This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

Distributed Adaptive Gradient Algorithm with Gradient Tracking for Stochastic Non-Convex Optimization

Dongyu Han, Kun Liu, *Senior Member, IEEE,* Yeming Lin, Yuanqing Xia, *Fellow, IEEE*

***Abstract*—This paper considers a distributed stochastic non-convex optimization problem, where the nodes in a network cooperatively minimize a sum of L-smooth local cost functions with sparse gradients. By adaptively adjusting the stepsizes according to the historical (possibly sparse) gradients, a distributed adaptive gradient algorithm is proposed, in which a gradient tracking estimator is used to handle the heterogeneity between different local cost functions. We establish an upper bound on the optimality gap, which indicates that our proposed algorithm can reach a first-order stationary solution dependent on the upper bound on the variance of the stochastic gradients. Finally, numerical examples are presented to illustrate the effectiveness of the algorithm.**

***Index Terms*—Distributed non-convex optimization, stochastic gradient, adaptive gradient algorithm, gradient tracking.**

I. Introduction

WITH the rapid development of big data, distributed optimization has raised interest in the fields of signal processing, machine learning and robot networks [1]–[3] due to its advantages in high computation and communication efficiency as well as the robustness to network uncertainty [4]. In distributed optimization, the computation nodes usually need to communicate with each other to minimize a finite-sum cost function, where each node only has access to partial knowledge of the entire network.

According to the convexity of local cost functions and constraints, most studies on distributed optimization can be classified into two categories: distributed convex optimization and distributed non-convex optimization. Distributed convex optimization has been widely investigated, and various algorithms have been developed over the past decade, e.g., distributed gradient descent [5]–[8], distributed primal-dual gradient [9], [10], distributed dual averaging [11], [12], distributed mirror descent [13], [14] and distributed proximal gradient algorithms [15], [16]. On the other hand, non-convex optimization problems also widely exist in practical scenarios. For example, in deep neural networks, the cost function is usually non-convex due to the interaction of multiple hidden layers with nonlinear activation functions [17], and in robotic networks, the physical constraints of robots sometimes are highly non-convex [18]. Compared to convex optimization problems, the algorithm design and analysis of non-convex optimization problems are relatively complex because of the absence of good properties of convexity [19]. Therefore, it is also of great necessity to investigate algorithms for distributed non-convex optimization problems. The distributed gradient descent algorithm for non-convex optimization problem was investigated in [20], where the cost functions are assumed to be L-smooth. In [21], a Distributed Stochastic Gradient Descent (DSGD) algorithm, in which the iterative solution can achieve a local minimum called ϵ-accurate stationary solution, was further developed for stochastic non-convex optimization problems, while the heterogeneity between local cost functions is not considered. It was investigated in [22] and [23] that the heterogeneity over the network could affect the stationarity performance of the DSGD algorithm for non-convex optimization problems through an additional bias term on the optimality gap, and the iterative solution may suffer from a consensus error [24].

To deal with the heterogeneity, a distributed Gradient Tracking (GT) algorithm was proposed in [25] for non-convex optimization problems over balanced networks. In distributed GT algorithm, an additional GT vector is introduced to estimate the global gradient of the whole cost function, which helps to find a stationary solution under the impact of heterogeneity. A stochastic variant of distributed GT algorithm was developed in [26] for the distributed stochastic empirical risk minimization problem with non-convex and L-smooth cost functions. Furthermore, a distributed derivative-free GT algorithm was proposed in [27], where the zero-order stochastic GT estimator is used in the iterations. In [28], the distributed stochastic GT algorithm was also extended to non-convex optimization problems over unbalanced networks. More recently, several variance reduction techniques were employed in distributed GT algorithm to reduce the variances of the stochastic gradient estimators [4], [19], [29]. Note that the above algorithms involve the use of pre-designed stepsizes in the iterations.

On the other hand, the data in real-world applications is usually complicated, which may result in the sparse gradients. When dealing with such sparsity, the aforementioned algorithms with pre-designed stepsizes may suffer from slow convergence with bad-chosen hyper-parameters, and show poor performance since the gradient is scaled uniformly in all dimensions [30], [31]. To overcome this problem, adap-

This work was supported in part by the National Natural Science Foundation of China under Grants 62273041 and 61873034, in part by the Joint Open Foundation of the State Key Laboratory of Synthetical Automation for Process Industries under Grant 2021-KF-21-05, in part by the Graduate Research and Innovation Training Foundation of Beijing Institute of Technology under Grant 2023YCXY034. *(Corresponding author: Kun Liu.)*

Dongyu Han, Kun Liu, Yeming Lin and Yuanqing Xia are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mails: handongyu@bit.edu.cn; kunliubit@bit.edu.cn; yeminglin@bit.edu.cn; xia_yuanqing@bit.edu.cn).

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

tive gradient algorithms [32]–[36] have received increasing attention since less tweaking of hyper-parameters are required to achieve satisfactory performance even if the gradients are sparse.

Moreover, a distributed adaptive gradient algorithm based on momentum gradient decent was presented in [37] under the convex setting. In [37], each dimension of the gradient is rescaled by adjusting the stepsizes based on the historical gradients, which leads to good performance on problems with sparse gradients. A distributed adaptive gradient algorithm with bounded stepsizes was further studied in [31] to improve the generalization capacity. By introducing a GT estimator, a novel distributed adaptive algorithm was developed in the notable work [38], which is proved to achieve a linear convergence rate under the strongly-convex setting.

Motivated by the above discussion, this paper investigates a distributed adaptive gradient algorithm for addressing distributed stochastic non-convex optimization problems with L-smooth cost functions. The main contributions are as follows:

(a) We propose a GT-based distributed adaptive gradient algorithm, in which each node performs a momentum gradient descent based on the adaptive stepsizes to find a stationary solution. The adaptive stepsizes are generated according to the historical gradients, enabling the algorithm to automatically coordinate the stepsizes among dimensions when the gradients are sparse. Inspired by [38], we utilize a GT estimator to aggregate the gradients over the network. Moreover, an clipping operator is used to mitigate the negative effects of extreme adaptive stepsizes.

(b) We provide a rigorous stationarity analysis for our proposed algorithm under the non-convex setting. We find that the GT estimator plays a crucial role in handling the heterogeneity of different local cost functions, since it can not only aggregate the directions of momentum gradient descent but also mitigate the disagreement of adaptive stepsizes between different nodes, as shown in Lemmas 2 and 5, respectively. These characteristics enable our algorithm to find a stationarity solution of the distributed stochastic non-convex optimization problem.

