Efficient Decentralized Stochastic Gradient Descent Method for Nonconvex Finite-Sum Optimization Problems

Wenkang Zhan[1], **Gang Wu**[2], **Hongchang Gao**[1]

[1]Department of Computer and Information Sciences, Temple University, PA, USA
[2]Adobe Research, CA, USA
wenkang.zhan@temple.edu, gawu@adobe.com, hongchang.gao@temple.edu

Abstract

Decentralized stochastic gradient descent methods have attracted increasing interest in recent years. Numerous methods have been proposed for the nonconvex finite-sum optimization problem. However, existing methods have a large sample complexity, slowing down the empirical convergence speed. To address this issue, in this paper, we proposed a novel decentralized stochastic gradient descent method for the nonconvex finite-sum optimization problem, which enjoys a better sample and communication complexity than existing methods. To the best of our knowledge, our work is the first one achieving such favorable sample and communication complexities. Finally, we have conducted extensive experiments and the experimental results have confirmed the superior performance of our proposed method.

Introduction

With the emergence of large-scale distributed data, the decentralized training method has attracted increasing interest in recent years in the machine learning community. In this paper, we are interested in optimizing the following decentralized nonconvex finite-sum optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n} f_i^{(k)}(\mathbf{x}) \,. \tag{1}$$

Here, it is assumed that there are K workers in a decentralized training system. $\frac{1}{n}\sum_{i=1}^{n} f_i^{(k)}(\mathbf{x})$ is the loss function on the k-th worker where $\mathbf{x}\in\mathbb{R}^d$ denotes the model parameter and n is the number of samples on each worker. Essentially, optimizing Eq. (1) is to learn the model parameter $\mathbf{x}$ via the collaboration between K workers.

To optimize Eq. (1), a wide variety of decentralized training methods have been proposed under both stochastic and finite-sum settings. For instance, under the stochastic setting, (Lian et al. 2017) developed the decentralized stochastic gradient descent (DSGD) method and provided the convergence analysis for nonconvex problems. In particular, to achieve the ϵ-accuracy solution, the sample complexity of DSGD is $O(K/\epsilon^4)$ and the communication complexity is also $O(1/\epsilon^4)$. Here, the network topology only affects the high-order term of these complexities. (Yu, Jin, and Yang 2019) developed the decentralized stochastic gradient descent with momentum (DSGDM) method, which has the same theoretical sample and communication complexities as DSGD. Recently, (Xin, Khan, and Kar 2021; Zhang et al. 2021b) proposed a hybrid decentralized stochastic gradient descent (HSGD) method, which achieves better sample and communication complexities. However, these methods focus on the stochastic setting, failing to disclose how the finite-sum structure affects those complexities.

As for the finite-sum setting, based on the variance reduction method developed in (Fang et al. 2018), (Sun, Lu, and Hong 2020) proposed the decentralized gradient estimation and tracking (DGET) method, whose sample complexity is $O(Kn+Kn^{1/2}/((1-\lambda)^p\epsilon^2))$ and communication complexity is $O(1/((1-\lambda)^p\epsilon^2))$ where $1-\lambda$ is the spectral gap and $p>1$. Afterwards, GT-SARAH (Xin, Khan, and Kar 2022) refined the theoretical analysis and achieved improved sample and communication complexities (See Table 1). However, DGET/GT-SARAH needs to compute the full gradient periodically to achieve such sample and communication complexities, which is prohibitive for large-scale data. As such, their sample complexity is suboptimal. Specifically, it is inferior to that of the existing centralized method (Li and Richtárik 2021).

To address the aforementioned problems, we developed a novel efficient decentralized stochastic gradient descent method for the nonconvex finite-sum optimization problem. Particularly, to improve the sample complexity, our method employs a variance reduction technique to estimate the gradient on each worker and uses the gradient tracking strategy to communicate the gradient across different workers. Our theoretical analysis demonstrates that our method enjoys the $O(K^{1/2}n^{1/2}/((1-\lambda)\epsilon^2))$ sample complexity and $O(1/((1-\lambda)\epsilon^2))$ communication complexity. It is worth noting that our communication complexity is much better than DGET and GT-SARAH (See Table 1). To the best of our knowledge, this is the first work achieving such favorable sample and communication complexities for the nonconvex finite-sum optimization problem. However, it is challenging to obtain this theoretical result. Specifically, the variance-reduced gradient makes it difficult to bound the consensus error under the gradient-tracking communication setting. In this paper, we developed novel techniques to ad-

Methods		Sample	Communication	Requirement	FG
Stochastic	DSGD (Lian et al. 2017)	$O(\frac{K}{\epsilon^4})$	$O(\frac{1}{\epsilon^4})$	-	-
	DSGDM (Yu, Jin, and Yang 2019)	$O(\frac{K}{\epsilon^4})$	$O(\frac{1}{\epsilon^4})$	-	-
	HSGD (Xin, Khan, and Kar 2021)	$O\left(\frac{1}{\epsilon^3}\right)$	$O\left(\frac{1}{K\epsilon^3}\right)$	$\epsilon \lesssim \min\left\{\frac{(1-\lambda)^3}{\lambda^4 K}, \frac{(1-\lambda)^{1.5}}{\lambda K}\right\}$	-
	DGET (Sun, Lu, and Hong 2020)	$O\left(\frac{K}{(1-\lambda)^p\epsilon^3}\right)$	$O\left(\frac{1}{(1-\lambda)^p\epsilon^2}\right)$	-	-
Finite-sum	DGET (Sun, Lu, and Hong 2020)	$O\left(Kn + \frac{Kn^{1/2}}{(1-\lambda)^p\epsilon^2}\right)$	$O\left(\frac{1}{(1-\lambda)^p\epsilon^2}\right)$	-	✓
	GT-SARAH (Xin, Khan, and Kar 2022)	$O\left(Kn + \frac{K^{1/2}n^{1/2}}{\epsilon^2}\right)$*	$O\left(\frac{1}{(1-\lambda)^3\epsilon^2}\right)$	$n = O\left(\frac{K}{(1-\lambda)^6}\right)$	✓
	EDSGD (Ours)	$O\left(\frac{K^{1/2}n^{1/2}}{(1-\lambda)\epsilon^2}\right)$	$O\left(\frac{1}{(1-\lambda)\epsilon^2}\right)$	-	✗

Table 1: The sample and communication complexity of different methods to achieve the ϵ-accuracy solution, i.e., $\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\bar{\mathbf{x}}_t)\|^2 \leq \epsilon^2$. Here, $1-\lambda \in (0,1)$ denotes the spectral gap of the network topology. The last column denotes whether it is necessary to compute the full gradient. Note that DGET can also be used in the stochastic setting, and $p \in \mathbb{R}^+$ is not explicitly given in (Sun, Lu, and Hong 2020). * The claimed spectral-gap-independent sample complexity of GT-SARAH in (Xin, Khan, and Kar 2022) is not true because n depends on the spectral gap.

dress this challenging problem and successfully established the convergence rate of our method. Finally, we applied our method to train the decentralized nonconvex logistic regression model. The extensive experimental results have demonstrated the superior performance of our proposed method. In summary, our work has made the following contributions.

