ZEROTH-ORDER DIFFUSION ADAPTATION OVER NETWORKS

Jie Chen[⋆] *Sijia Liu*[†] *Pin-Yu Chen*[‡]

[⋆]Center of Intelligent Acoustics and Immersive Communications
School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
[†]MIT-IBM Waston AI Lab, IBM research, Cambridge, USA
[‡]AI Foundations Group, IBM Thomas J. Watson Research Center, New York, USA
dr.jie.chen@ieee.org sijia.liu@ibm.com pin-yu.chen@ibm.com

ABSTRACT

Diffusion adaptation is an efficient strategy to perform distributed estimation over networks with streaming data. Existing diffusion-based estimation algorithms require the knowledge of analytical forms of the cost functions or their gradients associated with agents. This setting can be restrictive for practical applications where gradient calculation is difficult or systems operate in a black-box manner. Motivated by the advance of the zeroth-order (gradient-free) optimization, in this work we propose the zeroth-order (ZO) diffusion strategy using randomized gradient estimates. We also examine the stability conditions of the proposed ZO-diffusion strategy. Simulations are performed to examine properties of the algorithm and to compare it with its non-cooperative and stochastic gradient counterparts.

***Index Terms*—** Distributed estimation, online learning, diffusion adaptation, zeroth-order optimization, stochastic optimization.

1. INTRODUCTION

Distributed adaptation over networks allows a collection of interconnected nodes to perform estimation tasks from streaming measurements. For online parameter estimation, among various strategies [1–7], diffusion adaptation [6, 8] is an efficient strategy that is particularly attractive due to its enhanced adaptation performance and wider stability ranges [9]. Diffusion-based algorithms have been extensively studied, in respect of adaptation algorithms on agents, including diffusion least-mean square algorithm (LMS) [10, 11], diffusion affine projection algorithm (APA) [12], diffusion Kalman filtering [13, 14], diffusion recursive least-squares (RLS) [15], and in respect of cooperation strategies among agents [16, 17].

An inspection on existing diffusion adaption algorithms shows that they all require analytical forms of the cost functions associated with the agents during the optimization process. In many practical scenarios, this requirement can be restrictive since the explicit expression for a cost can be difficult to obtain, or the associated gradient can be difficult to compute. For example, in bandit optimization [18], a player receives partial feedback in terms of loss function values revealed by her adversary, and aims to determine the best decision based only on the observed function values. Similarly, in simulation-based optimization problems, there exist black-box computation models that only provide limited functional characteristics of the loss function, where explicit function expressions and their gradients are unavailable [19, 20]. In attacking black-box machine learning models, only the function values (e.g., prediction results) are provided [21]. Moreover, in some optimization tasks, acquiring the gradient information is difficult due to its complex form, e.g., involving high dimensional matrix inversion in problems of experimental design [22]. These facts motivate us to design gradient-free (namely, zeroth-order) optimization strategies.

Recently, zeroth-order optimization has received great attention by approximating the full gradient via a random gradient estimate [18, 23–26]. In our previous work [27], we derive ZO-Online ADMM by using such a random gradient estimate. In this work, we consider the problem of distributed estimation over networks with online streaming data, under a general scenario in which an agent only has access to the instantaneous cost function value. We apply zeroth-order gradient estimator to the adaptation step of the diffusion-based strategies, and propose the adapt-then-combine (ATC) ZO-diffusion adaption algorithm. We also provide the mean and mean-square stability conditions of the ATC ZO-diffusion algorithm. Finally, simulations are performed to examine convergence properties of the algorithm, and to compare it with its non-cooperative counterpart and the stochastic gradient counterpart.

Notation. All vectors are column vectors denoted by boldface small letters $\boldsymbol{x}$, and boldface capital letters $\boldsymbol{X}$ denote matrices. The superscript $(\cdot)^\top$ represents the transpose of a matrix or a vector. Mathematical expectation is denoted by $\mathbb{E}\{\cdot\}$. Identity matrix of size $N \times N$ is denoted by $\boldsymbol{I}_N$.We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself. The operator $\mathrm{col}\{\cdot\}$ stacks its vector arguments on the top of each other to generate a connected vector. The operator $\mathrm{bdiag}\{\cdots\}$ forms a block diagonal matrix with its arguments. The other symbols will be defined in the context where they are used.

