2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

A Dual Accelerated Method for a Class of Distributed Optimization Problems: From Consensus to Decentralized Policy Evaluation

Sheng Zhang, Ashwin Pananjady and Justin Romberg

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993276

***Abstract*— Motivated by decentralized sensing and policy evaluation problems, we consider a particular type of distributed stochastic optimization problem over a network, called the online stochastic distributed averaging problem. We design a dual-based method for this distributed consensus problem with Polyak–Ruppert averaging and analyze its behavior. We show that the proposed algorithm attains an accelerated deterministic error depending optimally on the condition number of the network, and also that it has an order-optimal stochastic error. This improves on the guarantees of state-of-the-art distributed stochastic optimization algorithms when specialized to this setting, and yields—among other things—corollaries for decentralized policy evaluation. Our proofs rely on explicitly studying the evolution of several relevant linear systems, and may be of independent interest.**

I. Introduction

Consider an online, stochastic distributed averaging problem in which noisy data becomes available sequentially to agents residing on a network. More precisely, we consider a network (undirected connected graph) $\mathscr{G} = (\mathscr{N}, \mathscr{E})$ consisting of a set of nodes $\mathscr{N} = \{1, \ldots, N\}$ and a set of edges $\mathscr{E} \subseteq \mathscr{N} \times \mathscr{N}$, where each edge $(i, j) \in \mathscr{E}$ is an unordered pair of distinct nodes. The set of neighbors of node $i \in \mathscr{N}$ is denoted by $\mathscr{N}_i = \{j \in \mathscr{N} \mid (i, j) \in \mathscr{E}\}$. At every time step $t = 0, 1, 2, \ldots$, each node $i \in \mathscr{N}$ receives a local random vector $R_t^i \in \mathbb{R}^n$, with mean vector $\mathbb{E}\left[R_t^i\right] = \mu_i \in \mathbb{R}^n$ and covariance matrix $\mathrm{Cov}\left[R_t^i\right] = \Sigma_i \in \mathbb{R}^{n \times n}$. Of particular interest will be the individual variances $\sigma_{i,1}^2, \ldots, \sigma_{i,n}^2$ of the n components of R_t^i. We assume that the local random vectors are generated independently across time and nodes. The goal is to iteratively estimate the average of the mean vectors $\bar{\mu} := \frac{1}{N}\sum_{i=1}^N \mu_i$ at every node, via a distributed algorithm in which the nodes can only communicate with their neighbors. Such a setting is motivated by the following two particular examples:

a) Distributed linear parameter estimation [2], [3]: Here, we want to estimate a parameter vector $\beta^* \in \mathbb{R}^d$ using observations from a network of N distributed sensors modeled by a graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$. At time $t \geq 0$, sensor $i \in \mathscr{N}$ makes an m_i-dimensional noisy measurement

$$Y_t^i = A_i \beta^* + \varepsilon_t^i,$$

where A_i is an $m_i \times d$ matrix known only to sensor i, and $\varepsilon_t^i \in \mathbb{R}^{m_i}$ is a zero-mean noise vector, that is independent across sensors and time, with covariance matrix $\Sigma_\varepsilon^i \in \mathbb{R}^{m_i \times m_i}$.

The full version [1] of this paper can be found at `https://arxiv.org/pdf/2207.11425.pdf`.

Sheng Zhang, Ashwin Pananjady and Justin Romberg are with Georgia Institute of Technology, Atlanta, GA 30332, USA, `shengzhang, ashwinpm@gatech.edu, jrom@ece.gatech.edu`.

We work in the fully decentralized setting where each sensor can only exchange data with its neighbors; there is no central fusion center. The goal is for each sensor to have an estimate of the unknown parameter β^*. To make the problem well-posed when $m_i < d$ for all $i \in \mathscr{N}$, we will assume that the matrix $\sum_{i=1}^N A_i{}^\top A_i$ is invertible (i.e. the "distributed invertibilty" condition from [4]).

With $\mu_i := \mathbb{E}\left[A_i^\top Y_t^i\right] = A_i^\top A_i \beta^*$, we have

$$\beta^* = \left(\frac{1}{N}\sum_{i=1}^N A_i{}^\top A_i\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^N \mu_i\right).$$

Thus, each sensor can form an estimate of β^* by estimating the global averages $\bar{A} := \frac{1}{N}\sum_{i=1}^N A_i^\top A_i$ and $\bar{\mu} := \frac{1}{N}\sum_{i=1}^N \mu_i$. The problem of estimating $\bar{A}$ and $\bar{\mu}$ are covered by our setting. We will present finite-time bounds on how accurately β^* can be approximated given each sensor's estimates of $\bar{A}$ and $\bar{\mu}$.

b) Decentralized multi-agent policy evaluation [5], [6]: A central problem in reinforcement learning is to estimate the value function of a given stationary policy in a Markov decision process, often referred to as the policy evaluation problem. Because the policy is given and applied automatically to select actions, such a problem is naturally formulated as value function estimation in a Markov reward process (MRP).