(c) It is shown that the upper bound on the optimality gap is of the order $O(1/T+\sigma^2)$ with T the iteration number and σ^2 an upper bound on the variance of the stochastic gradients, which aligns with the one observed in centralized adaptive gradient algorithm [39] for stochastic non-convex optimization problems.

Notations: Let $\mathbb{R}$ and $\mathbb{R}^+$ denote the set of real number and positive real number, respectively. Denote $\mathbb{R}^n$ as the n-dimensional real column vector space, and let $\mathbb{R}^{m\times n}$ be the set of $m\times n$-dimensional real matrix. Denote the inner product of two real vectors $x, y \in \mathbb{R}^d$ by $\langle x, y\rangle$. The notation $[x]_i$ stands for the i-th entry of vector x. For a matrix A, we use A' to denote its transpose and use $[A]_{ij}$ to denote its i, j-th entry. The notations $\|A\|$ and $\|A\|_F$ denote the spectral norm and the Frobenius norm of A, respectively. Denote the spectral radius of matrix $A \in \mathbb{R}^{n\times n}$ as $\rho(A)$. We use $\mathrm{diag}\{a_1, \ldots, a_n\}$ to denote the diagonal matrix that consists of the scalars $a_1, \ldots, a_n$, and the block diagonal matrix $\mathrm{blk\ diag}\{A_1, \ldots, A_n\}$ is defined in a similar way with $A_1, \ldots, A_n$ some real matrices. The notations $\max\{\cdot\}$ and $\min\{\cdot\}$ denote the maximum and the minimum element in $\{\cdot\}$, respectively. The column vectors of all ones and zeros with size n are denoted by $\mathbf{1}_n$ and $\mathbf{0}_n$, respectively. The $n\times n$ dimensional identity matrix is denoted by I_n. The Hardmard product and the Kronecker product is represented as '$\odot$' and '$\otimes$', respectively. Let $\mathbb{E}[\cdot]$ denote the expectation of random variables. We use $O(\cdot)$ to describe the limiting behavior of a function, e.g., for functions $f : \mathbb{R}^+ \to \mathbb{R}$ and $g : \mathbb{R}^+ \to \mathbb{R}^+$, we say $f(t) \le O(g(t))$ if there exist positive real numbers M and t_0 such that $|f(t)| \le Mg(t)$ for all $t \ge t_0$.

II. Problem formulation

Consider a non-convex optimization problem over a distributed network with n nodes. The nodes can communicate over an undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ and $\mathcal{E}$ are the node and the edge sets, respectively, and $A = [A_{ij}] \in \mathbb{R}^{n\times n}$ represents the weighted adjacency matrix. The nodes in $\mathcal{V}$ aim to solve the following consensus-based distributed optimization problem

$$\begin{aligned} \min_{\{x_i\}_{i=1}^n} \ & \frac{1}{n}\sum_{i=1}^n f_i(x_i) \\ \text{s.t.}\ & x_i = x_j,\ \ \forall i, j \in \mathcal{V} \end{aligned} \tag{1}$$

with

$$f_i(x_i) \triangleq \mathbb{E}_{\xi_i}[F_i(x_i, \xi_i)], \tag{2}$$

where $x_i \in \mathbb{R}^d$ is the decision variable of node i, $f_i : \mathbb{R}^d \to \mathbb{R}$ is the local cost function of node i that is dependent on random variable ξ_i and private (possibly non-convex) function F_i. Node i can evaluate the stochastic (possibly sparse) gradient $\nabla F_i(x_i, \xi_i)$ at the point x by randomly sampling ξ_i from a local distribution. Note that the functions F_i, for all $i \in \mathcal{V}$, are allowed to be different among the nodes, and the random variables ξ_i, for all $i \in \mathcal{V}$, may be sampled from different distributions, which leads to the heterogeneity of local cost functions. We also denote $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ as the average cost function over the network.

Then, we make the some assumptions on the above problem.

Assumption 1. *For $i \in \mathcal{V}$, the cost function f_i is differentiable and L-smooth for some positive scalar L, i.e., $\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|$, which is equivalent to $f_i(y) \le f_i(x) - \langle \nabla f_i(x), x - y\rangle + \frac{L}{2}\|x - y\|^2$, holds for any $x, y \in \mathbb{R}^d$.*

Assumption 2. *The stochastic gradient $\nabla F_i(x_i, \xi_i)$, $i \in \mathcal{V}$, satisfies the following conditions:*

(a) *$\nabla F_i(x_i, \xi_i)$ is an unbiased estimate of the true gradient, i.e., $\mathbb{E}_{\xi_i}[\nabla F_i(x_i, \xi_i)] = \nabla f_i(x_i)$.*

(b) *There exists a scalar $G > 0$ such that $\|\nabla F_i(x_i, \xi_i)\| \le G$ almost surely holds for any $x_i \in \mathbb{R}^d$ and ξ_i.*

(c) *The variance of the stochastic gradient is bounded, i.e., there exists a scalar $\sigma > 0$ such that $\mathbb{E}_{\xi_i}\|\nabla F_i(x_i, \xi_i) - \nabla f_i(x_i)\|^2 \le \sigma^2$.*

Remark 1. *Assumption 2(b) is commonly used in the works on adaptive gradient algorithms [31], [34]–[36], [40], [41] for*

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

stochastic non-convex optimization problems. This assumption implies that the gradient of the cost function is Lipschitz bounded, i.e., $\|\nabla f_i(x_i)\| \leq G$, which can be satisfied by a wide range of cost functions, e.g., the Huber function and Geman-McClure function in robust optimization [42] as well as the negative log-likelihood function for logistic regression [35]. On the other hand, Assumption 2(b) also implies that the stochastic error between the stochastic gradient and the true gradient, i.e., $\|\nabla F_i(x_i, \xi_i) - \nabla f_i(x_i)\|$, is bounded. This type of bounded stochastic error or noises, including the truncated Gaussian noise and the bounded uniform noise, commonly arises in various applications. For instance, in engineering applications, sensors often have limited measuring ranges, which leads to bounded noises. Moreover, in signal processing and data analysis tasks, extremely large noisy data is commonly considered as anomalous signal during the preprocessing, and then, is excluded from further computation. As a result, these scenarios naturally lead to the almost surely bounded gradient in Assumption 2(b).