2. DIFFUSION ADAPTATION FOR DISTRIBUTED ESTIMATION

2.1. Modeling assumptions

We consider a connected network composed of N nodes. The problem is to estimate, in a collaborative and distributed manner, an $M \times 1$ column vector $\boldsymbol{w}^\star$ that minimizes a global cost of the form

$$\mathcal{J}^{\mathrm{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} \mathcal{J}_k(\boldsymbol{w}) \tag{1}$$

with $\mathcal{J}_k(\boldsymbol{w})$ denoting a real-valued function accessible to node k that is assumed to be differentiable and strictly convex. In this work, we focus on the important case where all $\mathcal{J}_k(\boldsymbol{w})$ have the same minimizer $\boldsymbol{w}^\star$. This setting is referred to as single-task problems where nodes in a network need to work cooperatively to attain a common

The work of J. Chen was supported in part by NSFC grant 61671382.

object [28], while associating different minimizers to each node is a more generalized setting that could be studied via estimation over multitask networks [16].

2.2. Diffusion adaptation for distributed estimation

Among the several existing strategies for achieving the minimizer of (1), diffusion adaptation is an efficient approach that is particularly attractive due to its enhanced adaptation and performance and wider stability ranges. Diffusion strategies can be subdivided into two forms: the adapt-then-combine (ATC) and the combine-then-adapt (CTA) strategies. Let non-negative coefficients $c_{\ell k}$ and $a_{\ell k}$ be the (ℓ, k)-th entries of a right-stochastic matrix $\boldsymbol{C}$ and a left-stochastic matrices $\boldsymbol{A}$ such that

$$\boldsymbol{C}\mathbf{1}_N = \mathbf{1}_N,\ \boldsymbol{A}^\top \mathbf{1}_N = \mathbf{1}_N \tag{2}$$

and

$$c_{\ell k} = 0,\ a_{\ell k} = 0 \quad \text{if} \quad \ell \notin \mathcal{N}_k. \tag{3}$$

There are several ways to select the coefficients $c_{\ell k}$ and $a_{\ell k}$ such as using the averaging rule or the Metropolis rule. The ATC diffusion strategy is given by iterating the following two steps:

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \nabla_{\boldsymbol{w}} \mathcal{J}_\ell(\boldsymbol{w}_{k,n-1}) \tag{4}$$

$$\boldsymbol{w}_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,n} \tag{5}$$

on each node k. The parameter μ_k in (4) denotes the positive step size on node k. In this setting, evaluating the gradient of J_ℓ at $\boldsymbol{w}_{k,n-1}$ requires raw data exchange among agents. A simplified setting with $\boldsymbol{C} = \boldsymbol{I}_N$ leads to the ATC diffusion strategy without raw data exchange:

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} - \mu_k \nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}_{k,n-1}) \tag{6}$$

$$\boldsymbol{w}_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,n} \tag{7}$$

Note that the diffusion adaptation algorithms (4) and (5), (6) and (7) require the evaluation of the gradient of $\mathcal{J}_k$.

3. ZEROTH-ORDER DIFFUSION ADAPTATION

Using the gradient of $\mathcal{J}_k$, either in the exact form or in an stochastic form, relies on the fact that the analytical expression of $\mathcal{J}_k$ is available and the associated gradient can be calculated with affordable complexity. These assumptions can be violated in some scenarios, and this motivates us to propose the zeroth-order diffusion strategy via randomized gradient estimator.

3.1. Randomized gradient estimator

A randomized gradient estimator has been used to estimate the gradient of a smooth cost function in many types of zeroth-order optimization algorithms [23–26]. In a similar manner, we consider to apply such a strategy to the diffusion adaptation. This strategy replaces the gradient $\boldsymbol{g}$ of a function f defined on $\mathbb{R}^M$ with a randomized gradient estimate involving two function evaluations:

$$\hat{\boldsymbol{g}}_f(\boldsymbol{y}; \boldsymbol{z}, \varepsilon) = \frac{f(\boldsymbol{y} + \varepsilon \boldsymbol{z}) - f(\boldsymbol{y})}{\varepsilon} \boldsymbol{z} \tag{8}$$

where $\boldsymbol{z} \in \mathbb{R}^M$ is a random vector drawn from a distribution $\boldsymbol{z} \sim \mathcal{D}$ with $\mathbb{E}_{\mathcal{D}}\{\boldsymbol{z}\boldsymbol{z}^\top\} = \boldsymbol{I}_M$, and ε is a small positive smoothing constant. The rationale behind the estimator (8) is that $\hat{\boldsymbol{g}}$ is an unbiased estimator of the directional derivative when the smoothing parameter ε is taken close to zero [23].