Here, N agents operate in a common environment modeled by a finite MRP consisting of a set of states $\mathscr{S} = \{1, \cdots, n\}$, a state transition matrix $P \in [0,1]^{n \times n}$, rewards $r_i \in \mathbb{R}^n$ for agent $i \in \mathscr{N}$ being in each state, and a discount factor $\gamma \in [0, 1)$. We again work in the decentralized setting where agents can only communicate with their neighbors in a network $\mathscr{G} = (\mathscr{N}, \mathscr{E})$. Their goal is to cooperatively estimate the value function $J^* : \mathscr{S} \to \mathbb{R}$ defined for all $s \in \mathscr{S}$ as

$$J^*(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \bar{r}_{s_t} | s_0 = s\right],$$

where $s_{t+1} \sim P(s_t, \cdot)$ for all $t \geq 0$ and $\bar{r}_j$ is the j-th component of $\bar{r} := \frac{1}{N}\sum_{i=1}^N r_i$. It is known that J^* solves the Bellman equation

$$J = \bar{r} + \gamma P J \tag{1}$$

meaning $\quad J^* = (I - \gamma P)^{-1} \bar{r}$.

In the learning setting, P and $r_1, \ldots, r_N$ are unknown, and we instead assume access to a black box simulator. This observation model is often referred to as the generative model [7]: for each time step $t \geq 0$ and for each state $j \in \mathscr{S}$, each

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.

agent $i \in \mathcal{N}$ observes a common random next state $X_{t,j} \sim P(j,\cdot)$, and a local random reward $R^i_{t,j}$ with mean $r_{i,j}$ and variance $\sigma^2_{i,j}$. We assume that the local random rewards are generated independently across time and agents.

A natural approach for solving this problem is to use the samples collected to construct estimates $\left(\widehat{P},\widehat{r}\right)$ of the pair $(P,\bar{r})$, and then plug these estimates into the Bellman equation [8]. The problem of estimating $\bar{r}$ in a decentralized manner is covered by our framework, and we will provide finite-sample bounds on how precisely J^* can be approximated given each agent's estimates of the pair $(P,\bar{r})$.

A. Related work

There has been much recent interest in developing distributed algorithms for applications in robotics [9], power system control [10] and multi-agent reinforcement learning [5]. This is motivated mainly by the emergence of large-scale networks, where large amount of data are involved, and generation and processing of information is not centralized. Notable among these are those algorithms that can be used by a group agents to reach a consensus in a distributed manner. The distributed consensus problem has been studied extensively in the computer science literature [11] and has found a number of applications including coordination of UAVs [12], information processing in sensor networks [13], and distributed optimization [14].

The distributed averaging problem is a special case in which the goal is to compute the exact average of the initial values of the agents via a distributed algorithm. The most common distributed averaging algorithms are linear and iterative, which can be classified as deterministic or randomized. Several well-known deterministic distributed averaging algorithms were proposed and analyzed in [15], [16], [17], [18], [19], where at each time step, every agent takes a weighted average of its own value with values received from some of the other agents. For other deterministic algorithms, we refer the reader to [19] and the references therein. There are also two popular randomized algorithms, where at each time step, either two randomly selected nodes interchange information [20], or a randomly selected node broadcasts its value to all its neighbors [21]. For a discussion of other randomized algorithms, we refer the reader to [22] and the references therein.

Many existing algorithms for distributed averaging require that agents are able to receive precise measurement values. However, constrained by limited sensing, agents might only be able to observe noisy measurements. Moreover, modern distributed systems involve a large amount of data available in a sequential order. As each agent is subject to computation and storage constraints, it needs to process and distribute information received in an online fashion. Motivated by these considerations, in this paper, we study the natural online stochastic distributed averaging problem described above.

Our framework can be viewed as a special case of distributed stochastic optimization. The goal of distributed optimization is to minimize a global objective function given as a sum of local objective functions held by each agent, in a distributed manner. The distributed optimization problem has been studied for a long time and can be traced back to the seminal works [15], [16] in the context of parallel and distributed computation. It has gained growing renewed interest over the last decade due to its various applications in power systems [23], communication networks [24], machine learning [25], and wireless sensor networks [26]. Recent reviews can be found in the surveys [27], [28] and the books [29], [30].

Distributed deterministic optimization is quite well understood with many centralized algorithms having their decentralized counterparts. For example, there exist distributed subgradient methods [31], gradient methods [32], and many variants of accelerated gradient methods [33], [34], [35], which achieve both communication and oracle complexity lower bounds.

Optimal methods using a primal approach for smooth and strongly convex distributed stochastic optimization over networks were recently proposed and analyzed by [36] and [37]. There are also methods using dual approach [38], [39], which are akin to the methods we develop and analyze in this paper (for the special class of quadratic functions). Sections II and III provide detailed discussions of similarities and differences between our results and this body of work. In short, the dual approaches [38], [39] achieve optimal communication complexity in the general distributed optimization setting but fall short in terms of their oracle complexity.