Since node i only has information about F_i and ξ_i, all the nodes need to exchange information over the network in order to solve problem (1). The node j can receive information from node i if edge $(i, j) \in \mathcal{E}$. Denote $\mathcal{N}_i = \{j|(i, j) \in \mathcal{E}\}$ as the neighbor set of node i. Then, a basic assumption on the graph is given as follows:

Assumption 3. *The graph $\mathcal{G}$ is connected. Moreover, the weighted adjacency matrix A is doubly stochastic and satisfies $\rho\big(A - \frac{\mathbf{1}_n \mathbf{1}_n'}{n}\big) < 1$.*

Remark 2. *Assumption 3 can be satisfied in connected undirected networks through properly designed weight setting protocols, such as Metropolis weight protocol [43].*

In our considered distributed stochastic non-convex optimization problem, since each node maintains a local solution x_i, we employ the following metric:

$$\mathbb{E}\left[\|\nabla f(\bar{x})\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_i - \bar{x}\|^2\right] \tag{3}$$

with $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, to evaluate the first-order stationarity of the solutions $\{x_i\}_{i=1}^{n}$ of all the n nodes over the network. The metric (3), also known as optimality gap [4], captures both the gradient norm and the consensus error of the solutions, providing a comprehensive evaluation of the stationarity performance.

III. Algorithm Design

In this section, we present our proposed GT-based distributed adaptive gradient algorithm, as shown in Algorithm 1.

At t-th iteration, node i maintains a *local estimate* $x_{t,i} \in \mathbb{R}^d$ of the solution of the distributed optimization problem (1) and can evaluate the stochastic gradient $g_{t,i} \triangleq \nabla F_i(x_{t,i}, \xi_i)$. The vector $m_{t,i} \in \mathbb{R}^d$ denotes the *momentum gradient*. The vector $s_{t,i} \in \mathbb{R}^d$ represents the *GT estimator*, which is designed to track the gradient $\nabla f(x)$ of the global cost function. The notation $v_{t,i} \in \mathbb{R}^d$ represents the *adaptive vector* to rescale the stepsizes, while the vector $\hat{v}_{t,i} \in \mathbb{R}^d$ and the matrix $V_{t,i} \in \mathbb{R}^{d\times d}$ are auxiliary adaptive variables. In addition, the scalars $\beta_1 \in (0, 1)$ and $\beta_2 \in (0, 1)$ denote the exponential decay rates of $m_{t,i}$ and $v_{t,i}$, respectively, and $\alpha > 0$ is the stepsize.

Algorithm 1 GT-based distributed adaptive gradient algorithm

```
Initialization: Parameters $\beta_1 \in (0,1), \beta_2 \in (0,1)$, initial stepsize $\alpha > 0$, $v_{\max} \geq 1 \geq v_{\min} > 0$, iteration number $T$, and the initial states $\forall x_{1,i} \in \mathbb{R}^d$, $s_{1,i} = g_{1,i}$, $m_{1,i} = \mathbf{0}_d$ and $v_{1,i} = s_{1,i} \odot s_{1,i}$.
for $t = 1, \ldots, T$, node $i \in \mathcal{V}$ do
    Communicate the local estimate $x_{t-1,i}$ and the GT estimator $s_{t-1,i}$ with its neighbors.
    Update
        $m_{t+1,i} = \beta_1 m_{t,i} + (1-\beta_1) s_{t,i}$,   (4)
        $\hat{v}_{t+1,i} = \beta_2 v_{t,i} + (1-\beta_2) s_{t,i} \odot s_{t,i}$,   (5)
        $v_{t+1,i} = \mathrm{Clip}(\hat{v}_{t+1,i}, v_{\min}, v_{\max})$,   (6)
        $V_{t+1,i} = \mathrm{diag}\{[v_{t+1,i}]_1, \ldots, [v_{t+1,i}]_d\}$,   (7)
        $x_{t+1,i} = \sum_{j=1}^{n} A_{ij} x_{t,j} - \alpha V_{t+1,i}^{-1/2} m_{t+1,i}$,   (8)
        $s_{t+1,i} = \sum_{j=1}^{n} A_{ij} s_{t,j} + g_{t+1,i} - g_{t,i}$,   (9)
    where the stochastic gradient $g_{t,i} = \nabla F_i(x_{t,i}, \xi_i)$.
end for
Output: The local estimate $x_{T+1,i}$
```

As shown in (4)-(7), at t-th iteration, node i first performs an exponential moving averaging to update the *momentum gradient* $m_{t,i}$ as well as the *adaptive vector* $v_{t,i}$ based on the *GT estimator* $s_{t-1,i}$. Then, node i gathers information from its neighbors $j \in \mathcal{N}_i$ and updates the *local estimate* $x_{t,i}$ as in (8) based on a distributed momentum gradient descent. In (8) the descent direction is determined by the *momentum gradient* $m_{t,i}$, and the stepsizes in different dimensions are rescaled by matrix $\alpha V_{t,i}^{-1/2}$ to handle sparse gradients. Specifically, the stepsize is enlarged adaptively when it is in the dimension with a small gradient, and vice versa. To avoid extreme values of the adaptive stepsizes, an element-wise clipping operation

$$\mathrm{Clip}(v, v_{\min}, v_{\max}) \triangleq \max\{\min\{v, v_{\max}\}, v_{\min}\} \tag{10}$$

with $v_{\max}$ and $v_{\min}$ the upper and lower bound of $v_{t,i}$, respectively, is equipped to bound the value of $\hat{v}_{t,i}$. In addition, as in (9), the update of the *GT estimator* $s_{t,i}$ involves an accumulation of the gradient innovations, i.e., $g_{t,i} - g_{t-1,i}$, as well as a weighted average consensus operation to track the gradient $\nabla f(x)$ of the global cost function. Finally, node i outputs the *local estimate* $x_{T+1,i}$ of the solution.

Remark 3. *Compared to the GT-based distributed adaptive gradient algorithm for a strongly-convex problem in [38], our work aims to investigate the stationarity performance of the proposed Algorithm 1 for a more challenging distributed stochastic non-convex optimization problem. A notable characteristic of Algorithm 1 is the utilization of a clipping operator*

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

as in (6), which distinguishes our algorithm from the algorithm in [38] with an additive bias term. Specifically, in [38], an easily implemented additive bias term ϵ is employed on each $v_{t,i}$ to ensure that the adaptive vector $v_{t,i}+\epsilon$ is consistently greater than or equal to the constant ϵ. In contrast, our clipping operator provides a more direct approach to mitigate the negative impact of extreme stepsizes, since the adaptive vector will be clipped only if $v_{t,i}$ is smaller or larger than the threshold $v_{\min}$ or $v_{\max}$, respectively. Furthermore, we provide a rigorous stationarity analysis for our Algorithm 1, which establishes an explicit upper bound on the optimality gap, as demonstrated in Corollary 1 below. It is important to note that our analysis approach can also be extended and applied to the algorithm in [38] by adjusting the coefficients $v_{\min}, v_{\max}$ in our analysis according to the value of additive bias term ϵ in [38].