3.2. Zeroth-order diffusion adaptation

We now propose zeroth-order diffusion adaptation strategy. The major way to achieve the zeroth-order diffusion adaptation is to use the zeroth-order gradient estimator as introduced in (8).

We assume that at instant n, node k has only access to the instantaneous value of the cost function $\mathcal{J}_k$. This instantaneous function is denoted by $J_{k,n}(\boldsymbol{w})$ and parameterized by a random variable $\boldsymbol{x}_{k,n}$,

$$J_{k,n}(\boldsymbol{w}) = \mathcal{J}_k(\boldsymbol{w}; \boldsymbol{x}_{k,n}) \tag{9}$$

An example of this setting is the stochastic cost function with

$$\mathcal{J}_k(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}}\{J(\boldsymbol{w}; \boldsymbol{x}_{k,n})\} \tag{10}$$

where $\mathbb{E}_{\boldsymbol{x}}\{\cdot\}$ takes the expected value of its argument with respect to the random variable $\boldsymbol{x}$.

Following [23,29], we assume the following conditions for problem (1):

- **A.1:** $J_{k,n}$ is strictly convex, twice differentiable and Lipschitz continuous with $\mathbb{E}\{\|\nabla J_{k,n}(\boldsymbol{w})\|^2\} \leq c_1 < \infty$ for all k.
- **A.2:** The gradient of $J_{k,n}$ is Lipschitz continous with parameter L_k, equivalently $\nabla^2 J_{k,n} \leq L_k \boldsymbol{I}_M$.
- **A.3:** The random vector in (8), for $\boldsymbol{z} \sim \mathcal{D}$ on $\mathbb{R}^M$, the quantity $M(\mathcal{D}) = \sqrt{\mathbb{E}\{\|\boldsymbol{z}\|^6\}}$ is finite, and there is a function $s : \mathbb{N} \to \mathbb{R}^+$ such that $\mathbb{E}\{\|\langle \boldsymbol{a}, \boldsymbol{z}\rangle \boldsymbol{z}\|^2\} \leq s(M)\|\boldsymbol{a}\|^2$ for all $\boldsymbol{a} \in \mathbb{R}^M$.

Now applying the random gradient estimator (8) to the prototypes of diffusion adaptation steps (4) and (5), and considering that we only have access to instantaneous values of the functions $\mathcal{J}_k$, yielding the zeroth-order diffusion adaptation

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, \hat{\boldsymbol{g}}_{J_{\ell,n}}(\boldsymbol{w}_{k,n-1}; \boldsymbol{z}_{\ell,n}, \varepsilon_{\ell,n}) \tag{11}$$

$$\boldsymbol{w}_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,n} \tag{12}$$

If the gradient estimators from the neighbors are not considered, i.e., by setting $\boldsymbol{C} = \boldsymbol{I}$, we have the following algorithm:

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} - \mu_k\, \hat{\boldsymbol{g}}_{J_{k,n}}(\boldsymbol{w}_{k,n-1}; \boldsymbol{z}_{k,n}, \varepsilon_{k,n}) \tag{13}$$

$$\boldsymbol{w}_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,n} \tag{14}$$

3.3. Minibatch strategy

The use of zeroth-order gradient estimates makes the convergence rate dependent on the dimension of optimization variables [23, 27]. In order to improve the convergence property of ZO-diffusion adaptation, we propose to use the minibatch strategy motivated by first-order online learning algorithms [29–31]. Instead of using a single sample as in (8), the average of q random sub-samples $\{z_{t,i}\}_{i=1}^q$ are used for gradient estimation

$$\hat{\boldsymbol{g}}_f(\boldsymbol{y}; \{\boldsymbol{z}_i\}_{i=1}^q, \varepsilon) = \frac{1}{q} \sum_{i=1}^{q} \frac{f(\boldsymbol{y} + \varepsilon \boldsymbol{z}_i) - f(\boldsymbol{y})}{\varepsilon} \boldsymbol{z}_i. \tag{15}$$

We will empirically show that the convergence property of ZO-diffusion adaptation can be largely improved as the minibatch size q increases.