B. Contributions

In this paper, we follow the dual approach of [33] and propose a stochastic dual accelerated method using constant step-size and Polyak–Ruppert averaging for the online stochastic distributed averaging problem. We establish non-asymptotic convergence guarantees with explicit dependence on the network connectivity parameter and noise in the observations. Our analysis builds on a discrete-time dynamical system representation of the algorithm and relies on explicitly studying the evolution of several relevant linear systems, which may be of independent interest. Our mean-squared error upper bounds provide tight guarantees on the bias and variance term for the algorithm. We show that (i) the bias term decays linearly at an accelerated rate with exponent $\mathcal{O}\left(-\frac{T}{\sqrt{\kappa(L)}}\right)$, where $\kappa(L)$ is the condition number of the network and its precise definition is given in Section II, (ii) the variance term achieves the $\mathcal{O}\left(\frac{\sum_{j=1}^n \max_{i\in\mathcal{N}} \sigma^2_{i,j}}{T}\right)$ rate up to a higher order term in T and (iii) the convergence rate of the algorithm is optimal. Moreover, we show that our method outperforms a state-of-the-art primal accelerated method called D-MASG in a relevant non-asymptotic regime where $T \asymp \sqrt{\kappa(L)}$. Furthermore, when assuming that $\sigma^2_{i,j} = \sigma^2_{i',j}$ for all $i \neq i' \in \mathcal{N}$ and $j \in \{1,\ldots,n\}$, and letting $\sigma^2 := \sum_{j=1}^n \sigma^2_{i,j}$, we show that our method has optimal per-node oracle complexity $\mathcal{O}\left(\frac{\sigma^2}{\varepsilon}\right)$ and optimal communication complexity $\mathcal{O}\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$, where $\varepsilon > 0$

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.

is the desired accuracy. In contrast, when specialized to our setting, state-of-the-art dual accelerated methods [38], namely, R-RRMA-AC-SA^2 and SSTM_sc, have the same communication complexity $\mathscr{O}\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$ as our algorithm, but can only be shown to have much larger per-node oracle complexity $\mathscr{O}\left(\kappa(L)\frac{\sigma^2}{\varepsilon^2}\right)$.

II. Stochastic Dual Accelerated Method

In this section, we first cast distributed averaging as distributed optimization with quadratic local objective functions, which is a well-known correspondence. Then, we follow [33], [40] and use the dual formulation of the distributed optimization problem to design an algorithm that can be executed for the online stochastic distributed averaging problem.

First, notice that the target vector $\bar{\mu} = \frac{1}{N}\sum_{i=1}^N \mu_i$ is the unique optimal solution of the optimization problem

$$\min_{\theta\in\mathbb{R}^n} \frac{1}{2}\sum_{i=1}^N \|\theta-\mu_i\|_2^2, \tag{2}$$

A standard way to solve problem (2) in a decentralized setting is rewriting the problem as

$$\min_{\theta^1=\cdots=\theta^N} \frac{1}{2}\sum_{i=1}^N \|\theta^i-\mu_i\|_2^2. \tag{3}$$

In this paper, consider a dual approach for problem (3), which uses a decentralized communication scheme based on the gossip algorithm [20]. More specifically, during a communication step, each node $i\in\mathscr{N}$ broadcasts an n-dimensional vector to its neighbors and then computes a linear combination of the values received from its neighbors: node i sends θ^i to its neighbors and receives $\sum_{j\in\mathscr{N}_i} L_{i,j}\theta^j$. One round of communication over the network can be represented as multiplying the current estimates with a gossip matrix $L=[L_{i,j}]\in\mathbb{R}^{N\times N}$. In order to encode communication constraints imposed by the network, we impose standard assumptions on L [33], [40]:

1. L is symmetric and positive semi-definite,
2. The kernel (i.e. nullspace) of L is the set of constant vectors,
3. L is defined on the edges of the network: $L_{i,j}\neq 0$ only if $i=j$ or $(i,j)\in\mathscr{E}$.

The second condition will ensure consensus among agents and also allow us to rewrite the consensus agreement constraint $\theta^1=\ldots=\theta^N$ in a convenient way. Note that a simple choice of the gossip matrix—which underlies our choice of notation—is the the graph Laplacian matrix for $\mathscr{G}$, but other choices satisfying the above conditions are also valid.

We will denote by $0=\lambda_N(L)<\lambda_{N-1}(L)\leq\cdots\leq\lambda_1(L)$ the spectrum of L. Let $\kappa(L):=\frac{\lambda_1(L)}{\lambda_{N-1}(L)}$ be the ratio between the largest and the second smallest eigenvalue of L. This quantity is the condition number of L in the space orthogonal to $\ker(L)$, and characterizes the connectivity of the network and how fast the information is spread over the network. Since L is a real symmetric matrix, it can be decomposed as $L=Q\Lambda Q^\top$, where $\Lambda:=\operatorname{diag}(\lambda_1(L),\ldots,\lambda_N(L))$ is a diagonal matrix whose entries are the eigenvalues of L and Q is the orthogonal matrix whose i-th column is the eigenvector of L associated with $\lambda_i(L)$. Such a decomposition is not unique when the eigenvalues are not distinct; in this case, it suffices to choose any valid decomposition.