IV. Convergence Analysis

In this section, we provide the convergence analysis of our proposed algorithm for non-convex optimization problem in terms of the *optimality gap* given by (3).

We first introduce several auxiliary vectors

$$z_{t,i} \triangleq \begin{cases} x_{t,i}, & t=1, \\ \frac{1}{1-\beta_1}x_{t,i} - \frac{\beta_1}{1-\beta_1}x_{t-1,i}, & t\geq 2. \end{cases} \tag{11}$$

The notation

$$\boldsymbol{\zeta}_t \triangleq [\zeta_{t,1}', \dots, \zeta_{t,n}']' \in \mathbb{R}^{nd} \tag{12}$$

aggregates all the vectors $\zeta_{t,i}$ of the nodes $i \in \{1,\dots,n\}$, while the notations

$$\bar{\zeta}_t \triangleq \frac{1}{n}\sum_{i=1}^{n}\zeta_{t,i} \in \mathbb{R}^d, \text{ and } \tilde{\boldsymbol{\zeta}}_t \triangleq \mathbf{1}_n \otimes \bar{\zeta}_t \in \mathbb{R}^{nd} \tag{13}$$

represent the average vectors, where ζ can be any variables such that $\zeta \in \{x,s,z,m,v\}$. Moreover, define $V_t \triangleq \operatorname{blk\,diag}\{V_{t,1},\dots,V_{t,n}\} \in \mathbb{R}^{nd\times nd}$ as the aggregation form of the adaptive matrix $V_{t,i}$, and $\bar{V}_t \triangleq \frac{1}{n}\sum_{i=1}^{n}V_{t,i}$. Let $\eta_{t,i} = \nabla F_i(x_{t,i},\xi_i) - \nabla f_i(x_{t,i})$ represents the error between the stochastic gradient $\nabla F_i(x_{t,i},\xi_i)$ and the true gradient $\nabla f_i(x_{t,i})$.

With auxiliary vector $\bar{z}_t$ defined in (13), in the following lemma we establish an upper bound on the evolution of $f(\bar{z}_t)$ based on the L-smooth property of cost function f_i.

Lemma 1. *Under Assumptions 1 and 2(b), we have the following result:*

$$\begin{aligned} f(\bar{z}_{t+1}) \leq{}& f(\bar{z}_t) - \alpha\Big(v_{\max}^{-1/2} - \frac{\alpha v_{\min}^{-1/2}}{2} - \alpha v_{\min}^{-1}(L+1)\Big)\|\nabla f(\bar{x}_t)\|^2 \\ &+ M_1\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\|^2 + M_2\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2 + M_3\|\boldsymbol{m}_t - \tilde{\boldsymbol{m}}_t\|^2 \\ &+ M_4\|\tilde{\boldsymbol{m}}_t\|^2 + M_5\|\boldsymbol{v}_t - \tilde{\boldsymbol{v}}_t\|^2 + M_6\sum_{i=1}^{n}\|\eta_{t,i}\|^2 \end{aligned} \tag{14}$$

with

$$\begin{aligned} M_1 &= \frac{4v_{\min}^{-1}L}{n}\Big(v_{\max}^{1/2} + 2\alpha^2(L+1)\Big), \\ M_2 &= \frac{18v_{\min}^{-1}}{n}\Big(v_{\max}^{1/2} + 2\alpha^2(L+1)\Big), \\ M_3 = M_4 &= \frac{\beta_1^2 v_{\min}^{-1}}{n(1-\beta_1)^2}\Big(16v_{\max}^{1/2} + 32\alpha^2(L+1) + \alpha^2L^2\Big), \\ M_5 &= \frac{G^2 v_{\min}^{-3}}{n}\Big(v_{\max}^{1/2} + 2\alpha^2(L+1)\Big), \\ M_6 &= \frac{4v_{\min}^{-1}}{n}\Big(v_{\max}^{1/2} + 2\alpha^2(L+1)\Big). \end{aligned} \tag{15}$$

Proof. The proof can be found in [44, Appendix B]. □

Next, we focus on the following five error quantities on the right-hand side of (14):

(i) the consensus error $\|\boldsymbol{m}_t - \tilde{\boldsymbol{m}}_t\|^2$ of *momentum gradient*;
(ii) the second-order moment $\|\tilde{\boldsymbol{m}}_t\|^2$ of average *momentum gradient*;
(iii) the consensus error $\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\|^2$ of *local estimate*;
(iv) the consensus error $\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2$ of *GT estimator*;
(v) the consensus error $\|\boldsymbol{v}_t - \tilde{\boldsymbol{v}}_t\|^2$ of *adaptive vector*.

In the following Lemmas 2-6, we will establish the upper bounds on the expected summation over time horizon T of the above five quantities respectively.

Lemma 2. *(Consensus error of momentum gradient) Consider the iterates generated by Algorithm 1. It holds for $T \geq 1$ that*

$$\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{m}_t - \tilde{\boldsymbol{m}}_t\|^2] \leq 4\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2]. \tag{16}$$

Proof. The proof can be found in [44, Appendix C-I]. □

Lemma 3. *(Consensus error of local estimate) Under Assumption 3, the following inequality*

$$\begin{aligned} \sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\|^2] \leq{}& \frac{40\alpha^2 v_{\min}^{-1}}{(1-\rho_A^2)^2}\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2] \\ &+ \frac{2}{1-\rho_A^2}\Delta_1 \end{aligned} \tag{17}$$

holds for $T \geq 1$, where $\Delta_1 = \|\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1\|^2$ and

$$\rho_A = \rho\Big(A - \frac{\mathbf{1}_n\mathbf{1}_n'}{n}\Big). \tag{18}$$

Proof. The proof can be found in [44, Appendix C-II]. □

Lemma 4. *(Norm of momentum gradient) Under Assumption 1, the following inequality holds for $T \geq 1$:*

$$\begin{aligned} \sum_{t=1}^{T}\mathbb{E}\big[\|\tilde{\boldsymbol{m}}_t\|^2\big] \leq{}& \frac{240L^2\alpha^2 v_{\min}^{-1}}{(1-\rho_A^2)^2}\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2] \\ &+ 6n\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla f(\bar{x}_t)\|^2\big] + 6n\sigma^2T + \frac{12L^2}{1-\rho_A^2}\Delta_1. \end{aligned} \tag{19}$$

Proof. The proof can be found in [44, Appendix C-III]. □

Lemma 5. *(Consensus error of adaptive vector) Under Assumption 2(b), the following inequality*

$$\begin{aligned} \sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{v}_t - \tilde{\boldsymbol{v}}_t\|^2] \leq{}& \frac{36G^2}{(1-\rho_A)^2}\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2] \\ &+ \frac{2}{1-\beta_2^2}\Delta_2 \end{aligned} \tag{20}$$

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

holds for $T \geq 1$, where $\Delta_2 = \|\boldsymbol{v}_1 - \tilde{\boldsymbol{v}}_1\|^2$.