4. STABILITY ANALYSIS

In this section, we will examine the stability conditions of the ZO-diffusion adaptation. Due to the space limitation, we focus on the case without exchanging gradient estimators, namely, using iterations defined in (13) and (14). Our analysis can be extended to iterations (11) and (12) with minor changes. We denote the difference between the optimum $\boldsymbol{w}_k^\star$ and the instantaneous estimate $\boldsymbol{w}_{k,n}$ and intermediate estimate $\boldsymbol{\psi}_{k,n}$ respectively by:

$$\widetilde{\boldsymbol{w}}_{k,n} = \boldsymbol{w}_k^\star - \boldsymbol{w}_{k,n} \tag{16}$$

$$\widetilde{\boldsymbol{\psi}}_{k,n} = \boldsymbol{w}_k^\star - \boldsymbol{\psi}_{k,n} \tag{17}$$

We collect information from across the network into block vectors and matrices. Let us denote by $\widetilde{\boldsymbol{w}}_n$ the block weight error vector at instant n of size $MN \times 1$, that is

$$\widetilde{\boldsymbol{w}}_n = \mathrm{col}\{\widetilde{\boldsymbol{w}}_{1,n}, \ldots, \widetilde{\boldsymbol{w}}_{N,n}\} \tag{18}$$

Subtracting the optimum $\boldsymbol{w}_k^\star$ from both sides of (11), the error update data relation is given by

$$\widetilde{\boldsymbol{\psi}}_{k,n} = \widetilde{\boldsymbol{w}}_{k,n-1} + \mu_k\, \hat{\boldsymbol{g}}_{k,n}(\boldsymbol{w}_{k,n-1}) \tag{19}$$

where we use $\hat{\boldsymbol{g}}_{k,n}(\cdot) = \hat{\boldsymbol{g}}_{J_{k,n}}(\cdot; \boldsymbol{z}_{k,n}, \varepsilon_{k,n})$ for short. We then write the gradient estimator $\hat{\boldsymbol{g}}_{J_{k,n}}(\cdot)$ at $\boldsymbol{w}'$ in form of

$$\hat{\boldsymbol{g}}_{k,n}(\boldsymbol{w}') = \nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}') + \boldsymbol{v}_{k,n}(\boldsymbol{w}') \tag{20}$$

namely, we model the inaccuracy in a gradient vector with some gradient noise component $\boldsymbol{v}_{k,n}(\cdot)$.

Lemma 1 *Given $\boldsymbol{w}'$, $\varepsilon_{k,n}$ and $\boldsymbol{z}_{k,n} \sim \mathcal{D}$:*

$$\mathbb{E}_{\boldsymbol{z}}\{\hat{\boldsymbol{g}}_{k,n}(\boldsymbol{w}')\} = \nabla_{\boldsymbol{w}} \mathcal{J}_\ell(\boldsymbol{w}') + \varepsilon_{k,n} L_k \nu(\boldsymbol{w}', \varepsilon_{k,n}) \tag{21}$$

$$\mathbb{E}_{\boldsymbol{z}}\{\|\hat{\boldsymbol{g}}_{k,n}(\boldsymbol{w}')\|^2\} \le 2s(M)\|\nabla_{\boldsymbol{w}} \mathcal{J}_\ell(\boldsymbol{w}')\|^2 + \frac{1}{2}\varepsilon_{k,n}^2 L_k^2 M(\mathcal{D})^2 \tag{22}$$

with $\|\nu(\boldsymbol{w}', \beta_n)\| \le \frac{1}{2}\mathbb{E}_{\boldsymbol{z}}\{\|\boldsymbol{z}\|^3\}$.