We observe that the equality constraint $\theta^1=\cdots=\theta^N$ is equivalent to $\left(I_n\otimes\sqrt{L}\right)\Theta=0$, where $\sqrt{L}:=Q\sqrt{\Lambda}Q^\top$, I_n is the $n\times n$ identity matrix, $\otimes$ is the the Kronecker product and $\Theta:=\operatorname{vec}\left(\left[\theta^1\cdots\theta^N\right]^\top\right)\in\mathbb{R}^{Nn}$. Here, we use $\sqrt{L}$ instead of L because we will later square it via the change of variables, and $\operatorname{vec}(A)$ is the vectorization of a matrix A obtained by stacking the columns of the matrix A on top of one another. This observation leads to the following primal problem:

$$\begin{aligned}\min_{\Theta\in\mathbb{R}^{Nn}}\quad & \frac{1}{2}\sum_{i=1}^N\|\theta^i-\mu_i\|_2^2\\ \text{s.t.}\quad & \left(I_n\otimes\sqrt{L}\right)\Theta=0.\end{aligned} \tag{4}$$

The Lagrangian function ℓ associated with problem (4) is given by

$$\begin{aligned}\ell(\Theta,\lambda) &= \frac{1}{2}\sum_{i=1}^N\|\theta^i-\mu_i\|_2^2-\lambda^\top\left[\left(I_n\otimes\sqrt{L}\right)\Theta\right]\\ &= \sum_{i=1}^N\left[\frac{1}{2}\|\theta^i-\mu_i\|_2^2-(x^i)^\top\theta^i\right]\end{aligned} \tag{5}$$

where $\lambda\in\mathbb{R}^{Nn}$ is the Lagrange multiplier vector and $X:=\operatorname{vec}\left(\left[x^1\cdots x^N\right]^\top\right)=\left(I_n\otimes\sqrt{L}\right)\lambda$. Since strong duality holds, the convex program (4) can be equivalently written in its dual form:

$$\begin{aligned}&\max_{\lambda\in\mathbb{R}^{Nn}}\min_{\Theta\in\mathbb{R}^{Nn}}\ell(\Theta,\lambda)\\ &=-\min_{\lambda\in\mathbb{R}^{Nn}}\left\{\frac{1}{2}\lambda^\top(I_n\otimes L)\lambda+\left[\left(I_n\otimes\sqrt{L}\right)\mu\right]^\top\lambda\right\},\end{aligned} \tag{6}$$

where $\mu:=\operatorname{vec}\left(\left[\mu_1\cdots\mu_N\right]^\top\right)$. Note that a gradient step with step-size $\eta>0$ for problem (6) is

$$\lambda_{t+1}=\lambda_t-\eta\left(I_n\otimes\sqrt{L}\right)\left[\left(I_n\otimes\sqrt{L}\right)\lambda_t+\mu\right],$$

and the change of variables $X_t:=\left(I_n\otimes\sqrt{L}\right)\lambda_t$ yields the iteration

$$X_{t+1}=X_t-\eta\left(I_n\otimes L\right)\left(X_t+\mu\right). \tag{8}$$

Since μ is unknown in the online stochastic distributed averaging problem, Eq. (8) is not directly applicable. However, we have access to samples $\{R_t^1,\ldots,R_t^N\}$ at every time step $t\geq 0$. Thus, a natural way to obtain the stochastic version of (8) is to replace μ with its unbiased estimator $\widehat{\mu}_t:=\operatorname{vec}\left(\left[R_t^1\cdots R_t^N\right]^\top\right)$.

While the above calculations provide transparent intuition on which to base algorithm design, our proposed algorithm is the stochastic dual accelerated method (SDA) presented in Algorithm 1, which involves a more sophisticated (but still simple) iteration. In particular, it relies on Nesterov's accelerated gradient method [41] with constant step-size, used in conjunction with Polyak–Ruppert averaging of the

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Stochastic Dual Accelerated Algorithm (SDA)

Input: number of iterations $T > 0$, "burn-in" time $T_0 \in [0, T-1]$, gossip matrix $L \in \mathbb{R}^{N\times N}$, step-size $\eta > 0$ and momentum parameter $\zeta \geq 0$.

Initialization: each agent $i \in \mathcal{N}$ initializes $x_0^i = y_0^i = 0 \in \mathbb{R}^n$.

