Convergence rates for distributed stochastic optimization over random networks

Dusan Jakovetic, Dragana Bajovic, Anit Kumar Sahu and Soummya Kar

Abstract— **We establish the $O(\frac{1}{k})$ convergence rate for distributed stochastic gradient methods that operate over strongly convex costs and random networks. The considered class of methods is standard - each node performs a weighted average of its own and its neighbors' solution estimates (consensus), and takes a negative step with respect to a noisy version of its local function's gradient (innovation). The underlying communication network is modeled through a sequence of temporally independent identically distributed (i.i.d.) Laplacian matrices such that the underlying graphs are connected on average; the local gradient noises are also i.i.d. in time, have finite second moment, and possibly unbounded support. We show that, after a careful setting of the consensus and innovations potentials (weights), the distributed stochastic gradient method achieves a (order-optimal) $O(\frac{1}{k})$ convergence rate in the mean square distance from the solution. To the best of our knowledge, this is the first order-optimal convergence rate result on distributed strongly convex stochastic optimization when the network is random and the gradient noises have unbounded support. Simulation examples confirm the theoretical findings.**

1. Introduction

Distributed optimization and learning algorithms attract a great interest in recent years, thanks to their widespread applications including distributed estimation in networked systems [1], distributed control [2], and big data analytics [3].

In this paper, we study distributed stochastic optimization algorithms that operate over random networks and minimize smooth strongly convex costs. We consider standard distributed stochastic gradient methods where at each time step, each node makes a weighted average of its own and its neighbors' solution estimates, and performs a step in the negative direction of its noisy local gradient. The underlying network is allowed to be *randomly varying*, similarly to, e.g., the models in [4]–[6]. More specifically, the network is modeled through a sequence of independent identically distributed (i.i.d.) graph Laplacian matrices, where the network is assumed to be connected on average; in other words, the graph that supports the mean Laplacian matrix is assumed to be connected. (This translates into the requirement that the algebraic connectivity of the mean Laplacian matrix is strictly positive.) Random network models are highly relevant in, e.g., internet of things (IoT) and cyber physical systems (CPS) applications, like, e.g., predictive maintenance and monitoring in industrial manufacturing systems, monitoring smart buildings, etc. Therein, networked nodes often communicate through unreliable/intermittent wireless links, due to, e.g., low-power transmissions or harsh environments.

The main contributions of the paper are as follows. We show that, by carefully designing the consensus and the gradient weights (potentials), the considered distributed stochastic gradient algorithm achieves the order-optimal $O(1/k)$ rate of decay of the mean squared distance from the solution (mean squared error – MSE). This is achieved for twice continuously differentiable strongly convex local costs, assuming also that the noisy gradients are unbiased estimates of the true gradients and that the noise in gradients has bounded second moment. To the best of our knowledge, this is the first time an order-optimal convergence rate for distributed strongly convex stochastic optimization has been established for random networks and noises with unbounded support.

We now briefly review the literature to help us contrast this paper from prior work. There have been many works related with distributed stochastic optimization, e.g., [7]–[9]. The consensus problem, a special form of distributed optimization problems wherein each local function is scalar quadratic, has also been extensively studied under various stochastic imperfections, e.g., [10]–[12]. Closer to our work are the references on: 1) distributed strongly convex stochastic (sub)gradient methods; and 2) distributed (sub)gradient methods over random networks (both deterministic and stochastic methods). For the former thread of works, several papers give explicit convergence rates under different assumptions. Regarding the underlying network, references [13], [14] consider static networks, while the works [15]–[17] consider deterministic time-varying networks.

References [13], [14] consider distributed strongly convex optimization for static networks, assuming that the data distributions that underlie each node's local cost function are equal (reference [13] considers empirical risks while reference [14] considers risk functions in the form of expectation);

This work is supported by the I-BiDaaS project, funded by the European Commission under Grant Agreement No. 780787. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein. The research is also partly supported by European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 734331. The work of D. Jakovetic is also supported in part by the Serbian Ministry of Education, Science, and Technological Development, grant 174030. The work of AKS and SK was supported in part by National Science Foundation under grant CCF-1513936.

D. Bajovic is with University of Novi Sad, Faculty of Technical Sciences, Department of Power, Electronic and Communication Engineering, 21000 Novi Sad, Serbia dbajovic@uns.ac.rs

D. Jakovetic is with University of Novi Sad, Faculty of Sciences, Department of Mathematics and Informatics, 21000 Novi Sad, Serbia djakovet@uns.ac.rs

A. K. Sahu and S. Kar are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 {anits,soummyak}@andrew.cmu.edu

this essentially corresponds to each nodes' local function having the same minimizer. References [15]–[17] consider deterministically varying networks, assuming that the "union graph" over finite windows of iterations is connected. The papers [13]–[16] assume undirected networks, while [17] allows for directed networks but assumes a bounded support for the gradient noise. The works [13], [15]–[17] allow the local costs to be non-smooth, while [14] assumes smooth costs, as we do here. With respect to these works, we consider random networks, undirected networks, smooth costs, and allow the noise to have unbounded support.

Distributed optimization over random networks has been studied in [4]–[6], [18]. References [4], [5] consider non-differentiable convex costs and no (sub)gradient noise, while reference [6] considers differentiable costs with Lipschitz continuous and bounded gradients, and it also does not allow for gradient noise, i.e., it considers methods with exact (deterministic) gradients. Reference [18] proposes distributed proximal methods for composite optimization problems and establishes for the methods (in the absence of strong convexity assumption) $O(1/k)$ and $O(1/k^{1/2})$ rates for the noiseless and noisy gradient cases, respectively. In [19], we consider a distributed Kiefer-Wolfowitz-type stochastic approximation method and establish the method's $O(1/k^{1/2})$ convergence rate. Reference [19] complements the current paper by assuming that nodes only have access to noisy functions' estimates (zeroth order optimization), and no gradient estimates are available. In contrast, by assuming a noisy first-order (gradient) information is available, we show here that strictly faster rates than in [19] can be achieved.