The above results are directly obtained from [23, Lemma 1]. We see from Lemma 1 that the zeroth-order estimator of the gradient is biased. However, when the smoothing parameter $\varepsilon_{\ell,n}$ is small enough, $\hat{\boldsymbol{g}}_{k,n}(\boldsymbol{w}')$ becomes an unbiased estimator of $\nabla_{\boldsymbol{w}} \mathcal{J}_\ell(\boldsymbol{w}')$. Decomposition (20) and Lemma 1 directly lead us to the fact that the norm of the gradient is bounded, i.e.,

$$\mathbb{E}\{\|\boldsymbol{v}_{k,n}(\boldsymbol{w}')\|\}^2 \le \mathbb{E}\{\|\boldsymbol{v}_{k,n}(\boldsymbol{w}')\|^2\} \le \tau \tag{23}$$

with τ being a constant dependent of $s(M)$, c_1, $M(\mathcal{D})$. Furthermore, for the twice-differentiable function, we have

$$\nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}_{k,n-1}) \tag{24}$$

$$= \nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}_k^\star) - \Big[\int_0^1 \nabla_{\boldsymbol{w}}^2 \mathcal{J}_k(\boldsymbol{w}_k^\star - t\widetilde{\boldsymbol{w}}_{k,n-1})dt\Big]\widetilde{\boldsymbol{w}}_{k,n-1} \tag{25}$$

$$\triangleq -\boldsymbol{H}_{k,n-1}\widetilde{\boldsymbol{w}}_{k,n-1} \tag{26}$$

by noticing the fact $\nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}_k^\star) = 0$ and introducing the symmetric random matrix $\boldsymbol{H}_{k,n-1} = \int_0^1 \nabla_{\boldsymbol{w}}^2 \mathcal{J}_k(\boldsymbol{w}^\star - t\widetilde{\boldsymbol{w}}_{k,n-1})dt$. Under A.2, it is easy to verify that [8]

$$\boldsymbol{H}_{k,n-1} \le L_k \boldsymbol{I}_M \tag{27}$$

4.1. Mean stability analysis

With the expression of $\nabla_{\boldsymbol{w}} \mathcal{J}_k(\boldsymbol{w}_{k,n-1})$, weight error recursion (19) can then be written with $\boldsymbol{H}_{k,n-1}$ by:

$$\widetilde{\boldsymbol{\psi}}_{k,n} = (\boldsymbol{I}_L - \mu_k \boldsymbol{H}_{k,n-1})\widetilde{\boldsymbol{w}}_{k,n-1} + \boldsymbol{v}_{k,n}(\boldsymbol{w}_{k,n-1}) \tag{28}$$

Defining the matrix $\boldsymbol{\mathcal{A}} = \boldsymbol{A} \otimes \boldsymbol{I}_M$, the network weight error vector $\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\}$ evolves according to

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\} = \boldsymbol{B}\mathbb{E}\{\widetilde{\boldsymbol{w}}_{n-1}\} + \mathbb{E}\{\boldsymbol{v}_{n-1}\} \tag{29}$$

where we define

$$\boldsymbol{B}_n = \boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}_{NM} - \mu_k \mathrm{bdiag}\{\boldsymbol{H}_{1,n-1}, \ldots, \boldsymbol{H}_{N,n-1}\}) \tag{30}$$

and $\boldsymbol{B} = \mathbb{E}\{\boldsymbol{B}_n\}$. From (23) we know that $\mathbb{E}\{\boldsymbol{v}_{n-1}\}$ is bounded, thus the stability of (29) is governed by the stability of $\boldsymbol{B}$, namely,

$$\rho\big(\boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}_{NM} - \mu_k \mathrm{bdiag}\{\mathbb{E}\{\boldsymbol{H}_{1,n-1}\}, \ldots, \mathbb{E}\{\boldsymbol{H}_{N,n-1}\}\}\big) < 1. \tag{31}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix. Considering properties of $\boldsymbol{\mathcal{A}}^\top$ [6] and (27), this requirement leads to the condition:

$$\mu_k < \frac{2}{L_k}. \tag{32}$$

4.2. Mean-square stability analysis

We then perform the mean-square stability analysis relying on a set of inequality recursions that will enable us to bound the steady-state mean-square-error. Equating the squared Euclidean norms of both sides of (28), applying the expectation operator, we have