- We proposed a novel decentralized stochastic gradient descent method for the nonconvex finite-sum optimization problem, which can achieve the $O(K^{1/2}n^{1/2}/((1-\lambda)\epsilon^2))$ sample complexity and $O(1/((1-\lambda)\epsilon^2))$ communication complexity.
- We developed novel techniques for bounding the consensus error across different workers to establish the convergence rate of our method.
- We conducted extensive experiments to verify the convergence performance of our method and the experimental result can support our theoretical result.

Related Works

Decentralized optimization methods have been actively studied in recent years due to their efficiency and robustness in communication. In particular, different from the parameter-server schema where there might be a communication bottleneck in the central server, there is no central server in a decentralized training system, and the workers conduct peer-to-peer communication. In this regime, numerous decentralized optimization methods (Lian et al. 2017; Yu, Jin, and Yang 2019; Koloskova et al. 2020; Lu et al. 2019; Wang et al. 2019; Shi et al. 2015; Tang et al. 2018; Koloskova, Stich, and Jaggi 2019; Gao and Huang 2021; Gao, Xu, and Vucetic 2021; Gao and Huang 2020) have been proposed. In terms of the communication strategy, those methods can be categorized into two classes: the gossip-based method and gradient-tracking-based method. The latter one uses the gradient tracking technique to track the global gradient so that it is more stable than the gossip-based method. Hence, in this paper, we will focus on the gradient-tracking-based method.

To optimize the large-scale machine learning models efficiently, a wide variety of works have been proposed to improve the convergence performance of decentralized optimization algorithms. For instance, (Lian et al. 2017) developed DSGD and disclosed that the dominant term of its convergence rate is consistent with the centralized counterpart. (Koloskova et al. 2020) studied the convergence rate of decentralized SGD with changing topology and local updates for both convex and nonconvex problems. (Pu and Nedić 2021) investigated the convergence rate of the gradient-tracking-based decentralized SGD for convex problems, while (Lu et al. 2019) established its convergence rate for the nonconvex problems. Additionally, (Yu, Jin, and Yang 2019; Lin et al. 2021; Yuan et al. 2021) applied the momentum technique to decentralized SGD for accelerating the convergence speed. However, the sample complexity and communication complexity of these methods are suboptimal due to the large variance of stochastic gradients.

More recently, to improve the sample complexity and communication complexity of decentralized SGD, a line of research (Li et al. 2020; Qureshi et al. 2021; Sun, Lu, and Hong 2020; Xin, Khan, and Kar 2021; Zhang et al. 2021b,a; Xin, Khan, and Kar 2020) focuses on reducing the variance of stochastic gradients (Defazio, Bach, and Lacoste-Julien 2014; Fang et al. 2018; Nguyen et al. 2017; Zhou, Xu, and Gu 2018). For instance, (Xin, Khan, and Kar 2020) applied SAGA (Defazio, Bach, and Lacoste-Julien 2014) and SVRG (Johnson and Zhang 2013) to the decentralized SGD method and established the convergence rate for convex problems. (Sun, Lu, and Hong 2020) developed the decentralized gradient estimation and tracking (DGET) method by incorporating the SPIDER (Fang et al. 2018) gradient estimator into the gradient tracking framework, resulting better sample and communication complexities than traditional decentralized SGD method for nonconvex problems. Additionally, (Xin, Khan, and Kar 2021; Zhang et al. 2021b) developed another decentralized variance-reduced SGD (HSGD) method based on the STORM gradient estimator (Cutkosky and Orabona 2019). However, it has a worse communication complexity

than DGET. Thus, it is necessary to develop more efficient decentralized methods to improve the sample and communication complexity.

Efficient Decentralized Stochastic Gradient Descent Method

In Algorithm 1, we developed an efficient decentralized stochastic gradient descent (EDSGD) method to have better sample and communication complexities. In detail, in the t-th iteration, each worker k randomly selects a subset $\mathcal{S}_t^{(k)}$ from the local dataset and then computes the following variance-reduced gradient:

$$\begin{aligned}\mathbf{v}_t^{(k)} &= \frac{1}{s_t}\sum_{i\in\mathcal{S}_t^{(k)}}\Big(\nabla f_i^{(k)}(\mathbf{x}_t^{(k)}) - \nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)}\Big) \\ &+ \rho_t\Big(\frac{1}{s_t}\sum_{i\in\mathcal{S}_t^{(k)}}(\nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)}) - \mathbf{g}_{i,t-1}^{(k)}) + \frac{1}{n}\sum_{j=1}^{n}\mathbf{g}_{j,t-1}^{(k)}\Big) \\ &+ (1-\rho_t)\mathbf{v}_{t-1}^{(k)}\,,\end{aligned} \tag{2}$$

where $|\mathcal{S}_t^{(k)}| = s_t$, $\rho_t \in [0,1]$ is a hyperparameter, $\mathbf{g}_{i,t}^{(k)}$ is the historical gradient of the i-th sample on the k-th worker, which is updated by

$$\mathbf{g}_{i,t}^{(k)} = \begin{cases}\nabla f_i^{(k)}(\mathbf{x}_t^{(k)}), & \text{for } i \in \mathcal{S}_t^{(k)} \\ \mathbf{g}_{i,t-1}^{(k)}, & \text{otherwise}\,.\end{cases}$$

With this variance reduced gradient $\mathbf{v}_t^{(k)}$, our algorithm utilizes the gradient tracking strategy to update the model parameter as follows:

$$\begin{aligned}\mathbf{u}_t^{(k)} &= \sum_{j\in\mathcal{N}_k} w_{kj}\mathbf{u}_{t-1}^{(j)} + \mathbf{v}_t^{(k)} - \mathbf{v}_{t-1}^{(k)}\,, \\ \mathbf{x}_{t+1}^{(k)} &= \sum_{j\in\mathcal{N}_k} w_{kj}\mathbf{x}_t^{(j)} - \eta\mathbf{u}_t^{(k)}\,,\end{aligned} \tag{3}$$

where $\mathcal{N}_k$ denotes the neighbors of the k-th worker, $w_{kj} > 0$ denotes the edge weight between the k-th worker and the j-th worker, $\mathbf{u}_t^{(k)}$ is used to track the global gradient, and $\eta > 0$ is the learning rate.