Paper organization. The next paragraph introduces notation. Section 2 describes the model and the stochastic gradient method we consider. Section 3 states and proves the main result on the algorithm's MSE convergence rate. Section 4 provides a simulation example. Finally, we conclude in Section 5.

Notation. We denote by $\mathbb{R}$ the set of real numbers and by $\mathbb{R}^m$ the m-dimensional Euclidean real coordinate space. We use normal lower-case letters for scalars, lower case boldface letters for vectors, and upper case boldface letters for matrices. Further, we denote by: $\mathbf{A}_{ij}$ the entry in the i-th row and j-th column of a matrix $\mathbf{A}$; $\mathbf{A}^\top$ the transpose of a matrix $\mathbf{A}$; $\otimes$ the Kronecker product of matrices; $\mathbf{I}$, 0, and $\mathbf{1}$, respectively, the identity matrix, the zero matrix, and the column vector with unit entries; $\mathbf{J}$ the $N \times N$ matrix $\mathbf{J} := (1/N)\mathbf{1}\mathbf{1}^\top$. When necessary, we indicate the matrix or vector dimension through a subscript. Next, $A \succ 0\,(A \succeq 0)$ means that the symmetric matrix A is positive definite (respectively, positive semi-definite). We further denote by: $\|\cdot\| = \|\cdot\|_2$ the Euclidean (respectively, spectral) norm of its vector (respectively, matrix) argument; $\lambda_i(\cdot)$ the i-th smallest eigenvalue; $\nabla h(w)$ and $\nabla^2 h(w)$ the gradient and Hessian, respectively, evaluated at w of a function $h : \mathbb{R}^m \to \mathbb{R}$, $m \geq 1$; $\mathbb{P}(\mathcal{A})$ and $\mathbb{E}[u]$ the probability of an event $\mathcal{A}$ and expectation of a random variable u, respectively. Finally, for two positive sequences η_n and χ_n, we have: $\eta_n = O(\chi_n)$ if $\limsup_{n\to\infty} \frac{\eta_n}{\chi_n} < \infty$.

2. Model and algorithm

A. Optimization and network models

We consider the scenario where N networked nodes aim to collaboratively solve the following unconstrained problem:

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(\mathbf{x}), \tag{1}$$

where $f_i : \mathbb{R}^m \mapsto \mathbb{R}$ is a convex function available to node i, $i = 1, ..., N$, and minimization in (1) is with respect to variable $\mathbf{x} in \mathbb{R}^m$. We make the following standard assumption on the f_i's.

Assumption 1. For all $i = 1, ..., N$, function $f_i : \mathbb{R}^m \mapsto \mathbb{R}$ is twice continuously differentiable, and there exist constants $0 < \mu \leq L < \infty$, such that, for all $\mathbf{x} \in \mathbb{R}^m$, there holds:

$$\mu\,\mathbf{I} \preceq \nabla^2 f_i(\mathbf{x}) \preceq L\,\mathbf{I}.$$

Assumption 1 implies that each f_i is strongly convex with modulus μ, and it also has Lipschitz continuous gradient with Lipschitz constant L, i.e., the following two inequalities hold for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$:

$$f_i(\mathbf{y}) \geq f_i(\mathbf{x}) + \nabla f_i(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2$$
$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\,\|\mathbf{x} - \mathbf{y}\|.$$

Furthermore, under Assumption 1, problem (1) is solvable and has the unique solution, which we denote by $x^\star \in \mathbb{R}^m$. For future reference, introduce also the sum function $f : \mathbb{R}^m \to \mathbb{R}$, defined by $f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x})$.

We consider distributed stochastic gradient methods to solve (1) over random networks. Specifically, we adopt the following model. At each time instant $k = 0, 1, ...$, the underlying network $\mathcal{G}(k) = (V, \mathbf{E}(k))$ is undirected and random, with $V = \{1, ..., N\}$ the set of nodes, and $\mathbf{E}(k)$ the random set of undirected edges. We denote by $\{i, j\}$ the edge that connects nodes i and j. Further, denote by 7 $\Omega_i(k) = \{j \in V : \; \{i, j\} \in \mathbf{E}(k)\}$ the random neighborhood od node i at time k (excluding node i). We associate to $\mathcal{G}(k)$ its $N \times N$ (symmetric) Laplacian matrix $\boldsymbol{\mathcal{L}}(k)$, defined by: $\boldsymbol{\mathcal{L}}_{ij}(k) = -1$, if $\{i, j\} \in \mathbf{E}(k)$, $i \neq j$; $\boldsymbol{\mathcal{L}}_{ij}(k) = 0$, if $\{i, j\} \notin \mathbf{E}(k)$, $i \neq j$; and $\boldsymbol{\mathcal{L}}_{ii}(k) = -\sum_{j \neq i} \boldsymbol{\mathcal{L}}_{ij}(k)$. Denote by $\overline{\boldsymbol{\mathcal{L}}} = \mathbb{E}[\,\boldsymbol{\mathcal{L}}(k)\,]$ (as explained ahead, the expectation is independent of k), and let $\overline{\mathcal{G}} = (V, \overline{\mathbf{E}})$ be the graph induced by matrix $\overline{\boldsymbol{\mathcal{L}}}$, i.e., $\overline{\mathbf{E}} = \{\{i, j\} : \; i \neq j, \;\; \overline{\boldsymbol{\mathcal{L}}}_{ij} > 0\}$. We make the following assumption.