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

  for agent $i \in \mathcal{N}$ **do**

    observes a local random vector R_t^i and executes the local update:

$$\theta_t^i = x_t^i + R_t^i.$$

    exchanges θ_t^i with each agent $j \in \mathcal{N}_i$ and executes the local updates:

$$y_{t+1}^i = x_t^i - \eta \sum_{j\in\mathcal{N}_i\cup\{i\}} L_{i,j}\theta_t^j,$$
$$x_{t+1}^i = y_{t+1}^i + \zeta\left(y_{t+1}^i - y_t^i\right).$$

  end for

end for

Output: $\widehat{\theta}_T^i := \frac{1}{T-T_0}\sum_{t=T_0}^{T-1}\theta_t^i$ for all $i \in \mathcal{N}$.

last $T - T_0$ iterates [42]. SDA is a stochastic variant of the single-step dual accelerated algorithm proposed and analyzed in [33], which was developed for smooth and strongly convex distributed deterministic optimization. While both algorithms are similar in spirit, the analysis of SDA uses completely different techniques, since it applies to the stochastic setting for a special class of quadratic functions, as opposed to the deterministic setting for general smooth and strongly convex functions. The analysis in this paper builds on a dynamical system representation of the algorithm and relies on explicitly studying the evolution of several relevant linear systems, and may be of independent interest.

III. Main Results

In this section, we begin by stating our theorem regarding the performance of SDA, and discussing some of the consequences of this result. In order to state our theorem, we require the following definition:

Definition 1: k^* is the smallest positive integer such that for all integer $k \geq k^*$:

$$\left(1 + \frac{k}{\sqrt{\kappa(L)}+1}\right)\left(1 - \frac{1}{\sqrt{\kappa(L)}}\right)^k \leq \left(1 - \frac{1}{2\sqrt{\kappa(L)}}\right)^k.$$

Note that k^* is well-defined and there exists an absolute constant $C \geq 1$ such that $k^* \leq C\cdot\sqrt{\kappa(L)}$ (see Lemma 3 and its proof in the full version [1] of this paper). Now, we are ready to present the finite-time performance bound of SDA in the following theorem.

Theorem 1: Consider running SDA with the following parameters: $T_0 = \frac{T}{2} \geq k^*$, $\eta = \frac{1}{\lambda_1(L)}$ and $\zeta = \frac{\sqrt{\kappa(L)}-1}{\sqrt{\kappa(L)}+1}$, where k^* is defined according to Definition 1. Let $\{\widehat{\theta}_T^i\}_{i\in\mathcal{N}}$ be generated by SDA. Then we have

$$\mathbb{E}\left[\sum_{i=1}^N \left\|\widehat{\theta}_T^i - \bar{\mu}\right\|_2^2\right] \leq \frac{16\kappa(L)}{T^2}e^{-\frac{T}{2\sqrt{\kappa(L)}}}\sum_{i=1}^N\|\mu_i-\bar{\mu}\|_2^2 + \frac{24\left(k^*+\sqrt{\kappa(L)}\right)\sum_{i=1}^N\sum_{j=1}^n\sigma_{i,j}^2}{T^2} + \frac{2\sum_{j=1}^n\max_{i\in\mathcal{N}}\sigma_{i,j}^2}{T}. \tag{9}$$

See Section 5.1 in [1] for a proof. It is worth making a few comments on this theorem. To simplify the discussion, we assume that $\sigma_{i,j} = \sigma_{i',j}^2$ for all $i \neq i' \in \mathcal{N}$ and $j = 1,\ldots,n$, and let $\sigma^2 := \sum_{j=1}^n\sigma_{i,j}^2$. Such an assumption is similar to the standard assumption in distributed stochastic optimization that the stochastic first-order oracle has finite variance bounded by σ^2, i.e., the variance of the stochastic (dual) gradient is bounded by σ^2.

Remark 1: Let us interpret the terms appearing in the bound (9). Since $k^* \leq C\cdot\sqrt{\kappa(L)}$ for some absolute constant $C \geq 1$, the upper bound (9) simplifies to

$$\mathbb{E}\left[\sum_{i=1}^N\left\|\widehat{\theta}_T^i - \bar{\mu}\right\|_2^2\right] \leq \underbrace{\mathcal{O}\left(e^{-\frac{T}{\sqrt{\kappa(L)}}}\sum_{i=1}^N\|\mu_i-\bar{\mu}\|_2^2\right)}_{\text{"bias"}} + \underbrace{\mathcal{O}\left(\frac{N\sqrt{\kappa(L)}\sigma^2}{T^2} + \frac{\sigma^2}{T}\right)}_{\text{"variance"}}. \tag{10}$$

This bound is presented in terms of two components: a bias term which is deterministic and independent of the noise level, and a variance term that measures the effect of noise on the algorithm. Note that our proposed method achieves an accelerated $\mathcal{O}\left(-\frac{T}{\sqrt{\kappa(L)}}\right)$ linear decay rate in the bias term—in the sense that it depends on $\sqrt{\kappa(L)}$ rather than $\kappa(L)$—as well as an $\mathcal{O}\left(\frac{\sigma^2}{T}\right)$ decay rate in the variance term, up to the higher-order term in T. In fact, we can see from the variance term that the higher-order term in T, i.e. $\mathcal{O}\left(\frac{N\sqrt{\kappa(L)}\sigma^2}{T^2}\right)$, is dominated by $\frac{\sigma^2}{T}$ provided $T \gtrsim N\sqrt{\kappa(L)}$.