In Algorithm 1, the gradient estimator $\mathbf{v}_t^{(k)}$ has a smaller variance so that the convergence speed can be accelerated. Additionally, compared with DGET (Sun, Lu, and Hong 2020), Algorithm 1 does not need to compute the full gradient periodically. Thus, our method is more efficient than those methods. In fact, the gradient estimator in Eq. (1) was first developed in (Li and Richtárik 2021). It can be viewed as the combination of SPIDER (Fang et al. 2018; Nguyen et al. 2017) and SAGA (Defazio, Bach, and Lacoste-Julien 2014). However, (Li and Richtárik 2021) only studied its convergence rate for the standard nonconvex finite-sum problem, rather than the decentralized optimization problem. Hence, it is still unclear whether $\mathbf{v}_t^{(k)}$ can be applied to the decentralized setting. Especially when the gradient tracking communication strategy is used, it is unclear whether $\mathbf{v}_t^{(k)}$ can lead to better sample and communication complexities. In particular, the variance reduced gradient $\mathbf{v}_t^{(k)}$ and the tracked gradient $\mathbf{u}_t^{(k)}$ make it extraordinarily challenging to bound the consensus error $\|\mathbf{x}_t^{(k)} - \bar{\mathbf{x}}_t\|^2$ (where $\bar{\mathbf{x}}_t = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_t^{(k)}$) for establishing the convergence rate of our Algorithm 1. In this paper, we addressed this challenging problem and theoretically demonstrated that Algorithm 1 can achieve better sample and communication complexities than DGET.

Algorithm 1: Efficient Decentralized Stochastic Gradient Descent Method (EDSGD)

Input: $\mathbf{x}_{-1}^{(k)} = \mathbf{x}_0^{(k)} = \mathbf{x}_0$, $\mathbf{v}_{-1}^{(k)} = 0, \mathbf{g}_{i,-1}^{(k)} = 0$, $\rho_t \in [0,1]$, $\eta > 0$, $s_t > 0$

1: **for** $t = 0, \cdots, T-1$ **do**

2: Randomly select a subset of samples $\mathcal{S}_t^{(k)}$ with $|\mathcal{S}_t^{(k)}| = s_t$:
$\mathbf{v}_t^{(k)} = \frac{1}{s_t}\sum_{i\in\mathcal{S}_t^{(k)}}\Big(\nabla f_i^{(k)}(\mathbf{x}_t^{(k)}) - \nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)})\Big) + (1-\rho_t)\mathbf{v}_{t-1}^{(k)} + \rho_t\Big(\frac{1}{s_t}\sum_{i\in\mathcal{S}_t^{(k)}}(\nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)}) - \mathbf{g}_{i,t-1}^{(k)}) + \frac{1}{n}\sum_{j=1}^{n}\mathbf{g}_{j,t-1}^{(k)}\Big)$

3: Update $\mathbf{x}$:
$\mathbf{u}_t^{(k)} = \sum_{j\in\mathcal{N}_k} w_{kj}\mathbf{u}_{t-1}^{(j)} + \mathbf{v}_t^{(k)} - \mathbf{v}_{t-1}^{(k)}$
$\mathbf{x}_{t+1}^{(k)} = \sum_{j\in\mathcal{N}_k} w_{kj}\mathbf{x}_t^{(j)} - \eta\mathbf{u}_t^{(k)}$

4: Update $\mathbf{g}_{i,t}^{(k)}$:
$\mathbf{g}_{i,t}^{(k)} = \begin{cases}\nabla f_i^{(k)}(\mathbf{x}_t^{(k)}), & \text{for } i \in \mathcal{S}_t^{(k)} \\ \mathbf{g}_{i,t-1}^{(k)}, & \text{otherwise}\end{cases}$

5: **end for**

To establish the convergence rate of our method, we introduce the following commonly used assumptions.

Assumption 1. *(L-smooth) For $\forall \mathbf{x}, \mathbf{y}$, there exists a constant $L > 0$, such that*

$$\|\nabla f_i^{(k)}(\mathbf{x}) - \nabla f_i^{(k)}(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|\,, \forall i, k. \tag{4}$$

Assumption 2. *(Network topology) The adjacency matrix $W = [w_{ij}] \in \mathbb{R}_+^{K\times K}$ is symmetric and doubly stochastic. Here, $w_{ij} > 0$ indicates that the i-th worker and the j-th worker are connected. In addition, the eigenvalues $\{\lambda_i\}_{i=1}^n$ of W are assumed to satisfy $|\lambda_n| \le \cdots \le |\lambda_2| < |\lambda_1| = 1$.*

Based on Assumption 2, we can represent the spectral gap of the network topology as $1-\lambda$ where $\lambda \triangleq |\lambda_2| < 1$. With these two assumptions, we established the convergence rate of Algorithm 1 in Theorem 1.

Theorem 1. *Given Assumptions 1-2, for Algorithm 1, by setting $\eta \le \min\Big\{\frac{1}{2L}, \frac{s_1}{\sqrt{504nL}}, \Big(-\frac{2}{1-\lambda^2} + \sqrt{\frac{4}{(1-\lambda^2)^2} + \frac{42n}{s_1^2}\frac{(1-\lambda^2)^2}{\rho_1}}\Big)\Big/\frac{252nL}{s_1^2}, \frac{(1-\lambda^2)}{2L}\Big/\Big(1+\frac{504n}{s_1^2}\Big)\Big\}$, $s_0 \le n$, $s_t \equiv s_1$ for $t \ge 1$, $\rho_0 = 1$, $\rho_t \equiv \rho_1 = \frac{s_1}{2n}$ for $t \ge 1$,*

we can obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0)-F(\mathbf{x}_*))}{\eta T} + \frac{14(s_0-s_0^2/n)}{s_0 s_1}\frac{1}{TK}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 + \frac{24(n-s_0)}{s_0 s_1}\frac{1}{TK}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 , \quad (5)$$

where $\mathbf{x}_*$ denotes the optimal solution.