Assumption 2. The matrices $\{\mathbf{L}(k)\}$ are independent, identically distributed (i.i.d.). Furthermore, graph $\overline{\mathcal{G}}$ is connected.

It is well-known that the connectedness of $\overline{\mathcal{G}}$ is equivalent to the condition $\lambda_2\left(\overline{\boldsymbol{\mathcal{L}}}\right) > 0$.

B. Gradient noise model and the algorithm

We consider the following distributed stochastic gradient method to solve (1). Each node i, $i = 1, ..., N$, maintains over time steps (iterations) $k = 0, 1, ...$, its solution estimate $\mathbf{x}_i(k) \in \mathbb{R}^m$. Specifically, for arbitrary deterministic initial points $\mathbf{x}_i(0) \in \mathbb{R}^m$, $i = 1, ..., N$, the update rule at node i and $k = 0, 1, ...$, is as follows:

$$\begin{aligned} \mathbf{x}_i(k+1) &= \mathbf{x}_i(k) - \beta_k \sum_{j \in \Omega_i(k)} \left(\mathbf{x}_i(k) - \mathbf{x}_j(k)\right) \quad (2) \\ &- \alpha_k \left(\nabla f_i(\mathbf{x}_i(k)) + \mathbf{v}_i(k) \right). \end{aligned}$$

The update (2) is performed in parallel by all nodes $i = 1, ..., N$. The algorithm iteration is realized as follows. First, each node i broadcasts $\mathbf{x}_i(k)$ to all its available neighbors $j \in \Omega_i(k)$, and receives $\mathbf{x}_j(k)$ from all $j \in \Omega_i(k)$. Subsequently, each node i, $i = 1, ..., N$ makes update (2), which completes an iteration. In (2), α_k is the step-size that we set to $\alpha_k = \alpha_0/(k+1)$, $k = 0, 1, ...$, with $\alpha_0 > 0$; and β_k is the (possibly) time-varying weight that each node assigns to all its neighbors. We set $\beta_k = \beta_0/(k+1)^\nu$, $k = 0, 1, ...$, with $\nu \in [0, 1/2]$. Here, $\beta_0 > 0$ is a constant that should be taken to be sufficiently small; e.g., one can set $\beta_0 = 1/(1+\theta)$, where θ is the maximal degree (number of neighbors of a node) across network. Finally, $\mathbf{v}_i(k)$ is noise in the calculation of the f_i's gradient at iteration k.

For future reference, we also present algorithm (2) in matrix format. Denote by $\mathbf{x}(k) = \left[\mathbf{x}_1^\top(k), \cdots, \mathbf{x}_N^\top(k)\right]^\top \in \mathbb{R}^{Nm}$ the vector that stacks the solution estimates of all nodes. Also, define function $F : \mathbb{R}^{Nm} \mapsto \mathbb{R}$, by $F(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}_i)$, with $\mathbf{x} = \left[\mathbf{x}_1^\top, \cdots, \mathbf{x}_N^\top\right]^\top \in \mathbb{R}^{Nm}$. Finally, let $\mathbf{W}_k = (\mathbf{I} - \mathbf{L}_k) \otimes \mathbf{I}_m$, where $\mathbf{L}_k = \beta_k \boldsymbol{\mathcal{L}}(k)$. Then, for $k = 0, 1, ...$, algorithm (2) can be compactly written as follows:

$$\mathbf{x}(k+1) = \mathbf{W}_k \mathbf{x}(k) - \alpha_k \left(\nabla F(\mathbf{x}(k)) + \mathbf{v}(k)\right). \quad (3)$$

We make the following standard assumption on the gradient noises. First, denote by $\mathcal{F}_k$ the history of algorithm (2) up to time k; that is, $\mathcal{F}_k$, $k = 1, 2, ...$, is an increasing sequence of sigma algebras, where $\mathcal{F}_k$ is the sigma algebra generated by the collection of random variables $\{\, \boldsymbol{\mathcal{L}}(s),\, \mathbf{v}_i(t)\}$, $i = 1, ..., N$, $s = 0, ..., k-1$, $t = 0, ..., k-1$.

Assumption 3. For each $i = 1, ..., N$, the sequence of noises $\{\mathbf{v}_i(k)\}$ satisfies for all $k = 0, 1, ...$:

$$\begin{aligned} \mathbb{E}[\, \mathbf{v}_i(k) \,|\, \mathcal{F}_k \,] &= 0, \text{ almost surely (a.s.)} \quad (4) \\ \mathbb{E}[\, \|\mathbf{v}_i(k)\|^2 \,|\, \mathcal{F}_k \,] &\leq c_v \|\mathbf{x}_i(k)\|^2 + c_v', \text{ a.s.,} \quad (5) \end{aligned}$$

where c_v and c_v' are nonnegative constants.

Assumption 3 is satisfied, for example, when $\{\mathbf{v}_i(k)\}$ is an i.i.d. zero-mean, finite second moment, noise sequence such that $\mathbf{v}_i(k)$ is also independent of history $\mathcal{F}_k$. However, the assumption allows that the gradient noise $\mathbf{v}_i(k)$ be dependent on node i and also on the current point $\mathbf{x}_i(k)$; the next subsection gives some important machine learning settings encompassed by Assumption 3.