Remark 2: We argue that the convergence rate of SDA is optimal. We first consider the noiseless setting where $\sigma^2 = 0$. It follows from Eq. (10) that our algorithm has a linear convergence rate $\mathcal{O}\left(-\frac{T}{\sqrt{\kappa(L)}}\right)$. The proof of Theorem 2 in [33] implies that there exist a gossip matrix L and local functions f_i in the special class of quadratic functions considered in Section II such that for any black-box distributed optimization algorithm using L, the convergence rate is at least $\Omega\left(e^{-\frac{T}{\sqrt{\kappa(L)}}}\right)$. Thus, the bias term achieves the optimal rate. Next, we consider the noisy setting where $\sigma^2 > 0$. Suppose the network is fully connected (or the graph is complete), then every node can be viewed as the center node that receives information from every other node and thus the distributed setting is reduced to the centralized setting for every node. Standard information-theoretic lower bounds on estimating Gaussian means then yield a lower bound $\Omega\left(\frac{\sigma^2}{T}\right)$ (see also classical results due to [43]). Therefore, the rate of

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.

the variance term is optimal up to the higher-order term in T.

Remark 3: The primal accelerated method named D-MASG [36] is known to be asymptotically optimal for smooth and strongly convex distributed stochastic optimization problems [44]. If we apply Corollary 18 of [36] to our setting, we obtain that the estimates $\{\widehat{\mu}_T^i\}_{i=1}^N$ generated by this algorithm satisfy $\mathbb{E}\left[\sum_{i=1}^N \left\|\widehat{\mu}_T^i - \bar{\mu}\right\|_2^2\right] = \mathscr{O}\left(e^{-\frac{T}{\sqrt{\kappa(L)}}} + \frac{N\kappa(L)^2\sigma^2}{T^4} + \frac{\sigma^2}{T}\right)$. While the primal approach D-MASG and the dual approach SDA are rate-optimal for the online stochastic distributed averaging problem, their non-asymptotic behaviors can be significantly different. For example, in the non-asymptotic regime where $T \asymp \sqrt{\kappa(L)}$—which is the relevant regime given the accelerated rate of deterministic error—the upper bound on our algorithm scales as $\mathscr{O}\left(\frac{N\sigma^2}{T}\right)$, which is much better than the upper bound $\mathscr{O}\left(N\sigma^2\right)$ known to be achieved by D-MASG.

Remark 4: It is useful to compare with dual accelerated methods for smooth and strongly convex distributed stochastic optimization, R-RRMA-AC-SA^2 and SSTM_sc [38]. Applying Corollary 5.8 (for R-RRMA-AC-SA^2) and Corollary 5.14 (for SSTM_sc) from [38] to our setting, the oracle complexity (the number of oracle calls per node) and communication complexity (the number of communication rounds) for both methods are $\mathscr{O}\left(\kappa(L)\frac{\sigma^2}{\varepsilon^2}\right)$ and $\mathscr{O}\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$, where $\varepsilon > 0$ is the desired accuracy. Since these methods use batched stochastic dual gradients, let us make a small modification to SDA so as to facilitate a fair comparison. Specifically, suppose we change line 5 of SDA to: each agent i observes a batch local random vectors $\{R_{t,l}^i\}_{l=1}^{m_t}$ of size m_t, and executes the local update $\theta_t^i = x_t^i + \frac{1}{m_t}\sum_{l=1}^{m_t} R_{t,l}^i$. Under this modification, if we set $T = \mathscr{O}\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$ and $m_t = \mathscr{O}\left(\frac{\sigma^2}{\varepsilon\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)}\right)$ for all $t = 0,\ldots,T-1$, then Theorem 1 implies that $\mathbb{E}\left[\sum_{i=1}^N \left\|\widehat{\theta}_T^i - \bar{\mu}\right\|_2^2\right] \leq \mathscr{O}(\varepsilon)$. Therefore, the batched version of our method has oracle complexity $\mathscr{O}\left(\frac{\sigma^2}{\varepsilon}\right)$ and communication complexity $\mathscr{O}\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$. While our method and the ones above have the same communication complexity, the oracle complexity of our method is much smaller since it is independent of the condition number $\kappa(L)$ of the network and its dependence on ε is $\frac{1}{\varepsilon}$ instead of $\frac{1}{\varepsilon^2}$. It is also worth noting that the batched version of our method achieves the oracle complexity lower bound $\Omega\left(\frac{\sigma^2}{\varepsilon}\right)$ and the communication complexity lower bound $\Omega\left(\sqrt{\kappa(L)}\ln\left(\frac{1}{\varepsilon}\right)\right)$ simultaneously for our specific class of distributed stochastic optimization problems.