Proof. The proof can be found in [44, Appendix C-IV]. □

Lemma 6. *(Consensus error of GT estimator) Under Assumptions 1 and 3, for $\alpha^2 \leq \min\left\{\frac{1}{2N_1}, \frac{v_{\min}(1-\rho_A^2)^2}{72}\right\}$ and $T \geq 1$ it holds that*

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}\left[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2\right] \leq 2n\alpha^2 N_2 \sum_{t=1}^{T}\mathbb{E}\left[\|\nabla f(\bar{x}_t)\|^2\right] \\ + 2nTN_3\sigma^2 + 2N_4\Delta,\end{aligned} \tag{21}$$

where ρ_A is given by (18), $\Delta = \sum_{i=1}^{3}\Delta_i$ with Δ_1, Δ_2 given in Lemmas 3 and 5, respectively, while $\Delta_3 = \|\boldsymbol{s}_1 - \tilde{\boldsymbol{s}}_1\|^2$, and

$$\begin{aligned}N_1 &= \frac{14400L^2 v_{\min}^{-1}}{(1-\rho_A^2)^4} + \frac{72v_{\min}^{-1}}{(1-\rho_A^2)^2}\left(4 + \frac{20L^2}{3}\right), \\ N_2 &= \frac{504v_{\min}^{-1}}{(1-\rho_A^2)^2}, \qquad N_3 = 12 + \frac{216}{(1-\rho_A^2)^2}, \\ N_4 &= \max\left\{\frac{720L^2}{(1-\rho_A^2)^5} + \frac{24L^2}{1-\rho_A^2}, \frac{2\beta_1^2}{1-\beta_1^2}, \frac{1}{2}\right\}.\end{aligned} \tag{22}$$

Proof. The proof can be found in [44, Appendix C-V]. □

Next, we present our main result, which provides an upper bound on the average gradient norm by Lemmas 1-6.

Theorem 1. *Suppose Assumptions 1-3 hold, and there exists $f^* \in \mathbb{R}$ such that $f(x) \geq f^*$ holds for all $x \in \mathbb{R}^d$. For any positive scalar $\omega > 0$, by choosing parameter α such that $\alpha^2 < \min\left\{\frac{1}{2N_1}, \frac{v_{\min}(1-\rho_A^2)^2}{72}, \frac{v_{\max}^{-1}}{(N_2')^2}\right\}$ and β_1 such that $\frac{\beta_1^2}{\omega(1-\beta_1)^2} \leq \alpha^2$, then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla f(\bar{x}_t)\|^2\right] \leq O\left(\frac{f(\bar{x}_1) - f^* + \Delta}{T} + \sigma^2\right), \tag{23}$$

where

$$\begin{aligned}N_2' = N_2\left[\frac{L}{9}\mu + \mu + \frac{72G^3 v_{\min}^{-2}}{(1-\rho_A)^2}\right] + \frac{v_{\min}^{-1/2}}{2} + v_{\min}^{-1}(L+1) \\ + \omega\left(\mu + \frac{L^2}{36}\right)\left[3 + 4N_2(L^2+1)\right]\end{aligned} \tag{24}$$

with

$$\mu = 36v_{\min}^{-1}v_{\max}^{1/2} + L + 1. \tag{25}$$

Proof. By summing (14) over $t = 1, \ldots, T$ and taking the expectation, one has

$$\begin{aligned}&\mathbb{E}[f(\bar{z}_{T+1})] - f(\bar{z}_1) \\ &\leq -\alpha\Big(v_{\max}^{-1/2} - \frac{\alpha v_{\min}^{-1/2}}{2} - \alpha v_{\min}^{-1}(L+1)\Big)\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2] \\ &\quad + M_1\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\|^2] + M_2\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2] \\ &\quad + M_3\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{m}_t - \tilde{\boldsymbol{m}}_t\|^2] + M_4\sum_{t=1}^{T}\mathbb{E}[\|\tilde{\boldsymbol{m}}_t\|^2] \\ &\quad + M_5\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{v}_t - \tilde{\boldsymbol{v}}_t\|^2] + M_6\sum_{t=1}^{T}\sum_{i=1}^{n}\|\eta_{t,i}\|^2.\end{aligned} \tag{26}$$

Then, by Lemmas 2-6 and the fact that $\alpha^2 \leq \frac{v_{\min}(1-\rho_A^2)^2}{72}$ and $\frac{\beta_1^2}{\omega(1-\beta_1)^2} \leq \alpha^2$, we rearrange the terms and get

$$\begin{aligned}\mathbb{E}[f(\bar{z}_{T+1})] \leq& f(\bar{z}_1) - \alpha v_{\max}^{-1/2}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2] \\ &+ \alpha^2 N_2'\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla f(\bar{x}_t)\|^2\right] + TN_3'\sigma^2 + N_4'\Delta\end{aligned} \tag{27}$$

with N_2' given by (24), and

$$\begin{aligned}N_3' &= \mu N_3\left[\left(\frac{L}{9} + \omega(4+4L^2) + 1\right)\right] + N_3\left(\frac{\omega L^2}{9} + \frac{L^4}{10}\right) \\ &\quad + \mu\left(\frac{1}{9} + 3\omega\right) + \frac{\omega L^2}{24}, \\ N_4' &= \frac{\mu N_4}{n}\left[\frac{10L}{81} + 1 + \omega\alpha^2\left(8 + \frac{20L^2}{3}\right) + \frac{72G^3 v_{\min}^{-2}}{(1-\rho_A)^2}\right] \\ &\quad + \max\left\{\frac{2L\mu}{9n(1-\rho_A^2)} + \frac{12\omega L^2\alpha^2}{n(1-\rho_A^2)}, \frac{G^2 v_{\min}^{-2}\mu}{18n(1-\beta_2^2)}\right\}.\end{aligned} \tag{28}$$

Note that the value of N_2' is independent of α and β_1. Then, by setting $0 < \alpha < \frac{v_{\max}^{-1/2}}{N_2'}$, which implies that $\alpha(v_{\max}^{-1/2} - \alpha N_2') > 0$, we can rearrange the terms in (27) and obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2] \leq \frac{f(\bar{x}_1) - f^* + TN_3'\sigma^2 + N_4'\Delta}{\alpha T(v_{\max}^{-1/2} - \alpha N_2')} \tag{29}$$

with N_2' given in (24) and N_3', N_4' in (28). The inequality (29) implies that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2] \leq O\left(\frac{f(\bar{x}_1) - f^* + \Delta}{T} + \sigma^2\right), \tag{30}$$

which completes the proof. □

Finally, by using Theorem 1, Lemmas 3 and 6, we establish the upper bound on the optimality gap in the following corollary.