C. Motivation

The optimization-algorithmic model defined by Assumptions 1–3 subsumes, e.g., important machine learning applications. Consider the scenario where f_i corresponds to the risk function associated with the node i's local data, i.e.,

$$f_i(\mathbf{x}) = \mathbb{E}_{\mathbf{d}_i \sim P_i} \left[\, \ell_i\left(\mathbf{x}; \mathbf{d}_i\right) \right] + \Psi_i(\mathbf{x}). \quad (6)$$

Here, P_i is node i's local distribution according to which its data samples $\mathbf{d}_i \in \mathbb{R}^q$ are generated; $\ell_i(\cdot; \cdot)$ is a loss function that is convex in its first argument for any fixed value of its second argument; and $\Psi : \mathbb{R}^m \to \mathbb{R}$ is a strongly convex regularizer. Similarly, f_i can be an empirical risk function:

$$f_i(\mathbf{x}) = \frac{1}{n_i} \left(\sum_{j=1}^{n_i} \ell_i\left(\mathbf{x}; \mathbf{d}_{i,j}\right) \right) + \Psi_i(\mathbf{x}), \quad (7)$$

where $\mathbf{d}_{i,j}$, $j = 1, ..., n_i$, is the set of training examples at node i. Examples for the loss $\ell_i(\cdot; \cdot)$ include the following:

$$\begin{aligned} \ell_i(\mathbf{x}; \mathbf{a_i}, b_i) &= \frac{1}{2}\left(\mathbf{a}_i^\top \mathbf{x} - b_i\right)^2 \quad \text{(quadratic loss)} \quad (8) \\ \ell_i(\mathbf{x}; \mathbf{a_i}, b_i) &= \ln\left(1 + \exp(-b_i(\mathbf{a}_i^\top \mathbf{x}))\right) \quad \text{(logistic loss)} \end{aligned}$$

For the quadratic loss above, a data sample $\mathbf{d}_i = (\mathbf{a}_i, b_i)$, where $\mathbf{a}_i$ is a regressor vector and b_i is a response variable; for the logistic loss, $\mathbf{a}_i$ is a feature vector and $b_i \in \{-1, +1\}$ is its class label. Clearly, both the risk (6) and the empirical risk (7) satisfy Assumption 1 for the losses in (8).

We next discuss the search directions in (2) and Assumption 3 for the gradient noise. A common search direction in machine learning algorithms is the gradient of the loss with respect to a single data point[1]:

$$g_i(\mathbf{x}) = \nabla \ell_i\left(\mathbf{x}; \mathbf{d}_i\right) + \nabla \Psi_i(\mathbf{x}).$$

In case of the risk function (8), $\mathbf{d}_i$ is drawn from distribution P_i; in case of the empirical risk (7), $\mathbf{d}_i$ can be, e.g., drawn uniformly at random from the set of data points $\mathbf{d}_{i,j}$, $j = 1, ..., n_i$, with repetition along iterations. In both cases, gradient noise $\mathbf{v}_i = g_i(\mathbf{x}) - \nabla f_i(\mathbf{x})$ clearly satisfies assumption (4). To see this, consider, for example, the risk function (6), and let us fix iteration k and node i's estimate $\mathbf{x}_i(k) = \mathbf{x}_i$. Then,

$$\begin{aligned} &\mathbb{E}\left[\mathbf{v}_i(k) \,|\, \mathcal{F}_k\right] = \mathbb{E}\left[\mathbf{g}_i(k) - \nabla f_i(\mathbf{x}_i(k)) \,|\, \mathbf{x}_i(k) = \mathbf{x}_i\right] \\ = \;& \mathbb{E}[\nabla \ell_i\left(\mathbf{x}_i(k); \mathbf{d}_i\right) \,|\, \mathbf{x}_i(k) = \mathbf{x}_i] + \nabla \Psi_i(\mathbf{x}_i) \\ - \;& (\nabla f_i(\mathbf{x}_i) + \nabla \Psi_i(\mathbf{x})) \\ = \;& \mathbb{E}_{\mathbf{d}_i \sim P_i}[\nabla \ell_i\left(\mathbf{x}; \mathbf{d}_i\right)] + \nabla \Psi_i(\mathbf{x}_i) \\ - \;& (\mathbb{E}_{\mathbf{d}_i \sim P_i}[\nabla \ell_i\left(\mathbf{x}; \mathbf{d}_i\right)] + \nabla \Psi_i(\mathbf{x}_i)) = 0. \end{aligned}$$

Further, for the empirical risk, assumption (5) holds trivially. For the risk function (6), assumption (5) holds for a sufficiently "regular" distribution P_i. For instance, it is easy to show that the assumption holds for the logistic loss in (8) when P_i has finite second moment, while it holds for the square loss in (8) when P_i has finite fourth moment.

Note that our setting allows that the data generated at different nodes be generated through different distributions P_i, as well as that the nodes utilize different losses ℓ_i's and regularizers Ψ_i's. Mathematically, this means that $\nabla f_i(x^\star) \neq 0$,

[1] Similar considerations hold for a loss with respect to a mini-batch of data points; this discussion is abstracted for simplicity.

in general. In words, if a node i relies only on its local data $\mathbf{d}_i$, it cannot recover the true solution $x^\star$. Nodes then engage in a collaborative algorithm (2) through which, as shown ahead, they can recover the global solution $x^\star$.

We close the section by motivating the random network model considered here (Assumption 2). Random network models are adequate for the scenarios of unreliable wireless communications, when links may fail intermittently at random times. In addition, they are useful to model randomized communication protocols like gossip [20].

3. Performance Analysis

A. Statement of main results and auxiliary lemmas

We are now ready to state our main result.