Remark 1. *(Communication complexity) From Theorem 1, it can be observed that $\eta = O(1-\lambda)$. Thus, to achieve the ϵ-accuracy solution, the convergence rate (i.e., the communication complexity) of Algorithm 1 is $O\left(\frac{1}{(1-\lambda)\epsilon^2}\right)$. It is worth noting that this communication complexity enjoys better dependence on the spectral gap $1-\lambda$ than existing methods. Specifically, DGET (Sun, Lu, and Hong 2020) has the communication complexity $O\left(\frac{1}{(1-\lambda)^p\epsilon^2}\right)$ where $p>1$, and the best communication complexity of GT-SARAH (Xin, Khan, and Kar 2022) is $O\left(\frac{1}{(1-\lambda)^2\epsilon^2}\right)$ in the big-data or large-network regime. Thus, our communication complexity is better than those methods according to the spectral gap.*

Remark 2. *(Sample complexity) By setting $s_0 = s_1 = \sqrt{n/K}$ where $K < n$, then $\eta \leq \min\Big\{\frac{1}{2L}, \frac{1}{\sqrt{504K}L}, \frac{1}{252KL}\Big(- \frac{2}{1-\lambda^2} + \sqrt{\frac{4}{(1-\lambda^2)^2}+84n^{1/2}K^{3/2}(1-\lambda^2)^2}\Big), \frac{(1-\lambda^2)}{2(1+504K)L}\Big\}$, we can get*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0)-F(\mathbf{x}_*))}{\eta T} + \frac{14+24K}{TK}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 . \quad (6)$$

Hence, to achieve the ϵ-accuracy solution, the sample complexity of Algorithm 1 is $K\times s_1\times T = O\left(\frac{K^{1/2}n^{1/2}}{(1-\lambda)\epsilon^2}\right)$. Obviously, it is better than the sample complexity $O\left(Kn+\frac{Kn^{1/2}}{((1-\lambda)^p\epsilon^2)}\right)$ of DGET. This improvement is because our method does not need to compute the full gradient periodically as DGET. Note that GT-SARAH claims that it can achieve a topology-independent sample complexity $O\left(Kn+\frac{K^{1/2}n^{1/2}}{\epsilon^2}\right)$ when the number of samples n is as large as $O\left(\frac{K}{(1-\lambda)^6}\right)$ and the communication complexity is increased to $O\left(\frac{1}{(1-\lambda)^3\epsilon^2}\right)$. However, it is not true because n depends on the spectral gap heavily. Thus, its sample complexity is worse than ours according to the spectral gap.

In summary, our method can achieve better sample and communication complexities than DGET and GT-SARAH. Hence, our method is more efficient than those existing decentralized methods.

Theoretical Analysis

In this section, we present the proof sketch of Theorem 1. The detailed proof can be found in Supplementary Material.

Throughout our theoretical analysis, we use $\bar{\mathbf{m}}_t = \frac{1}{K}\sum_{k=1}^{K}\mathbf{m}_t^{(k)}$ to represent the averaged variable across all workers, where $\mathbf{m}_t^{(k)}$ includes $\mathbf{x}_t^{(k)}$, $\mathbf{v}_t^{(k)}$, $\mathbf{u}_t^{(k)}$. In addition, we use $M_t = [\mathbf{m}_t^{(1)}, \mathbf{m}_t^{(2)}, \cdots, \mathbf{m}_t^{(K)}]$ to denote the variables in all workers. Furthermore, we denote $\bar{M}_t = [\bar{\mathbf{m}}_t, \bar{\mathbf{m}}_t, \cdots, \bar{\mathbf{m}}_t]$, which includes K copies of $\bar{\mathbf{m}}_t$. To establish the convergence rate of Algorithm 1, an important step is to bound the consensus errors $\sum_{k=1}^{K}\|\mathbf{x}_t^{(k)}-\bar{\mathbf{x}}_t\|^2$ and $\sum_{k=1}^{K}\|\mathbf{u}_t^{(k)}-\bar{\mathbf{u}}_t\|^2$. However, it is challenging due to the interaction between the variance-reduced gradient $\mathbf{v}_t^{(k)}$ and the tracked gradient $\mathbf{u}_t^{(k)}$. To address this challenging problem, we first constructed the recursive upper bound for these two consensus errors and then developed a novel Lyapunov function. With these novel techniques, we can establish the convergence rate of Algorithm 1. To this end, we first introduce two auxiliary lemmas.

Lemma 1. *With Assumption 2, we can get*

$$\sum_{k=1}^{K}\|\mathbf{x}_{t+1}^{(k)}-\mathbf{x}_t^{(k)}\|^2 \leq 12\sum_{k=1}^{K}\|\mathbf{x}_t^{(k)}-\bar{\mathbf{x}}_t\|^2 + 3\eta^2\sum_{k=1}^{K}\|\mathbf{u}_t^{(k)}-\bar{\mathbf{u}}_t\|^2 + 3\eta^2 K\|\bar{\mathbf{v}}_t\|^2 . \quad (7)$$

Lemma 2. *With Assumption 1, for $t>0$, we set the batch size as $s_t = s_1$ and $\rho_t = \rho_1$. we can get*

$$\begin{aligned}&\mathbb{E}[\|\mathbf{v}_{t+1}^{(k)}-\mathbf{v}_t^{(k)}\|^2]\\ &\leq \frac{3L^2}{s_1}\mathbb{E}[\|\mathbf{x}_{t+1}^{(k)}-\mathbf{x}_t^{(k)}\|^2] + 3\rho_1^2\mathbb{E}[\|\mathbf{v}_t^{(k)}-\nabla f^{(k)}(\mathbf{x}_t^{(k)})\|^2]\\ &\quad + \frac{3\rho_1^2}{s_1}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[\|\nabla f_j^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{g}_{j,t}^{(k)}\|^2] .\end{aligned} \quad (8)$$

Based on these two lemmas, we can establish the upper bound for the consensus error in Lemma 3 and Lemma 4, respectively.

