou, Y. 2020. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming* 180(1): 237–284.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553): 436–444.

Lei, Y.; and Ying, Y. 2020. Fine-Grained analysis of stability and generalization for SGD. In *Proceedings of the 37th International Conference on Machine Learning, 2020*.

Li, J.; Luo, X.; and Qiao, M. 2019. On Generalization Error Bounds of Noisy Gradient Methods for Non-Convex Learning. *Proceedings of Machine Learning Research* 1: 37.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 5330–5340.

Lian, X.; Zhang, W.; Zhang, C.; and Liu, J. 2018. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, 3043–3052.

Lin, J.; Camoriano, R.; and Rosasco, L. 2016. Generalization properties and implicit regularization for multiple passes SGM. In *International Conference on Machine Learning*, 2340–2348.

Marshall, A. W.; Olkin, I.; and Arnold, B. C. 1979. *Inequalities: theory of majorization and its applications*, volume 143. Springer.

Montenegro, R. R.; and Tetali, P. 2006. *Mathematical aspects of mixing times in Markov chains*. Now Publishers Inc.

Mou, W.; Wang, L.; Zhai, X.; and Zheng, K. 2018. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, 605–638.

Nedic, A.; and Ozdaglar, A. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control* 54(1): 48–61.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization* 19(4): 1574–1609.

Olfati-Saber, R.; Fax, J. A.; and Murray, R. M. 2007. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE* 95(1): 215–233.

Rakhlin, A.; Shamir, O.; and Sridharan, K. 2012. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1571–1578.

Ram, S. S.; Nedić, A.; and Veeravalli, V. V. 2010a. Asynchronous gossip algorithm for stochastic optimization: Constant stepsize analysis. In *Recent Advances in Optimization and its Applications in Engineering*, 51–60. Springer.

Ram, S. S.; Nedić, A.; and Veeravalli, V. V. 2010b. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications* 147(3): 516–545.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 693–701.

Rennie, J. D.; and Rifkin, R. 2001. Improving multiclass text classification with the support vector machine. *MIT Technical Reports: AI Memo AIM-2001-026* .

Richards, D.; et al. 2020. Graph-dependent implicit regularisation for distributed stochastic subgradient descent. *Journal of Machine Learning Research* 21(2020).

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 400–407.

Shalev-Shwartz, S.; Shamir, O.; Srebro, N.; and Sridharan, K. 2010. Learnability, stability and uniform convergence. *Journal of Machine Learning Research* 11: 2635–2670.

Sirb, B.; and Ye, X. 2016. Consensus optimization with delayed and stochastic gradients on decentralized networks. In *2016 IEEE International Conference on Big Data (Big Data)*, 76–85. IEEE.

Srivastava, K.; and Nedic, A. 2011. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing* 5(4): 772–790.

Sun, T.; Chen, T.; Sun, Y.; Liao, Q.; and Li, D. 2019. Decentralized Markov Chain Gradient Descent. *preprint arXiv:1909.10238* .

Wang, B.; Nguyen, T. M.; Sun, T.; Bertozzi, A. L.; Baraniuk, R. G.; and Osher, S. J. 2020. Scheduled restart momentum for accelerated stochastic gradient descent. *preprint arXiv:2002.10583* .

Yuan, K.; Ling, Q.; and Yin, W. 2016. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization* 26(3): 1835–1854.