We now turn to describing how we can apply Theorem 1 to the two examples introduced in Section I to obtain their finite-time performance bounds. Due to the page limit, the corollaries and their proofs are omitted; see the full version [1] for details.

a) Distributed linear parameter estimation: We cover how to generate an estimate of $\beta^* = \bar{A}^{-1}\bar{\mu}$ at each sensor in this setting by estimating the global averages $\bar{A} = \frac{1}{N}\sum_{i=1}^N A_i^\top A_i$ and $\bar{\mu} = \frac{1}{N}\sum_{i=1}^N \mu_i$, respectively. We first run SDA with local variables $R_t^i = A_i{}^\top A_i$ for all $t = 0,\ldots,T'-1$ and $i \in \mathscr{N}$ to obtain the estimates $\{\widehat{A}_{T'}^i\}_{i\in\mathscr{N}}$ of $\bar{A}$. Since R_t^i is deterministic, iterate averaging is not necessary for the algorithm to converge. So we let SDA output the final iterates. Applying Theorem 1, we obtain the linear convergence bound

$$\sum_{i=1}^N \left\|\widehat{A}_{T'}^i - \bar{A}\right\|_2^2 \leq \mathscr{O}\left(e^{-\frac{T'}{\sqrt{\kappa(L)}}}\sum_{i=1}^N \left\|A_i^\top A_i - \bar{A}\right\|_2^2\right). \quad (11)$$

We assume that T' is chosen large enough such that $\left\|\bar{A}^{-1}\right\|_2\left\|\widehat{A}_{T'}^i - \bar{A}\right\|_2 \leq \frac{1}{2}$ for all $i \in \mathscr{N}$. Consequently, $\widehat{A}_{T'}^i$ is invertible for all $i \in \mathscr{N}$. Indeed, since $\bar{A}$ is assumed to be invertible, we have $\widehat{A}_{T'}^i = \bar{A}\left[I + \bar{A}^{-1}\left(\widehat{A}_{T'}^i - \bar{A}\right)\right]$, which implies that the invertibility of $\widehat{A}_{T'}^i$ is equivalent to the invertibility of $I + \bar{A}^{-1}\left(\widehat{A}_{T'}^i - \bar{A}\right)$. Since $\left\|\bar{A}^{-1}\left(\widehat{A}_{T'}^i - \bar{A}\right)\right\|_2 \leq \left\|\bar{A}^{-1}\right\|_2\left\|\widehat{A}_{T'}^i - \bar{A}\right\|_2 \leq \frac{1}{2}$, the matrix $I + \bar{A}^{-1}\left(\widehat{A}_{T'}^i - \bar{A}\right)$ has strictly positive eigenvalues and thus is invertible.

Next, we run SDA with local random variables $R_t^i = A_i^\top Y_t^i$ for all $t = 0,\ldots,T-1$ and $i \in \mathscr{N}$ to obtain the estimates $\{\widehat{\mu}_T^i\}_{i\in\mathscr{N}}$ of $\bar{\mu}$. Noting that R_t^i has zero mean and covariance matrix $A_i^\top \Sigma_\varepsilon^i A_i$, the individual variance $\sigma_{i,j}^2$ is the j-th element on the principal diagonal of $A_i^\top \Sigma_\varepsilon^i A_i$. Finally, each sensor i computes its own estimate $\widehat{\beta}_{T,T'}^i := \left(\widehat{A}_{T'}^i\right)^{-1}\widehat{\mu}_T^i$ of the unknown parameter β^*.

b) Decentralized multi-agent policy evaluation: In this setting, we first construct an unbiased estimator $\widehat{P}_T$ of the true transition matrix P using the common state transition samples. For $t = 0,\cdots,T-1$, each agent $i \in \mathscr{N}$ uses the set of sample state transitions $\{X_{t,j} | j \in \mathscr{S}\}$ to form a random binary matrix $Z_t \in \{0,1\}^{n\times n}$, in which row j has a single non-zero entry corresponding to the index of the sample $X_{t,j}$. Thus, the location of the non-zero entry in row j is drawn from the probability distribution $P(j,\cdot)$. Based on these observations, we define the common sample transition matrix $\widehat{P}_T := \frac{1}{T}\sum_{t=0}^{T-1} Z_t$. Next, we run SDA with local random vectors $R_t^i := \left[R_{t,1}^i,\ldots,R_{t,n}^i\right]^\top$ for all $t = 0,\ldots,T-1$ and $i \in \mathscr{N}$ to obtain estimates $\left\{\widehat{r}_T^i\right\}_{i\in\mathscr{N}}$ of $\bar{r}$. Finally, each agent i "plugs in" the estimates $(\widehat{P}_T, \widehat{r}_T^i)$ into the Bellman equation (1) to obtain the value function estimate $\widehat{J}_T^i := \left(I - \gamma\widehat{P}_T\right)^{-1}\widehat{r}_T^i$.

IV. Discussion

Let us conclude by mentioning a two future directions. One drawback of our setting is that the communication network is static. In many applications, the underlying connectivity structure of the network may vary with time, so a future direction is to extend our approach to this more challenging setting. Another drawback, from the perspective of distributed optimization, is that our setting only considers

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.

the special class of quadratic objective functions. Thus, a natural next step is to generalize our dual method to strongly convex and smooth local objective functions. This would allow us to make progress on the design of optimal dual-based algorithms under this setting, which is known to be an open problem in the literature [44].

REFERENCES

[1] S. Zhang, A. Pananjady, and J. Romberg, "A dual accelerated method for online stochastic distributed averaging: From consensus to decentralized policy evaluation," *arXiv preprint arXiv:2207.11425*, 2022.