Lemma 3. *With Assumption 2, we can get*

$$\begin{aligned}\sum_{k=1}^{K}\|\mathbf{u}_{t+1}^{(k)}-\bar{\mathbf{u}}_{t+1}\|^2 &\leq \frac{1+\lambda^2}{2}\sum_{k=1}^{K}\|\mathbf{u}_t^{(k)}-\bar{\mathbf{u}}_t\|^2\\ &+ \frac{6L^2}{(1-\lambda^2)s_1}\sum_{k=1}^{K}\|\mathbf{x}_{t+1}^{(k)}-\mathbf{x}_t^{(k)}\|^2\\ &+ \frac{6\rho_1^2}{1-\lambda^2}\sum_{k=1}^{K}\|\mathbf{v}_t^{(k)}-\nabla f^{(k)}(\mathbf{x}_t^{(k)})\|^2\\ &+ \frac{6\rho_1^2}{(1-\lambda^2)s_1}\sum_{k=1}^{K}\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{g}_{j,t}^{(k)}\|^2 .\end{aligned} \quad (9)$$

Lemma 4. *With Assumption 2, we can get*

$$\sum_{k=1}^{K}\|\mathbf{x}_{t+1}^{(k)}-\bar{\mathbf{x}}_{t+1}\|^2 \leq \frac{1+\lambda^2}{2}\sum_{k=1}^{K}\|\mathbf{x}_t^{(k)}-\bar{\mathbf{x}}_t\|^2 + \frac{2\eta^2}{1-\lambda^2}\sum_{k=1}^{K}\|\mathbf{u}_t^{(k)}-\bar{\mathbf{u}}_t\|^2 . \tag{10}$$

Furthermore, we introduce the following descent lemma for the objective function value.

Lemma 5. *With Assumption 1, we can get*

$$F(\bar{\mathbf{x}}_{t+1}) \leq F(\bar{\mathbf{x}}_t) - \frac{\eta}{2}\|\nabla F(\bar{\mathbf{x}}_t)\|^2 + \Big(\frac{\eta^2 L}{2}-\frac{\eta}{2}\Big)\|\bar{\mathbf{v}}_t\|^2 + \frac{\eta L^2}{K}\sum_{k=1}^{K}\|\bar{\mathbf{x}}_t-\mathbf{x}_t^{(k)}\|^2 + \frac{\eta}{K}\sum_{k=1}^{K}\|\nabla f^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{v}_t^{(k)}\|^2 . \tag{11}$$

The proof of the aforementioned lemmas can be found in Supplementary Material. Moreover, we include two additional lemmas to bound the gradient variance. Their proof can be found in (Li and Richtárik 2021).

Lemma 6. *(Li and Richtárik 2021) For $t>0$, we set the batch size as $s_t=s_1$ and $\rho_t=\rho_1$. Then, we can get*

$$\begin{aligned}\mathbb{E}[\|\mathbf{v}_t^{(k)}-\nabla f^{(k)}(\mathbf{x}_t^{(k)})\|^2] &\leq \frac{2L^2}{s_1}\mathbb{E}[\|\mathbf{x}_t^{(k)}-\mathbf{x}_{t-1}^{(k)}\|^2] \\ &+ \frac{2\rho_1^2}{s_1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\|\nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)})-\mathbf{g}_{i,t-1}^{(k)}\|^2] \\ &+ (1-\rho_1)^2\mathbb{E}[\|\mathbf{v}_{t-1}^{(k)}-\nabla f^{(k)}(\mathbf{x}_{t-1}^{(k)})\|^2] .\end{aligned} \tag{12}$$

For $t=0$, we set the batch size as s_0 and $\rho_0=1$, then we can get

$$\begin{aligned}&\mathbb{E}[\|\mathbf{v}_0^{(k)}-\nabla f^{(k)}(\mathbf{x}_0)\|^2] \\ &= \frac{n-s_0}{(n-1)s_0}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 .\end{aligned} \tag{13}$$

Lemma 7. *(Li and Richtárik 2021) For $t>0$, we set the batch size as $s_t=s_1$ and $\alpha_1>0$. Then, we can get*

$$\begin{aligned}&\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{g}_{i,t}^{(k)}\|^2] \\ &\leq 2L^2(1-\frac{s_1}{n})(1+\frac{1}{\alpha_1})\mathbb{E}[\|\mathbf{x}_t^{(k)}-\mathbf{x}_{t-1}^{(k)}\|^2] \\ &\quad + (1-\frac{s_1}{n})(1+\alpha_1)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\|\nabla f_i^{(k)}(\mathbf{x}_{t-1}^{(k)})-\mathbf{g}_{i,t-1}^{(k)}\|^2] .\end{aligned} \tag{14}$$

When $t=0$, we can get

$$\begin{aligned}&\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)-\mathbf{g}_{i,0}^{(k)}\|^2] \\ &= \frac{n-b_0}{n^2}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 .\end{aligned} \tag{15}$$

To establish the convergence rate of Algorithm 1, we further developed a novel Lyapunov function

$$\begin{aligned}H_t &= \mathbb{E}[F(\bar{\mathbf{x}}_t)] + \frac{C_1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{v}_t^{(k)}\|^2] \\ &+ \frac{C_2}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[\|\nabla f_j(\mathbf{x}_t^{(k)})-\mathbf{g}_{j,t}^{(k)}\|^2] \\ &+ \frac{C_3}{K}\sum_{k=1}^{K}\mathbb{E}[\|\bar{\mathbf{x}}_t-\mathbf{x}_t^{(k)}\|^2] + \frac{C_4}{K}\sum_{k=1}^{K}\mathbb{E}[\|\bar{\mathbf{u}}_t-\mathbf{u}_t^{(k)}\|^2] ,\end{aligned} \tag{16}$$

where $C_1=\frac{3\eta}{\rho_1}$, $C_2=\frac{14n\eta\rho_1}{s_1^2}$, $C_3=L$, $C_4=\frac{(1-\lambda^2)\eta}{6\rho_1}$. For all the items in H_t, we have established their upper bound in the aforementioned lemmas. Then, based on this novel Lyapunov function, we can bound the gradient norm in each iteration. Consequently, the convergence rate of Algorithm 1 can be established. In the following, we present the details to prove Theorem 1.