Corollary 1. *Under the conditions of Theorem 1, it holds that*

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_{t,i} - \bar{x}_t\|^2]\right] \\ &\leq \frac{80N_2\alpha^2 v_{\min}^{-1}}{\alpha(1-\rho_A^2)^2(v_{\max}^{-1/2} - \alpha N_2')}\cdot\frac{f(\bar{x}_1) - f^*}{T} \\ &\quad + \left[\frac{80\alpha^2 v_{\min}^{-1}}{(1-\rho_A^2)^2}\left(\frac{N_2 N_4'}{\alpha(v_{\max}^{-1/2} - \alpha N_2')} + \frac{N_4}{n}\right) + \frac{2}{n(1-\rho_A^2)}\right]\cdot\frac{\Delta}{T} \\ &\quad + \frac{40\alpha^2 v_{\min}^{-1}}{(1-\rho_A^2)^2}\cdot\left(\frac{2N_2N_3'}{\alpha(v_{\max}^{-1/2} - \alpha N_2')} + 2N_3\right)\sigma^2 \\ &\leq O\Big(\frac{f(\bar{x}_1) - f^* + \Delta}{T} + \sigma^2\Big).\end{aligned} \tag{31}$$

Proof. As we have already shown in Theorem 1 that $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\bar{x}_t)\|^2] \leq O\big(\frac{f(\bar{x}_1) - f^* + \Delta}{T} + \sigma^2\big)$, we then focus on the term $\frac{1}{nT}\sum_{t=1}^{T}\mathbb{E}[\sum_{i=1}^{n}\|x_{t,i} - \bar{x}_t\|^2]$.

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

By applying Lemmas 3 and 6, it is easy to prove that both $\sum_{t=1}^{T}\mathbb{E}\left[\|\boldsymbol{s}_{t+1}-\tilde{\boldsymbol{s}}_{t+1}\|^2\right]$ and $\sum_{t=1}^{T}\mathbb{E}\left[\|\boldsymbol{x}_{t+1}-\tilde{\boldsymbol{x}}_{t+1}\|^2\right]$ are of the order $O\big(\frac{f(\bar{x}_1)-f^*+\Delta}{T}+\sigma^2\big)$. The detailed proof is provided in [44, Appendix D]. □

Remark 4. *As shown in Corollary 1, when the variance of the stochastic gradient is upper bounded by σ^2, the optimality gap of our proposed algorithm is in the order of $O(\frac{f(\bar{x}_1)-f^*+\Delta}{T}+\sigma^2)$, which is consistent with the one in centralized adaptive gradient algorithm [39] for non-convex stochastic optimization with L-smooth cost functions. Note that the optimality gap will converge to some constant determined by the variance of the stochastic gradient. To mitigate the impact of the variance on the optimality gap, some variance reduction method [4], [19], [29] can be further taken into consideration.*

Remark 5. *Distributed optimization algorithms with adaptive stepsizes were developed in [31] and [37] for convex optimization problems, in which the local gradients are used in the momentum-gradient descent. Since the heterogeneity between different local cost functions sometimes leads to the disagreement on the momentum gradients as well as the adaptive stepsizes of different nodes, the algorithms in [31] and [37] implement diminishing stepsizes to handle such disagreement. However, the effect of gradient descent will become weaker as the stepsizes decay, which may affect the convergence performance.*

To mitigate the influence of heterogeneity, in our Algorithm 1 we employ a GT estimator $\boldsymbol{s}_t$ to track the average stochastic gradient $\tilde{\boldsymbol{s}}_t$. It is proved that the convergence of the norm $\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|^2$ will result in the convergence of the consensus errors of the local estimate $\|\boldsymbol{x}_t-\tilde{\boldsymbol{x}}_t\|^2$ and the momentum vector $\|\boldsymbol{m}_t-\tilde{\boldsymbol{m}}_t\|^2$, as shown in Lemmas 2 and 3, respectively. Moreover, our Lemma 5 further highlights that a smaller value of $\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|^2$ also helps to reduce the disagreement on the adaptive vector $\|\boldsymbol{v}_t-\tilde{\boldsymbol{v}}_t\|^2$, which plays a vital role in the stationarity analysis for our considered distributed stochastic non-convex optimization problem. As a result, the utilization of the GT estimator enables our Algorithm 1 to achieve a stationary performance even in the presence of heterogeneity, as illustrated in Corollary 1.

V. Numerical Examples

In this section, the performance of our proposed Algorithm 1 is verified through numerical examples.

A. Distributed state estimation problem

We first consider a robust linear regression problem [42] based on the Huber loss function $H_\varsigma(\cdot):\mathbb{R}\to\mathbb{R}^+$, defined by

$$H_\varsigma(z)=\begin{cases}\frac{1}{2}z^2, & \text{if } |z|\le\varsigma,\\ \varsigma\left(|z|-\frac{1}{2}\varsigma\right), & \text{otherwise}\end{cases}\tag{32}$$

with scalar $\varsigma>0$.

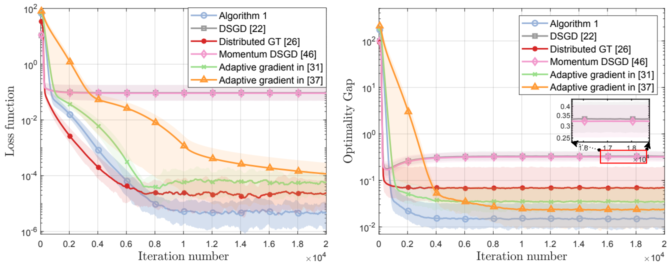
In this example, the nodes in a network are to minimize the optimization problem given by (1) with Huber-type local cost function $f_i:\mathbb{R}^m\to\mathbb{R}^+$ as follows [45]:

$$f_i(x)=\mathbb{E}\left[\boldsymbol{H}_\varsigma\left(\theta_i-\Phi_i x\right)\right],\tag{33}$$

where the matrix $\Phi_i\in\mathbb{R}^{d\times d}$ and $\theta_i\in\mathbb{R}^d$ represents the estimate target, while the function $\boldsymbol{H}_\varsigma:\mathbb{R}^m\to\mathbb{R}^m$ takes the Huber loss on each entry of vector $z=[z_j]_{j=1}^m$, i.e., $\boldsymbol{H}_\varsigma(\boldsymbol{z})=[H_\varsigma(z_j)]_{j=1}^m$.