Theorem 3.1. *Consider algorithm* (2) *with step-sizes* $\alpha_k = \frac{\alpha_0}{k+1}$ *and* $\beta_k = \frac{\beta_0}{(k+1)^\nu}$, *where* $\beta_0 > 0$, $\alpha_0 > 2\,N/\mu$, *and* $\nu \in [0, 1/2]$. *Further, let Assumptions 1–3 hold. Then, for each node i's solution estimate* $\mathbf{x}_i(k)$ *and the solution* $\mathbf{x}^\star$ *of problem* (1), *there holds:*

$$\mathbb{E}\left[\|\mathbf{x}_i(k) - \mathbf{x}^\star\|^2\right] = O(1/k).$$

We remark that the condition $\alpha_0 > 2\,N/\mu$ can be relaxed to require only a positive α_0, in which case the rate becomes $O(\ln(k)/k)$, instead of $O(1/k)$.[2] Also, to avoid large step-sizes at initial iterations for a large α_0, step-size α_k can be modified to $\alpha_k = \alpha_0/(k + k_0)$, for arbitrary positive constant k_0, and Theorem 3.1 continues to hold. Theorem 3.1 establishes the $O(1/k)$ MSE rate of convergence of algorithm (2); due to the assumed f_i's strong convexity, the theorem also implies that $\mathbb{E}\left[f(\mathbf{x}_i(k)) - f(\mathbf{x}^\star)\right] = O(1/k)$. Note that the expectation in Theorem 3.1 is both with respect to randomness in gradient noises and with respect to the randomness in the underlying network. The $O(1/k)$ rate does not depend on the statistics of the underlying random network, as long as the network is connected on average (i.e., satisfies Assumption 2.) The hidden constant depends on the underlying network statistics, but simulation examples suggest that the dependence is usually not strong (see Section 4).

Our strategy for proving Theorem 3.1 is as follows. We first establish the mean square boundedness (uniform in k) of the iterates $\mathbf{x}_i(k)$, which also implies the uniform mean square boundedness of the gradients $\nabla f_i(\mathbf{x}_i(k))$ (Subsection 3-B). We then bound, in the mean square sense, the disagreements of different nodes' estimates, i.e., quantities $(\mathbf{x}_i(k) - \mathbf{x}_j(k))$, showing that $\mathbb{E}[\,\|\mathbf{x}_i(k) - \mathbf{x}_j(k)\|^2\,] = O(1/k)$ (Subsection 3-C). This allows us to show that the (hypothetical) global average of the nodes' solution estimates $\overline{\mathbf{x}}(k) := \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i(k)$ evolves according to a stochastic gradient method with the gradient estimates that have a sufficiently small bias and finite second moment. This allows us to show the $O(1/k)$ rate on the mean square error at the global average, which in turn allows to derive a similar bound at the individual nodes' estimates (Subsection 3-D).

We provide here proof outlines where we do not detail intermediate constants but focus on establishing the convergence rate in terms of the iteration counter; the convergence constants will be characterized in more detail elsewhere.

In completing the strategy above, we make use of the following Lemma; the Lemma is a minor modification of Lemmas 4 and 5 in [1].

Lemma 3.2. *Let* $z(k)$ *be a nonnegative (deterministic) sequence satisfying:*

$$z(k+1) \leq (1 - r_1(k))\, z_1(k) + r_2(k),$$

where $\{r_1(k)\}$ *and* $\{r_2(k)\}$ *are deterministic sequences with*

$$\frac{a_1}{(k+1)^{\delta_1}} \leq r_1(k) \leq 1 \text{ and } r_2(k) \leq \frac{a_2}{(k+1)^{\delta_2}},$$

with $a_1, a_2, \delta_1, \delta_2 > 0$. *Then, (a) if* $\delta_1 = \delta_2 = 1$, *there holds:* $z(k) = O(1)$; *(b) if* $\delta_1 = 1/2$ *and* $\delta_2 = 3/2$, *then* $z(k) = O(1/k)$; *and (c) if* $\delta_1 = 1$, $\delta_2 = 2$, *and* $a_1 > 1$, *then* $z(k) = O(1/k)$.

Subsequent analysis in Subsections 3-b until 3-d restricts to the case when $\nu = 1/2$, i.e., when consensus weights equal $\beta_k = \frac{\beta_0}{(k+1)^{1/2}}$. That is, for simplicity of presentation, we prove Theorem 3.1 for case $\nu = 1/2$. As it can be verified in subsequent analysis, the proof of Theorem 3.1 extends to a generic $\mu \in [0, 1/2)$ as well. As another step in simplifying notations, throughout Subsections 3-b and 3-c, we let $m = 1$ to avoid extensive usage of Kronecker products; again, the proofs extend to a generic $m > 1$.

B. Mean square boundedness of the iterates

This Subsection shows the uniform mean square boundedness of the algorithm iterates.

Lemma 3.3. *Consider algorithm* (2), *and let Assumptions 1-3 hold. Then, there exist nonnegative constants* c_x *and* $c_{\partial f}$ *such that, for all* $k = 0, 1, ...$, *there holds:*

$$\mathbb{E}[\,\|\mathbf{x}(k)\|^2\,] \leq c_x \text{ and } \mathbb{E}[\,\|\nabla F(\mathbf{x}(k))\|^2\,] \leq c_{\partial f}.$$

Proof.