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{\psi}}_{k,n}\|^2\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}\|^2_{\boldsymbol{B}^2_{k,n-1}}\} + \mu_k^2\, \mathbb{E}\{\big\|\boldsymbol{v}_{k,n}(\widetilde{\boldsymbol{w}}_{k,n-1})\big\|^2\} \\ + \mu_k \mathbb{E}\{\widetilde{\boldsymbol{w}}_{k,n-1}^\top \boldsymbol{B}_{k,n-1}\boldsymbol{v}_{k,n}(\widetilde{\boldsymbol{w}}_{k,n-1})\}\end{aligned} \tag{33}$$

with $\boldsymbol{B}_{k,n-1} = \boldsymbol{I}_M - \mu_k \boldsymbol{H}_{k,n-1}$. Using Cauchy-Schwarz inequality and the bound in (23), we have an upper bound of $\mathbb{E}\{\|\widetilde{\boldsymbol{\psi}}_{k,n}\|^2\}$

$$\begin{aligned}&\mathbb{E}\{\|\widetilde{\boldsymbol{\psi}}_{k,n}\|^2\} \\ &\le \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}\|^2_{\boldsymbol{B}^2_{k,n-1}}\} + \mu_k^2\tau + \mu_k\sqrt{\tau}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}^\top \boldsymbol{B}_{k,n-1}\|\} \\ &\le \gamma_k^2 \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}\|^2\} + \mu_k^2\tau + \mu_k\gamma_k\sqrt{\tau}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}\|\}\end{aligned} \tag{34}$$

where $\gamma_k = |1 - \mu_k L_k|$, note that $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n-1}\|\}$ is bounded with the mean stability condition (32). Subtract $\boldsymbol{w}_k^\star$ from both side of (14), and note that $\boldsymbol{A}$ is a left-stochastic matrix, the weight error vectors $\widetilde{\boldsymbol{w}}_{k,n}$ and $\widetilde{\boldsymbol{\psi}}_{k,n}$ are related by $\widetilde{\boldsymbol{w}}_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \widetilde{\boldsymbol{\psi}}_{\ell,n}$. Further considering that $\|\cdot\|^2$ is a convex function, by Jensen's inequality and taking expectation we obtain

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n}\|^2\} \le \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \mathbb{E}\{\|\widetilde{\boldsymbol{\psi}}_{\ell,n}\|^2\}. \tag{35}$$

We then introduce the following network mean-square-error vectors:

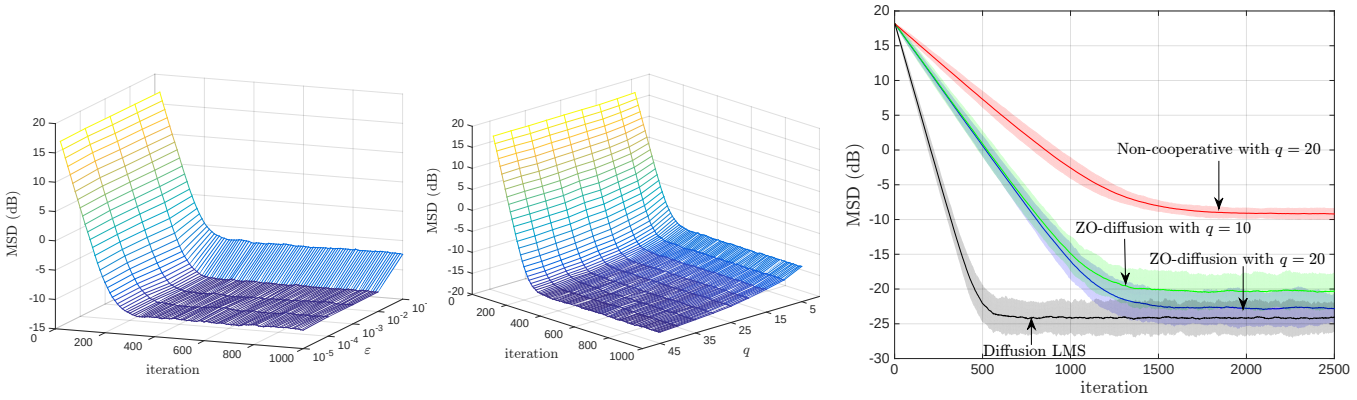
$$\boldsymbol{\zeta}_n = \big(\mathbb{E}\|\widetilde{\boldsymbol{w}}_{1,n}\|^2, \ldots, \mathbb{E}\|\widetilde{\boldsymbol{w}}_{N,n}\|^2\big)^\top \tag{36}$$