[2] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005, pp. 63–70.

[3] S. Kar, J. M. Moura, and H. V. Poor, "Distributed linear parameter estimation: Asymptotically efficient adaptive strategies," *SIAM Journal on Control and Optimization*, vol. 51, no. 3, pp. 2200–2229, 2013.

[4] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[5] T. T. Doan, S. T. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation for multi-agent reinforcement learning," *arXiv preprint arXiv:1902.07393*, 2019.

[6] X. Zhang, Z. Liu, J. Liu, Z. Zhu, and S. Lu, "Taming communication and sample complexities in decentralized policy evaluation for cooperative multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[7] M. Kearns and S. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," *Advances in neural information processing systems*, pp. 996–1002, 1999.

[8] A. Pananjady and M. J. Wainwright, "Instance-dependent ℓ_∞-bounds for policy evaluation in tabular reinforcement learning," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 566–585, 2020.

[9] S. Giordani, M. Lujak, and F. Martinelli, "A distributed algorithm for the multi-robot task allocation problem," in *International conference on industrial, engineering and other applications of applied intelligent systems*. Springer, 2010, pp. 721–730.

[10] Q. Peng and S. H. Low, "Distributed algorithm for optimal power flow on a radial network," in *53rd IEEE Conference on decision and control*. IEEE, 2014, pp. 167–172.

[11] N. A. Lynch, *Distributed algorithms*. Elsevier, 1996.

[12] H. Li and X. Li, "Distributed consensus of heterogeneous linear time-varying systems on UAVs–USVs coordination," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1264–1268, 2019.

[13] H. Zhang, X. Zhou, Z. Wang, H. Yan, and J. Sun, "Adaptive consensus-based distributed target tracking with dynamic cluster in sensor networks," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1580–1591, 2018.

[14] R. Tutunov, H. Bou-Ammar, and A. Jadbabaie, "Distributed Newton method for large-scale consensus optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3983–3994, 2019.

[15] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology. Laboratory for Information and . . . , 1984.

[16] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[17] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[18] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, vol. 67, no. 1, pp. 33–46, 2007.

[19] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM journal on control and optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[20] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[21] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

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

[23] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[24] H. Li, Q. Lü, and T. Huang, "Convergence analysis of a distributed optimization algorithm with a general unbalanced directed communication network," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 237–248, 2018.

[25] A. Nedic, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 92–101, 2020.

[26] S. Dougherty and M. Guay, "An extremum-seeking controller for distributed optimization over sensor networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 928–933, 2016.

[27] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[28] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[29] A. Nedich *et al.*, "Convergence rate of distributed averaging dynamics and optimization in networks," *Foundations and Trends® in Systems and Control*, vol. 2, no. 1, pp. 1–100, 2015.

[30] P. Giselsson and A. Rantzer, *Large-scale and distributed optimization*. Springer, 2018, vol. 2227.

[31] 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.

[32] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[33] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *International conference on machine learning*. PMLR, 2017, pp. 3027–3036.

[34] H. Li, Z. Lin, and Y. Fang, "Variance reduced extra and diging and their optimal acceleration for strongly convex decentralized optimization," *arXiv preprint arXiv:2009.04373*, 2020.

[35] H. Hendrikx, F. Bach, and L. Massoulie, "An optimal algorithm for decentralized finite-sum optimization," *SIAM Journal on Optimization*, vol. 31, no. 4, pp. 2753–2783, 2021.

[36] A. Fallah, M. Gurbuzbalaban, A. Ozdaglar, U. Simsekli, and L. Zhu, "Robust distributed accelerated stochastic gradient methods for multi-agent networks," *arXiv preprint arXiv:1910.08701*, 2019.

[37] A. Rogozin, M. Bochko, P. Dvurechensky, A. Gasnikov, and V. Lukoshkin, "An accelerated method for decentralized distributed stochastic optimization over time-varying graphs," *arXiv preprint arXiv:2103.15598*, 2021.

[38] E. Gorbunov, D. Dvinskikh, and A. Gasnikov, "Optimal decentralized distributed algorithms for stochastic convex optimization," *arXiv preprint arXiv:1911.07363*, 2019.

[39] D. Dvinskikh, E. Gorbunov, A. Gasnikov, P. Dvurechensky, and C. A. Uribe, "On dual approach for distributed stochastic convex optimization over networks," *arXiv preprint arXiv:1903.09844*, 2019.

[40] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for non-smooth distributed optimization in networks," *arXiv preprint arXiv:1806.00291*, 2018.

[41] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.

[42] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[43] A. S. Nemirovski and D. B. Yudin, "Problem complexity and method efficiency in optimization," 1983.

[44] E. Gorbunov, A. Rogozin, A. Beznosikov, D. Dvinskikh, and A. Gasnikov, "Recent theoretical advances in decentralized distributed convex optimization," *arXiv preprint arXiv:2011.13259*, 2020.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on March 15,2025 at 20:27:42 UTC from IEEE Xplore. Restrictions apply.