Denote by $\mathbf{x}^o = x^* \mathbf{1}_N$ and recall (3). Then, we have:

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}^o &= \mathbf{W}_k(\mathbf{x}(k) - \mathbf{x}^o) \\ &- \alpha_k\left(\nabla F(\mathbf{x}(k)) - \nabla F(\mathbf{x}^o)\right) \\ &- \alpha_k \mathbf{v}(k) - \alpha_k \nabla F(\mathbf{x}^o). \end{aligned} \tag{9}$$

By mean value theorem, we have:

$$\begin{aligned} &\nabla F(\mathbf{x}(k)) - \nabla F(\mathbf{x}^o) \\ &= \left[\int_{s=0}^1 \nabla^2 F\left(\mathbf{x}^o + s(\mathbf{x}(k) - \mathbf{x}^o)\right) d\,s\right](\mathbf{x}(k) - \mathbf{x}^o) \\ &= \mathbf{H}_k\left(\mathbf{x}(k) - \mathbf{x}^o\right). \end{aligned} \tag{10}$$

Note that $L\mathbf{I} \succcurlyeq \mathbf{H}_k \succcurlyeq \mu\mathbf{I}$. Using (10) in (9) we have:

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}^o &= \left(\mathbf{W}_k - \alpha_k \mathbf{H}_k\right)(\mathbf{x}(k) - \mathbf{x}^o) \quad (11) \\ - \quad & \alpha_k \mathbf{v}(k) - \alpha_k \nabla F(\mathbf{x}^o). \end{aligned}$$

[2] This subtlety comes from equation (32) ahead and the requirement that $c_{20} > 1$. If $c_{20} \leq 1$, it can be shown that in (34) the right hand side modifies to a $O(\ln(k)/k)$ quantity.

Denote by $\boldsymbol{\zeta}(k) = \mathbf{x}(k) - \mathbf{x}^o$ and by $\boldsymbol{\xi}(k) = (\mathbf{W}_k - \alpha_k \mathbf{H}_k)\,(\mathbf{x}(k) - \mathbf{x}^o) - \alpha_k\,\nabla F(\mathbf{x}^o)$. Then, there holds:

$$\begin{aligned} & \mathbb{E}[\,\|\boldsymbol{\zeta}(k+1)\|^2\,|\,\mathcal{F}_k\,] \le \|\boldsymbol{\xi}(k)\|^2 \\ - \quad & 2\alpha_k\,\boldsymbol{\xi}(k)^\top \mathbb{E}[\,\mathbf{v}(k)\,|\,\mathcal{F}_k\,] + \alpha_k^2\,\mathbb{E}[\,\|\mathbf{v}(k)\|^2\,|\,\mathcal{F}_k\,] \\ \le \quad & \|\boldsymbol{\xi}(k)\|^2 + N\,\alpha_k^2\,(c_v\,\|\mathbf{x}(k)\|^2 + c_v'),\ \text{a.s.,} \end{aligned} \tag{12}$$

where we used Assumption 3 and the fact that $\boldsymbol{\xi}(k)$ is measurable with respect to $\mathcal{F}_k$. We next bound $\|\boldsymbol{\xi}(k)\|^2$. Note that $\|\mathbf{W}_k - \alpha_k\,\boldsymbol{H}_k\| \le 1 - \mu\,\alpha_k$ for sufficiently large k. Therefore, we have for sufficiently large k:

$$\|\boldsymbol{\xi}(k)\| \le (1 - \mu\,\alpha_k)\,\|\boldsymbol{\zeta}(k)\| + \alpha_k\,\|\nabla F(\mathbf{x}^o)\|. \tag{13}$$

We now use the following inequality:

$$(a+b)^2 \le (1+\theta)\,a^2 + \left(1 + \frac{1}{\theta}\right) b^2, \tag{14}$$

for any $a, b \in \mathbb{R}$ and $\theta > 0$. We set $\theta = \frac{c_0}{k+1}$, with $c_0 > 0$. Using the inequality (14) in (13), we have:

$$\begin{aligned} & \|\boldsymbol{\xi}(k)\|^2 \le \left(1 + \frac{c_0}{k+1}\right)(1 - \alpha_k \mu)^2 \\ \times \quad & \|\boldsymbol{\zeta}(k)\|^2 + \left(1 + \frac{k+1}{c_0}\right)\alpha_k^2 \|\nabla F(\mathbf{x}^o)\|^2. \end{aligned}$$

Next, for $c_0 < \alpha_0 \mu$, the last inequality implies:

$$\begin{aligned} & \|\boldsymbol{\xi}(k)\|^2 \le \left(1 - \frac{c_1}{k+1}\right)\|\boldsymbol{\zeta}(k)\|^2 \\ & + \frac{c_2}{k+1}\|\nabla F(\mathbf{x}^o)\|^2, \end{aligned} \tag{15}$$

for some constants $c_1, c_2 > 0$. Combining (15) and (12), we get:

$$\begin{aligned} \mathbb{E}[\,\|\boldsymbol{\zeta}(k+1)\|^2\,|\,\mathcal{F}_k\,] \quad & \le \quad \left(1 - \frac{c_1'}{k+1}\right)\|\boldsymbol{\zeta}(k)\|^2 \\ & + \quad \frac{c_2'}{k+1}, \end{aligned} \tag{16}$$

for some $c_1', c_2' > 0$. Taking expectation in (16) and applying Lemma 3.2, it follows that $\mathbb{E}[\,\|\boldsymbol{\zeta}(k)\|^2\,] = \mathbb{E}[\,\|\mathbf{x}(k) - \mathbf{x}^o\|^2\,]$ is uniformly (in k) bounded from above by a positive constant. It is easy to see that the latter implies that $\mathbb{E}[\,\|\mathbf{x}(k)\|^2\,]$ is also uniformly bounded. Using the Lipschitz continuity of ∇F, we finally also have that $\mathbb{E}[\,\|\nabla F(\mathbf{x}(k))\|^2\,]$ is also uniformly bounded. The proof of Lemma 3.3 is now complete.