We consider the problem setup with $n=16$, $d=10$, and use a randomly generated Erdős–Rényi (ER) graph $\mathcal{G}$ with probability 0.7. We set the parameters of our proposed Algorithm 1 as $\alpha=0.01$, $\beta_1=0.9$, $\beta_2=0.999$, $v_{\max}=100$ and $v_{\min}=10^{-8}$. For comparison, we consider the DSGD [21], distributed GT [26], Momentum DSGD [46] and the distributed adaptive gradient algorithms proposed in [31], [37]. The constant stepsize in DSGD, distributed GT, Momentum DSGD (with $\beta_1=0.9$) is set as $\alpha=0.01$, while the diminishing stepsizes in adaptive gradient algorithms [31], [37] (with $\beta_1=0.9,\beta_2=0.999$) decay in the form of $\alpha_t=\frac{100\alpha}{100+\sqrt{t}}$.

Fig. 1 illustrates the evolutions of the loss functions and the optimality gaps over $T=2\times10^4$ iterations across 50 random seeds. In the example, the optimal solution x^* is randomly generated from $[-1,1]^d$, and each matrix Φ_i is generated with eigenvalues ranging in $[0.05,1]$. Moreover, the noisy estimate target is generated following $\theta_i(t)=\Phi_i x^*+\eta_{t,i}$ with $\eta_{t,i}$ a random vector whose entries are sampled from truncated Gaussian distribution [47] with variance 0.04 and truncated threshold 0.1. The results demonstrate the superior performance of our proposed algorithm in terms of both the decay of the loss function and the optimality gap. We can observe that the DSGD and the momentum DSGD algorithms exhibit a fast rate at the beginning, however, these two algorithms converge to larger values of the loss function and the optimality gap as the iteration number increases. On the other hand, the distributed adaptive gradient algorithms proposed in [31] and [37] show a slower convergence due to their diminishing stepsizes. On the contrary, our Algorithm 1 exhibits a superior stationarity performance, displaying relatively smaller values of the loss function and the optimality gap as the iteration increases.



(a) Loss functions

(b) Optimality gaps

Fig. 1: Comparison of different algorithms on distributed robust linear regression problems. Solid curves and shaded regions represent the average value and range statistics, respectively.

B. Distributed logistic regression problem

Consider the logistic regression problem commonly arises in parameter estimation and machine learning. We present

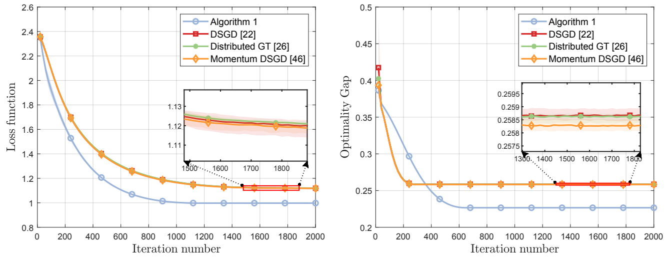
Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

the examples based on three real-world datasets, i.e., the a9a, the Covertype, and the MNIST datasets[1], where the data pieces belonging to c classes are distributed among n nodes. Denote the sampled data assigned to node i as $(y_{i,s}, l_{i,s})$, $s = 1, \dots, m$, where $y_{i,s} \in \mathbb{R}^d$ represents the features and $l_{i,s} = [l^1_{i,s}, \dots, l^c_{i,s}] \in \{0,1\} \times \cdots \times \{0,1\}$ represents the label of the s-th sample at node i using one-hot encoding. Consequently, the cost function of node i can be expressed as follows:

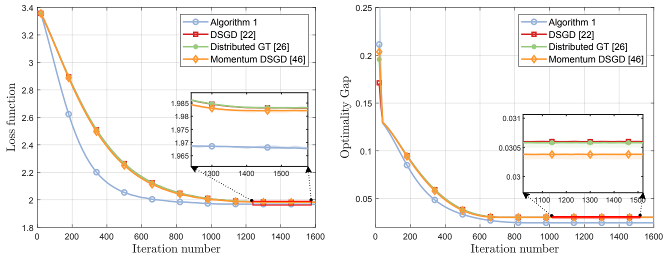
$$f_i(W) = -\frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{c} l^k_{i,j} \log\left(\frac{\exp(w_k' y_{i,j})}{\sum_{\hat{k}=1}^{c}\exp(w_{\hat{k}}' y_{i,j})}\right) + h(W), \tag{34}$$

where $W = [w_1, \dots, w_c]' \in \mathbb{R}^{c \times d}$ is the weight of the logistic regression model to be optimized and $h(W) = \sum_{j_1 \le c, j_2 \le d} \frac{0.01([w_{j_1}]_{j_2})^2}{1+([w_{j_1}]_{j_2})^2}$ is the non-convex Geman-McClure regularize function.



(a) Loss functions (b) Optimality gaps

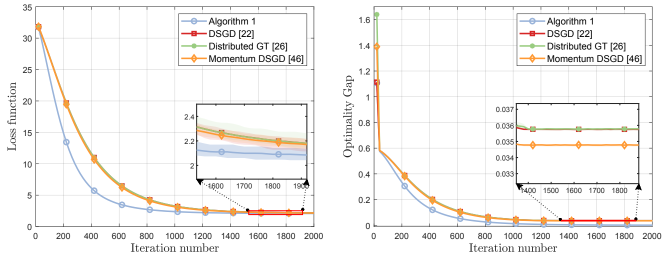
Fig. 2: Comparison of different algorithms on distributed logistic regression on a9a dataset.



(a) Loss functions (b) Optimality gaps

Fig. 3: Comparison of different algorithms on distributed logistic regression on Covertype dataset.

The datasets are evenly assigned to $n = 16$ nodes that can communicate through an ER graph. The batch size and the iteration numbers are set to be 32 and 2000, respectively. We set the parameters as $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $v_{\max} = 100$ and $v_{\min} = 10^{-8}$ for our Algorithm 1. We also compare our algorithm with some state-of-the-art distributed non-convex optimization algorithms, including DSGD, distributed GT, and Momentum DSGD (with stepsizes $\alpha = 0.1$). Numerical results conducted on the a9a, the Covtype and the MNIST datasets are presented in Figures 2-4, respectively, with the number of random seeds set to 10. It is shown that our Algorithm 1 demonstrates the best performance in terms of the training loss and the optimality gap in these three distributed logistic regression tasks with non-convex regularizer.



(a) Loss functions (b) Optimality gaps

Fig. 4: Comparison of different algorithms on distributed logistic regression on MNIST dataset.