Proof. Based on the aforementioned lemmas, we can get

$$\begin{aligned}&H_{t+1}-H_t \\ &\leq \mathbb{E}[F(\bar{\mathbf{x}}_t)] - \frac{\eta}{2}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_t)\|^2] \\ &\quad + \frac{A_1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\mathbf{x}_t^{(k)}-\bar{\mathbf{x}}_t\|^2] + \frac{A_2}{K}\sum_{k=1}^{K}\mathbb{E}[\|\mathbf{u}_t^{(k)}-\bar{\mathbf{u}}_t\|^2] \\ &\quad + \frac{A_4}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[\|\nabla f_j^{(k)}(\mathbf{x}_t^{(k)})-\mathbf{g}_{j,t}^{(k)}\|^2] \\ &\quad + \frac{A_5}{K}\sum_{k=1}^{K}\mathbb{E}[\|\mathbf{v}_t^{(k)}-\nabla f^{(k)}(\mathbf{x}_t^{(k)})\|^2] + A_3\mathbb{E}[\|\bar{\mathbf{v}}_t\|^2] ,\end{aligned} \tag{17}$$

where

$$\begin{aligned}A_1 &= 12\Big(2L^2(1-\frac{s_1}{n})(1+\frac{1}{\alpha_1})C_2 + \frac{2L^2}{s_1}C_1 \\ &\quad + \frac{6L^2}{(1-\lambda^2)s_1}C_4\Big) - \frac{1-\lambda^2}{2}C_3 + \eta L^2 , \\ A_2 &= 3\eta^2\Big(2L^2(1-\frac{s_1}{n})(1+\frac{1}{\alpha_1})C_2 + \frac{2L^2}{s_1}C_1 \\ &\quad + \frac{6L^2}{(1-\lambda^2)s_1}C_4\Big) - \frac{1-\lambda^2}{2}C_4 + \frac{2\eta^2}{1-\lambda^2}C_3 , \\ A_3 &= 3\eta^2\Big(2L^2(1-\frac{s_1}{n})(1+\frac{1}{\alpha_1})C_2 + \frac{2L^2}{s_1}C_1 \\ &\quad + \frac{6L^2}{(1-\lambda^2)s_1}C_4\Big) + \Big(\frac{\eta^2 L}{2}-\frac{\eta}{2}\Big) , \\ A_4 &= \frac{2\rho_1^2}{s_1}C_1 + \Big((1-\frac{s_1}{n})(1+\alpha_1)-1\Big)C_2 \\ &\quad + \frac{6\rho_1^2}{(1-\lambda^2)s_1}C_4 , \\ A_5 &= (1-\rho_1)^2C_1 - C_1 + \eta + \frac{6\rho_1^2}{1-\lambda^2}C_4 .\end{aligned} \tag{18}$$

Dataset	Instance	Dimensionality
a9a (LIBSVM)	32,561	123
w8a (LIBSVM)	49,749	300
ijcnn1 (LIBSVM)	49,990	22
cod-rna (LIBSVM)	59,535	8
covtype (LIBSVM)	581012	54
MiniBooNE (OpenML)	130,064	50

Table 2: Description of Benchmark Datasets

By setting $\rho_1 = \frac{s_1}{2n}$ and $\eta \leq \min\Big\{\frac{1}{2L}, \frac{s_1}{\sqrt{504n}L}, \Big(-\frac{2}{1-\lambda^2} + \sqrt{\frac{4}{(1-\lambda^2)^2} + \frac{42n}{s_1^2}\frac{(1-\lambda^2)^2}{\rho_1}}\Big)\Big/\frac{252nL}{s_1^2}, \frac{(1-\lambda^2)}{2L}\Big/\Big(1+\frac{504n}{s_1^2}\Big)\Big\}$, we can get $A_i \leq 0$ for $i = 1, 2, 3, 4, 5$. Hence, by summing t from 1 to $T-1$, we can get

$$
\begin{aligned}
&\frac{\eta}{2}\sum_{t=1}^{T-1}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_t)\|^2] \leq H_1 - H_T \\
&\leq \mathbb{E}[F(\bar{\mathbf{x}}_1)] - F(\mathbf{x}_*) + \frac{C_1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f^{(k)}(\mathbf{x}_1^{(k)}) - \mathbf{v}_1^{(k)}\|^2] \\
&\quad + \frac{C_2}{K}\sum_{k=1}^{K}\mathbb{E}[\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(\mathbf{x}_1^{(k)}) - \mathbf{g}_{j,1}^{(k)}\|^2] \\
&\quad + \frac{C_3}{K}\sum_{k=1}^{K}\mathbb{E}[\|\bar{\mathbf{x}}_1 - \mathbf{x}_1^{(k)}\|^2] + \frac{C_4}{K}\sum_{k=1}^{K}\mathbb{E}[\|\bar{\mathbf{u}}_1 - \mathbf{u}_1^{(k)}\|^2] \\
&\leq F(\bar{\mathbf{x}}_0) - F(\mathbf{x}_*) - \frac{\eta}{2}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_0)\|^2] \\
&\quad + \frac{14n\eta\rho_1}{s_1^2}(1-\frac{s_0}{n})\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 \\
&\quad + \frac{3\eta}{\rho_1}\frac{n-s_0}{(n-1)s_0}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 \\
&\leq F(\mathbf{x}_0) - F(\mathbf{x}_*) - \frac{\eta}{2}\mathbb{E}[\|\nabla F(\bar{\mathbf{x}}_0)\|^2] \\
&\quad + \frac{7\eta(s_0 - s_0^2/n)}{s_0 s_1}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2 \\
&\quad + \frac{12\eta(n-s_0)}{s_0 s_1}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i^{(k)}(\mathbf{x}_0)\|^2
\end{aligned}
\tag{19}
$$

where the second step follows from the definition of H_1 and $H_T \geq F(\mathbf{x}_*)$, the third step follows from Lemmas 3, 4, 6, 7, the last step follows from $\rho_1 = \frac{s_1}{2n}$. By reformulating this inequality and dividing $\frac{\eta T}{2}$ on both sides, we complete the proof. More details can be found in Supplementary Material. □

Experiments

To verify the performance of our method, we use Algorithm 1 to optimize the decentralized logistic regression problem which is defined as:

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^d} -\frac{1}{K}\sum_{k=1}^{K}\Bigg\{\frac{1}{n}\sum_{i=1}^{n}\Big(b_i^{(k)}\log(\frac{1}{1+e^{-\mathbf{x}^T\mathbf{a}_i^{(k)}}}) \\
+ (1-b_i^{(k)})\log(\frac{e^{-\mathbf{x}^T\mathbf{a}_i^{(k)}}}{1+e^{-\mathbf{x}^T\mathbf{a}_i^{(k)}}})\Big)\Bigg\} + \gamma\sum_{j=1}^{d}\frac{\mathbf{x}_j^2}{1+\mathbf{x}_j^2},
\end{aligned}
\tag{20}
$$

Here, $(\mathbf{a}_i^{(k)}, b_i^{(k)})$ represents the i-th sample on the k-th worker where $\mathbf{a}_i^{(k)}$ is the feature vector and $b_i^{(k)}$ is its label. Throughout our experiments, the regularization coefficient γ is set to 0.001.