C. Disagreement bounds

Recall the (hypothetically available) global average of nodes' estimates $\overline{\mathbf{x}}(k) = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i(k)$, and denote by $\widetilde{\mathbf{x}}_i(k) = \mathbf{x}_i(k) - \overline{\mathbf{x}}(k)$ the quantity that measures how far apart is node i's solution estimate from the global average. Introduce also vector $\widetilde{\mathbf{x}}(k) = (\,\widetilde{\mathbf{x}}_1(k), ..., \widetilde{\mathbf{x}}_N(k)\,)^\top$, and note that it can be represented as $\widetilde{\mathbf{x}}(k) = (\mathbf{I} - \mathbf{J})\,\mathbf{x}(k)$, where we recall $\mathbf{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. We have the following Lemma.

Lemma 3.4. *Consider algorithm* (2) *under Assumptions 1–3. Then, there holds:*

$$\mathbb{E}[\,\|\widetilde{\mathbf{x}}(k)\|^2\,] = O(1/k).$$

As detailed in the next Subsection, Lemma 3.4 is important as it allows to sufficiently tightly bound the bias in the gradient estimates according to which the global average $\overline{\mathbf{x}}(k)$ evolves.

Proof. It is easy to show that the process $\{\widetilde{\mathbf{x}}(k)\}$ follows the recursion:

$$\widetilde{\mathbf{x}}(k+1) = \widetilde{\mathbf{W}}(k)\widetilde{\mathbf{x}}(k) - \alpha_k\,(\mathbf{I} - \mathbf{J})\,\underbrace{(\nabla F(\mathbf{x}(k)) + \mathbf{v}(k))}_{\mathbf{w}(k)}, \tag{17}$$

where $\widetilde{\mathbf{W}}(k) = \mathbf{W}(k) - \mathbf{J} = \mathbf{I} - \mathbf{L}(k) - \mathbf{J}$. Note that, $\mathbb{E}\left[\|\mathbf{w}(k)\|^2\right] \le c_7 < \infty$, which follows due to the mean square boundedness of $\mathbf{x}(k)$ and $\nabla F(\mathbf{x}(k))$. Then, we have:

$$\|\widetilde{\mathbf{x}}(k+1)\| \le \left\|\widetilde{\mathbf{W}}(k)\right\|\,\|\widetilde{\mathbf{x}}(k)\| + \alpha_k\,\|\mathbf{w}(k)\|\,.$$

We now invoke Lemma 4.4 in [21] to note that, after an appropriately chosen k_1, we have for $\forall k \ge k_1$,

$$\|\widetilde{\mathbf{x}}(k+1)\| \le (1 - r(k))\,\|\widetilde{\mathbf{x}}(k)\| + \alpha_k\,\|\mathbf{w}(k)\|\,, \tag{18}$$

with $r(k)$ being a $\mathcal{F}_k$-adapted process that satisfies $r(k) \in [0, 1]$, a.s., and:

$$\mathbb{E}\left[r(k)|\mathcal{F}_k\right] \ge c_8 \beta_k = \frac{c_9'}{(k+1)^{\frac{1}{2}}}\ a.s., \tag{19}$$

for some constants $c_8, c_9' > 0$. Using (14) in (18), it can be shown that:

$$\begin{aligned} & \|\widetilde{\mathbf{x}}(k+1)\|^2 \le (1+\theta_k)\,(1 - r(k))^2\,\|\widetilde{\mathbf{x}}(k)\|^2 \\ + \quad & \left(1 + \frac{1}{\theta_k}\right)\alpha_k^2\,\|\mathbf{w}(k)\|^2\,, \end{aligned}$$

for $\theta_k = \frac{c_{10}}{(k+1)^{\frac{1}{2}}}$. Further, it can be shown tha:

$$\begin{aligned} & \mathbb{E}\left[\|\widetilde{\mathbf{x}}(k+1)\|^2\,|\mathcal{F}_k\right] \le (1+\theta_k)\left(1 - \frac{c_9}{(k+1)^{\frac{1}{2}}}\right)^2 \|\widetilde{\mathbf{x}}(k)\|^2 \\ & + \left(1 + \frac{1}{\theta_k}\right)\alpha_k^2\,\mathbb{E}[\,\|\mathbf{w}(k)\|^2\,|\,\mathcal{F}_k\,],\ \text{a.s.,} \end{aligned}$$

for appropriately chosen positive constant c_9. Next, for $c_{10} < c_9$ (c_{10} can be chosen freely), we have:

$$\begin{aligned} & \mathbb{E}\left[\|\widetilde{\mathbf{x}}(k+1)\|^2\right] \le \left(1 - \frac{c_{11}}{(k+1)^{\frac{1}{2}}}\right)\mathbb{E}\left[\|\widetilde{\mathbf{x}}(k)\|^2\right] \\ & + \frac{c_{12}}{(k+1)^{\frac{3}{2}}}, \end{aligned} \tag{20}$$

where c_{11} and c_{12} are appropriately chosen positive constants. Utilizing Lemma 3.2, inequality (20) finally yields $\mathbb{E}\left[\|\widetilde{\mathbf{x}}(k+1)\|^2\right] = O\left(\frac{1}{k}\right)$. The proof of the Lemma is complete.

D. Proof of Theorem 3.1

We are now ready to prove Theorem 3.1.

Proof.