Combining all three inequalities leads to

$$\boldsymbol{\zeta}_n = \boldsymbol{A}^\top \mathrm{diag}\{\gamma_1^2, \ldots, \gamma_N^2\}\boldsymbol{\zeta}_{n-1} + \boldsymbol{A}^\top \boldsymbol{p}_{n-1} \tag{37}$$

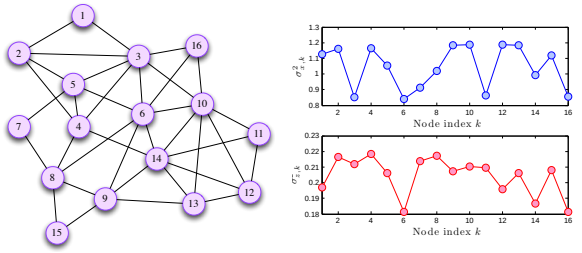
with $\boldsymbol{p}_{n-1}$ denoting the bounded vector whose entries are given by the last two terms in (34). For such a recursion with a bounded driven term, the stability is governed by $\gamma_k < 1$, which also leads to

$$\mu_k < \frac{2}{L_k}. \tag{38}$$



(a) MSD learning curves with varied ε. (b) MSD learning curves with varied q. (c) Compared MSD learning curves. Shading regions represent three standard deviations of MSD fluctuations.

Fig. 1. MSD learning curve properties and comparisons.



(a) Network topology (b) Input variances (top) and noise variances (bottom)

Fig. 2. Network topology, input variances, and noise variances.

5. SIMULATIONS

We now report simulation results to illustrate the properties of the proposed ZO-diffusion adaptation algorithm. All nodes were initialized with zero parameter vectors $\boldsymbol{w}_{k,0} = \boldsymbol{0}$. Simulation curves were obtained by averaging over 100 trials.

Consider the network of 16 agents depicted in Fig. 2(a). We suppose agent k is associated with the local cost $\mathcal{J}_k(\boldsymbol{w})$ in form of

$$\mathcal{J}_k(\boldsymbol{w}) = \mathbb{E}\{(d_{k,n} - \boldsymbol{w}^\top \boldsymbol{x}_{k,n})^2\} \tag{39}$$

where $\{\boldsymbol{x}_{k,n}\}$ are a random vectors of length M, and $d_{k,n} = \boldsymbol{x}_{k,n}^\top \boldsymbol{w}^\star + z_{k,n}$ where $z_{k,n}$ are a zero-mean Gaussian noise with variance σ_z^2, independent of any other signals. It is clear that (39) is the mean-square error criterion with regressor $\boldsymbol{x}_{k,n}$ and dependent variable $d_{k,n}$. We use (39) as an illustrative example by considering that we only have access to instantaneous function value $J_{k,n} = (d_{k,n} - \boldsymbol{w}^\top \boldsymbol{x}_{k,n})^2$, without knowledge of $\{\boldsymbol{w}_{k,n}, d_{k,n}\}$. To perform simulations, the system order is set to $M = 50$. The regression inputs $\boldsymbol{x}_{k,n}$ are zero-mean $M \times 1$ random vectors governed by a Gaussian distribution with covariance matrices $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2 \boldsymbol{I}_M$. We set the unknown variable vector $\boldsymbol{w}^\star$ to be a fixed set of variables sampled from $\mathcal{N}(0,1)$. For the combination step, we use a uniform combination matrix $\boldsymbol{A}$ such that $a_{\ell k} = \frac{1}{|\mathcal{N}_k|}$, and set $\boldsymbol{C} = \boldsymbol{I}_N$ for simplicity. For generating random gradient estimators, the random vector $\boldsymbol{z}$ is sampled from $\mathcal{N}(0, \boldsymbol{I}_M)$.