In our experiments, we use six classification datasets, which are available at LIBSVM[1] and OpenML[2]. The statistic information of these datasets is summarized in Table **??**. In our experiment, ten workers are used to collaboratively train the logistic regression model. To simulate the communication graph, we use the Erdós-Rényi random graph to generate the connection between different workers, where the edge probability is set to 0.4. Then, the samples are randomly distributed to ten workers and each worker uses its own dataset to compute the stochastic gradient for updating model parameters.

The baseline methods used include DSGD (Lian et al. 2017), DSGDM (Yu, Jin, and Yang 2019), HSGD (Xin, Khan, and Kar 2021; Zhang et al. 2021b), DGET (Sun, Lu, and Hong 2020), and GT-SARAH (Xin, Khan, and Kar 2022). According to the theoretical results of those baseline methods, we set the batch size of the first three methods to 256 and DGET to $\sqrt{n}$. As for GT-SARAH and our method, we set it to $\sqrt{n/K}$. Similar to (Sun, Lu, and Hong 2020), we set the learning rate to 0.001 for all methods.

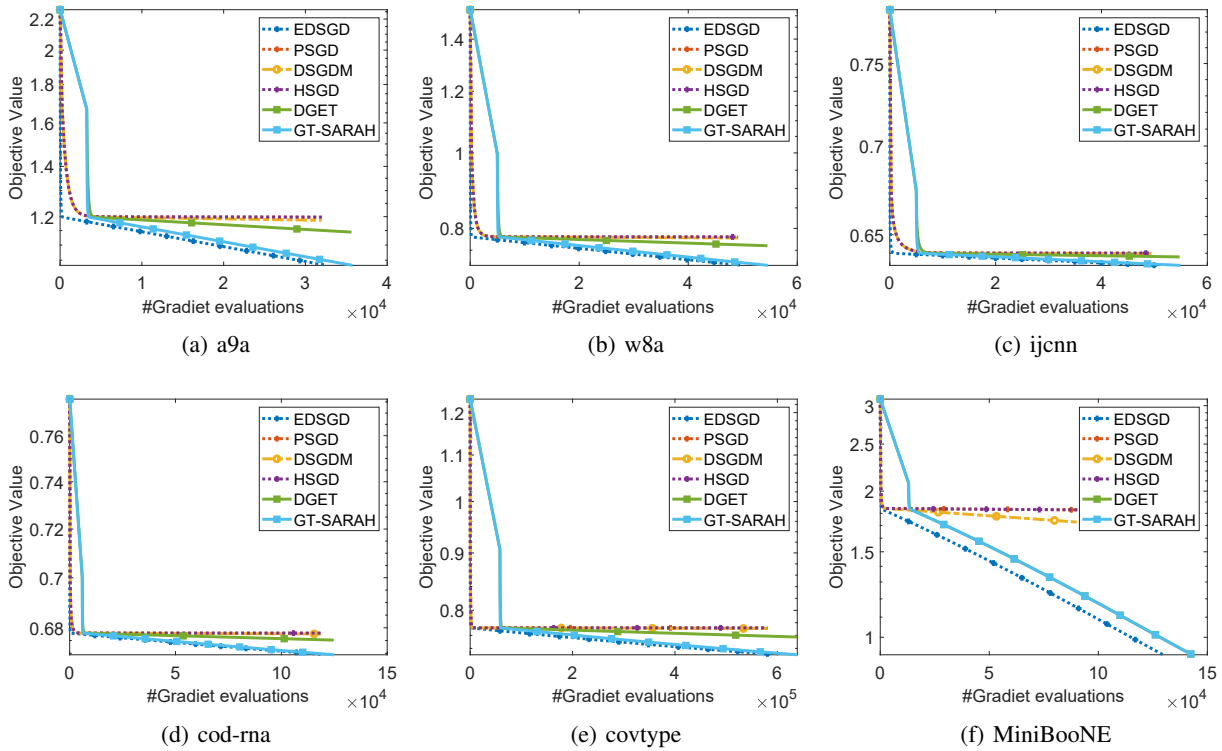
In Figure 1, we plot the loss function value with respect to the number of gradient evaluations. It can be observed that our proposed EDSGD method converges faster than DGET and GT-SARAH, confirming that our method is sample efficient. The reason is that our method does not need to periodically compute the full gradient. In Figure 2, we plot the gradient norm with respect to the number of gradient evaluations. Similarly, EDSGD outperforms DGET and GT-SARAH, which further confirms the effectiveness of our method. In summary, our method outperforms baseline methods theoretically and empirically.

Conclusion

In this paper, we developed a novel decentralized stochastic gradient descent method. Specifically, our method does not need to compute the full gradient as existing methods. We further developed novel techniques to bound the consensus error and a new Lyapunov function to establish the convergence rate of our methods, showing that our method enjoy better sample and communication complexities than existing methods. Both the theoretical and empirical results demonstrate that our method is superior to existing methods.