VI. Conclusion

In this paper, we investigated a distributed adaptive gradient algorithm for stochastic non-convex optimization problems with L-smooth cost functions, where a GT estimator is employed to handle the heterogeneity between different local cost functions, and the stepsizes are adjusted adaptively to enhance the performance of our algorithm when the gradients are sparse. We analyzed the first-order stationarity performance of our proposed algorithm in terms of the optimality gap. It is shown that an upper bound of the optimality gap is of the order $O(1/T + \sigma^2)$, which aligns with the one observed in centralized adaptive gradient algorithm [39]. The effectiveness of our proposed algorithm was validated through numerical examples. Our future work includes research on distributed adaptive gradient algorithms with variance reduction and compressed communication for non-convex problems.

[1] The datasets are available at https://www.openml.org/

References

[1] S. Patterson, Y. C. Eldar, and I. Keidar, “Distributed compressed sensing for static and time-varying networks,” *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 4931–4946, 2014.

[2] D. Jia, D. Wei, S. Richard, L. Li, K. Li, and F. Li, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[3] X. Jiang, X. Zeng, J. Sun, J. Chen, and Y. Wei, “A fully distributed hybrid control framework for non-differentiable multi-agent optimization,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 10, pp. 1792–1800, 2022.

[4] H. Sun, S. Lu, and M. Hong, “Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking,” in *Proceedings of International Conference on Machine Learning*, pp. 9217–9228, 2020.

[5] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[6] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[7] J. Zhang and K. You, “AsySPA: An exact asynchronous algorithm for convex optimization over digraphs,” *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2494–2509, 2019.

[8] Q. Lü, X. Liao, T. Xiang, H. Li, and T. Huang, “Privacy masking stochastic subgradient-push algorithm for distributed online optimization,” *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3224–3237, 2020.

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2024.3380710

[9] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2011.

[10] X. Li, X. Yi, and L. Xie, "Distributed online optimization for multi-agent networks with coupled inequality constraints," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3575–3591, 2021.

[11] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2011.

[12] D. Han, K. Liu, H. Sandberg, S. Chai, and Y. Xia, "Privacy-preserving dual averaging with arbitrary initial conditions for distributed optimization," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3172–3179, 2022.

[13] D. Yuan, Y. Hong, D. W. Ho, and S. Xu, "Distributed mirror descent for online composite optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 714–729, 2020.

[14] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Transactions on Signal Processing*, vol. 68, pp. 731–746, 2020.

[15] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[16] R. Dixit, A. S. Bedi, and K. Rajawat, "Online learning over dynamic graphs via distributed proximal gradient algorithm," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5065–5079, 2020.

[17] Y. Cui, Z. He, and J.-S. Pang, "Multicomposite nonconvex optimization for training deep neural networks," *SIAM Journal on Optimization*, vol. 30, no. 2, pp. 1693–1723, 2020.

[18] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4982–4989, 2010.

[19] X. Jiang, X. Zeng, J. Sun, and J. Chen, "Distributed stochastic gradient tracking algorithm with variance reduction for non-convex optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5310–5321, 2023.

[20] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[21] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *Proceedings of International Conference on Machine Learning*, pp. 344–353, 2019.

[22] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *Proceedings of International Conference on Machine Learning*, pp. 5381–5393, 2020.

[23] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *Proceedings of International Conference on Machine Learning*, pp. 3043–3052, 2018.

[24] S. A. Alghunaim and K. Yuan, "A unified and refined convergence analysis for non-convex decentralized learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3264–3279, 2022.

[25] I. Tziotis, C. Caramanis, and A. Mokhtari, "Second order optimality in decentralized non-convex optimization via perturbed gradient tracking," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 33, pp. 21162–21173, 2020.

[26] J. Zhang and K. You, "Decentralized stochastic gradient tracking for non-convex empirical risk minimization," *arXiv: 1909.02712*, 2019.

[27] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 269–281, 2020.

[28] V. Kungurtsev, M. Morafah, T. Javidi, and G. Scutari, "Decentralized asynchronous non-convex stochastic optimization on directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 4, pp. 1796–1804, 2023.

[29] R. Xin, U. A. Khan, and S. Kar, "Variance-reduced decentralized stochastic optimization with accelerated convergence," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6255–6271, 2020.

[30] T. Chen, Z. Guo, Y. Sun, and W. Yin, "CADA: Communication-adaptive distributed ADAM," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 613–621, 2021.

[31] M. Zhang, B. Hao, Q. Ge, J. Zhu, R. Zheng, and Q. Wu, "Distributed adaptive subgradient algorithms for online learning over time-varying networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4518–4529, 2021.

[32] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 2121–2159, 2011.

[33] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv: 1212.5701*, 2012.

[34] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of ADAM and beyond," in *Proceedings of International Conference on Learning Representations*, 2018.

[35] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations*, 2015.

[36] "On the convergence of a class of ADAM-type algorithms for non-convex optimization," in *Proceedings of International Conference on Learning Representations*, 2019.

[37] X. Shen, D. Li, R. Fang, Y. Zhou, and X. Wu, "Distributed adaptive online learning for convex optimization with weight decay," *Asian Journal of Control*, vol. 24, no. 2, pp. 562–575, 2022.

[38] G. Carnevale, F. Farina, I. Notarnicola, and G. Notarstefano, "GTAdam: Gradient tracking with adaptive momentum for distributed online optimization," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 3, pp. 1436–1448, 2023.

[39] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, "Adaptive methods for nonconvex optimization," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 31, pp. 9793–9803, 2018.

[40] R. Ward, X. Wu, and L. Bottou, "Adagrad stepsizes: Sharp convergence over nonconvex landscapes," in *Proceedings of International Conference on Machine Learning*, pp. 6677–6686, 2019.

[41] X. Li and F. Orabona, "On the convergence of stochastic gradient descent with adaptive stepsizes," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, vol. 89, pp. 983–992, 2019.

[42] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[43] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[44] D. Han, K. Liu, Y. Lin, and Y. Xia, "Distributed adaptive gradient algorithm with gradient tracking for stochastic non-convex optimization," *arXiv:2403.11557*, 2024.

[45] D. Ghaderyan, N. S. Aybat, A. P. Aguiar, and F. L. Pereira, "A fast row-stochastic decentralized method for distributed optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 275–289, 2024.

[46] K. Yuan, Y. Chen, X. Huang, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "DecentLaM: Decentralized momentum SGD for large-batch deep training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3029–3039, 2021.

[47] N. Chopin, "Fast simulation of truncated gaussian distributions," *Statistics and Computing*, vol. 21, pp. 275–288, 2011.

Authorized licensed use limited to: KAUST. Downloaded on July 29,2024 at 11:26:09 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.