In order to examine the effects of parameters in the ZO-diffusion adaptation algorithm, we vary the gradient estimate parameter ε and minibatch parameter q. We first test ε with values $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ with fixed $q = 10$, and then text q with values 5 to 50 with increments of 5, using fixed $\varepsilon = 10^{-2}$. All these tests are with a fixed step size $\mu_k = 0.01$ for all nodes. Fig. 1(a) shows that the mean-square deviation (MSD) convergence behavior with varied ε. It is observed that MSD learning curves have almost the same convergence rates and steady-state performance with all these values, except for a large value $\varepsilon = 0.1$. This result indicates that the convergence property is not significantly varied with a reasonably small ε. Fig. 1(b) shows that ZO-diffusion adaptation achieves a better steady state performance with an increased minibatch size q, since a larger q yields a better estimate of the gradient.

We then compare the ZO-diffusion adaptation strategy with its non-cooperative counterpart, and also with the diffusion LMS algorithm. In this simulation, we set $q = 20$ and $\varepsilon = 0.01$ for zeroth-order strategies. We set step size $\mu_k = 0.002$ for all ZO-type algorithms, and $\mu_k = 0.01$ for diffusion LMS. MSD learning curves of these algorithms and settings are shown in Fig. 1(c). The cooperative ZO-diffusion adaptation has significant advantage over its non-cooperative counterpart in convergence property. Diffusion LMS exhibits a better performance while it requires the knowledge of analytical form of the cost.

6. CONCLUSION AND PERSPECTIVE

In this paper, we considered a network operating with applications where the analytical forms of cost functions are not accessible. In order to perform distributed estimation, we introduced the zeroth-order gradient estimator into the diffusion-based adaptation strategy, and proposed a combine-then-adapt ZO-diffusion algorithm. Stability conditions were established for the algorithm. Simulations validated the proposed strategy. Our future work will include analyzing the convergence rate of ZO-diffusion algorithm, and considering costs with extra optimization constraints.

7. REFERENCES

[1] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3375–3380, Jul. 2008.

[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[3] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[4] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[5] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[6] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Librarary in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., vol. 3, pp. 322–454. Elsevier, 2014.

[7] A. H. Sayed, "Adaptive networks," *Proc. of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[8] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[9] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive netowrks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[10] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[11] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[12] L. Li and J. Chambers, "Distributed adaptive estimation based on the apa algorithm over diffusion networks with changing topology," in *Proc. IEEE SSP*, 2009, pp. 757–760.

[13] F. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[14] J. Hu, L. Xie, and C. Zhang, "Diffusion Kalman filtering based on covariance intersection," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 891–902, 2011.

[15] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[16] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[17] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[18] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. 10th Conference on Learning Theory*, 2010, pp. 28–40.

[19] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *J. Global Optimizaiton*, vol. 56, no. 3, 2013.

[20] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*, vol. 6, John Wiley & Sons, 2005.

[21] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," *arXiv preprint arXiv:1708.03999*, 2017.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[23] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Information Theory*, vol. 61, no. 6, 2015.

[24] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optimiz.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[25] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Fundations of Computational Mathematics*, vol. 2, no. 17, pp. 527–566, 2015.

[26] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two- point feedback," *J. Mach. Learn. Res.*, vol. 18, no. 52, pp. 1–11, 2017.

[27] S. Liu, J. Chen, P.-Y. Chen, and A. O. Hero, "Zeroth-order online alternating direction method of multipliers," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), also available at arXiv:1710.07804*, Playa Blanca, Canary Islands, 2018.

[28] S. Liu, P.-Y. Chen, and A. O. Hero, "Accelerated distributed dual averaging over evolving networks of growing connectivity," *arXiv*, https://arxiv.org/abs/1704.05193, 2017.

[29] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. Int. Conf. Machine Learn.*, 2013, pp. 392–400.

[30] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Effcient mini-batch training for stochastic optimization," in *Proc. ACM SIKDD*, 2014, pp. 661–670.

[31] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan, "Better mini-batch algorithms via accelerated gradient methods," *Adv. Neural Inf. Process. Syst.*, pp. 1647–1655, 2011.