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

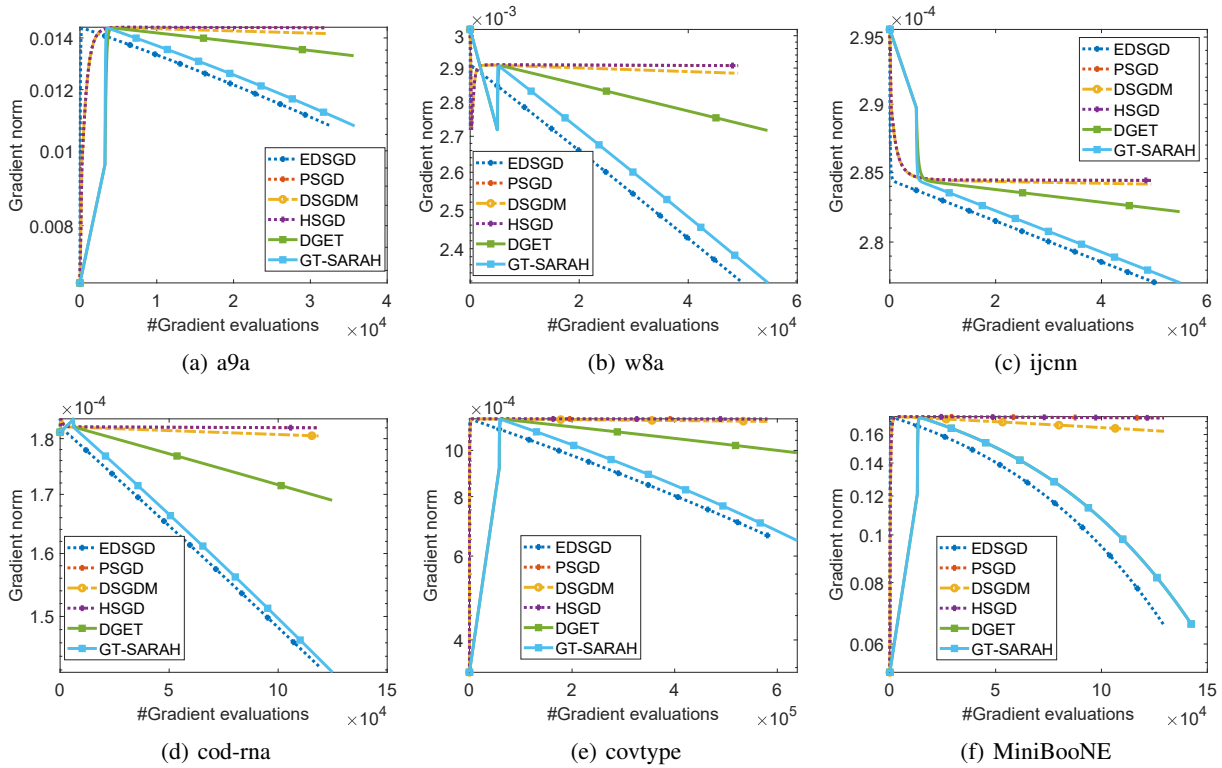
[2] https://www.openml.org



(a) a9a (b) w8a (c) ijcnn

(d) cod-rna (e) covtype (f) MiniBooNE

Figure 1: The objective function value versus the number of gradient evaluations.



(a) a9a (b) w8a (c) ijcnn

(d) cod-rna (e) covtype (f) MiniBooNE

Figure 2: The gradient norm versus the number of gradient evaluations.

References

Cutkosky, A.; and Orabona, F. 2019. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, 1646–1654.

Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*.

Gao, H.; and Huang, H. 2020. Periodic stochastic gradient descent with momentum for decentralized training. *arXiv preprint arXiv:2008.10435*.

Gao, H.; and Huang, H. 2021. Fast Training Method for Stochastic Compositional Optimization Problems. *Advances in Neural Information Processing Systems*, 34.

Gao, H.; Xu, H.; and Vucetic, S. 2021. Sample Efficient Decentralized Stochastic Frank-Wolfe Methods for Continuous DR-Submodular Maximization. *Thirtieth International Joint Conference on Artificial Intelligence*.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26: 315–323.

Koloskova, A.; Loizou, N.; Boreiri, S.; Jaggi, M.; and Stich, S. 2020. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, 5381–5393. PMLR.

Koloskova, A.; Stich, S.; and Jaggi, M. 2019. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, 3478–3487. PMLR.

Li, B.; Cen, S.; Chen, Y.; and Chi, Y. 2020. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. In *International Conference on Artificial Intelligence and Statistics*, 1662–1672. PMLR.

Li, Z.; and Richtárik, P. 2021. ZeroSARAH: Efficient Nonconvex Finite-Sum Optimization with Zero Full Gradient Computation. *arXiv preprint arXiv:2103.01447*.

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. *arXiv preprint arXiv:1705.09056*.

Lin, T.; Karimireddy, S. P.; Stich, S. U.; and Jaggi, M. 2021. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2102.04761*.

Lu, S.; Zhang, X.; Sun, H.; and Hong, M. 2019. GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, 315–321. IEEE.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, 2613–2621. PMLR.

Pu, S.; and Nedić, A. 2021. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187(1): 409–457.

Qureshi, M. I.; Xin, R.; Kar, S.; and Khan, U. A. 2021. Push-SAGA: A decentralized stochastic algorithm with variance reduction over directed graphs. *IEEE Control Systems Letters*.

Shi, W.; Ling, Q.; Wu, G.; and Yin, W. 2015. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2): 944–966.

Sun, H.; Lu, S.; and Hong, M. 2020. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, 9217–9228. PMLR.

Tang, H.; Gan, S.; Zhang, C.; Zhang, T.; and Liu, J. 2018. Communication compression for decentralized training. *arXiv preprint arXiv:1803.06443*.

Wang, J.; Sahu, A. K.; Yang, Z.; Joshi, G.; and Kar, S. 2019. Matcha: Speeding up decentralized sgd via matching decomposition sampling. In *2019 Sixth Indian Control Conference (ICC)*, 299–300. IEEE.

Xin, R.; Khan, U. A.; and Kar, S. 2020. Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68: 6255–6271.

Xin, R.; Khan, U. A.; and Kar, S. 2021. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. *arXiv preprint arXiv:2102.06752*.

Xin, R.; Khan, U. A.; and Kar, S. 2022. Fast Decentralized Nonconvex Finite-Sum Optimization with Recursive Variance Reduction. *SIAM Journal on Optimization*, 32(1): 1–28.

Yu, H.; Jin, R.; and Yang, S. 2019. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *International Conference on Machine Learning*, 7184–7193. PMLR.

Yuan, K.; Chen, Y.; Huang, X.; Zhang, Y.; Pan, P.; Xu, Y.; and Yin, W. 2021. DecentLaM: Decentralized Momentum SGD for Large-batch Deep Training. *arXiv preprint arXiv:2104.11981*.

Zhang, X.; Liu, J.; Zhu, Z.; and Bentley, E. S. 2021a. GT-STORM: Taming Sample, Communication, and Memory Complexities in Decentralized Non-Convex Learning. *arXiv preprint arXiv:2105.01231*.

Zhang, X.; Liu, J.; Zhu, Z.; and Bentley, E. S. 2021b. Low Sample and Communication Complexities in Decentralized Learning: A Triple Hybrid Approach. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10. IEEE.

Zhou, D.; Xu, P.; and Gu, Q. 2018. Stochastic nested variance reduction for nonconvex optimization. *arXiv preprint arXiv:1806.07811*.