Consider global average $\overline{\mathbf{x}}(k) = \frac{1}{N}\sum_{n=1}\mathbf{x}_i(k)$. From (17), we have:

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{x}}(k) - \alpha_k \left[\frac{1}{N}\sum_{i=1}^{N} \nabla f_i\left(\mathbf{x}_i(k)\right) + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i(k)}_{\overline{\mathbf{v}}(k)} \right]$$

which implies:

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{x}}(k) - \frac{\alpha_k}{N}\left[\sum_{i=1}^{N} \nabla f_i\left(\mathbf{x}_i(k)\right)\right.$$
$$\left. - \nabla f_i\left(\overline{\mathbf{x}}(k)\right) + \nabla f_i\left(\overline{\mathbf{x}}(k)\right)\right] - \alpha_k \overline{\mathbf{v}}(k).$$

Recall $f(\cdot) = \sum_{i=1}^{N} f_i(\cdot)$. Then, we have:

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{x}}(k) - \frac{\alpha_k}{N}\nabla f\left(\overline{\mathbf{x}}(k)\right) \tag{21}$$
$$- \frac{\alpha_k}{N}\left[\sum_{i=1}^{N} \nabla f_i\left(\mathbf{x}_i(k)\right) - \nabla f_i\left(\overline{\mathbf{x}}(k)\right)\right] - \alpha_k \overline{\mathbf{v}}(k),$$

which implies:

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{x}}(k) \tag{22}$$
$$- \frac{\alpha_k}{N}\left[\nabla f\left(\overline{\mathbf{x}}(k)\right) + \mathbf{e}(k)\right],$$

where

$$\mathbf{e}(k) = N\overline{\mathbf{v}}(k) + \underbrace{\sum_{i=1}^{N}\left(\nabla f_i\left(\mathbf{x}_i(k)\right) - \nabla f_i\left(\overline{\mathbf{x}}(k)\right)\right)}_{\boldsymbol{\epsilon}(k)}. \tag{23}$$

Note that, $\left\|\nabla f_i\left(\mathbf{x}_i(k)\right) - \nabla f_i\left(\overline{\mathbf{x}}(k)\right)\right\| \leq L\left\|\mathbf{x}_i(k) - \overline{\mathbf{x}}(k)\right\| = L\left\|\widetilde{\mathbf{x}}_i(k)\right\|$. Thus, we can conclude for

$$\boldsymbol{\epsilon}(k) = \sum_{i=1}^{N}\left(\nabla f_i\left(\mathbf{x}_i(k)\right) - \nabla f_i\left(\overline{\mathbf{x}}(k)\right)\right)$$

the following:

$$\mathbb{E}\left[\left\|\boldsymbol{\epsilon}(k)\right\|^2\right] \leq \frac{c_{15}}{(k+1)}, \tag{24}$$

where c_{15} is a positive constant. Note here that (22) is an inexact gradient method for minimizing f with step size α_k/N and the random gradient error $\mathbf{e}(k) = N\overline{\mathbf{v}}(k) + \boldsymbol{\epsilon}(k)$. The term $N\overline{\mathbf{v}}(k)$ is zero-mean, while the gradient estimate bias is induced by $\boldsymbol{\epsilon}(k)$; as per (24), the bias is at most $O(1/k)$ in the mean square sense.

With the above development in place, we rewrite (21) as follows:

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{x}}(k) - \frac{\alpha_k}{N}\nabla f\left(\overline{\mathbf{x}}(k)\right) - \frac{\alpha_k}{N}\boldsymbol{\epsilon}(k) - \alpha_k\overline{\mathbf{v}}(k). \tag{25}$$

This implies (recall that $\mathbf{x}^\star$ is the solution to (1)):

$$\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star = \overline{\mathbf{x}}(k) - \mathbf{x}^\star \tag{26}$$
$$- \frac{\alpha_k}{N}\left[\nabla f\left(\overline{\mathbf{x}}(k)\right) - \underbrace{\nabla f\left(\mathbf{x}^\star\right)}_{=0}\right] - \frac{\alpha_k}{N}\boldsymbol{\epsilon}(k) - \alpha_k\overline{\mathbf{v}}(k). \tag{27}$$

By the mean value theorem, we have:

$$\nabla f\left(\overline{\mathbf{x}}(k)\right) - \nabla f\left(\mathbf{x}^\star\right) \tag{28}$$
$$= \underbrace{\left[\int_{s=0}^{1} \nabla^2 f\left(\mathbf{x}^\star + s\left(\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right)\right)\right] d\,s}_{\overline{\mathbf{H}}_k}$$
$$\times\left(\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right), \tag{29}$$

where it is to be noted that $NL\mathbf{I} \succcurlyeq \overline{\mathbf{H}}_k \succcurlyeq N\mu\mathbf{I}$. Using (29) in (25), we have:

$$\left(\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right) = \left[\mathbf{I} - \frac{\alpha_k}{N}\overline{\mathbf{H}}_k\right]\left(\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right) \tag{30}$$
$$- \frac{\alpha_k}{N}\boldsymbol{\epsilon}(k) - \alpha_k\overline{\mathbf{v}}(k).$$

Denote by $\mathbf{m}(k) = \left[\mathbf{I} - \frac{\alpha_k}{N}\overline{\mathbf{H}}_k\right]\left(\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right) - \frac{\alpha_k}{N}\boldsymbol{\epsilon}(k)$. Then, (30) is rewritten as:

$$\left(\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right) = \mathbf{m}(k) - \alpha_k\overline{\mathbf{v}}(k), \tag{31}$$

and so:

$$\left\|\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right\|^2 \leq \left\|\mathbf{m}(k)\right\|^2 - 2\alpha_k\mathbf{m}(k)^\top\overline{\mathbf{v}}(k)$$
$$+ \alpha_k^2\left\|\overline{\mathbf{v}}(k)\right\|^2.$$

The latter inequality implies:

$$\mathbb{E}[\,\left\|\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right\|^2 \,|\, \mathcal{F}_k\,] \leq \left\|\mathbf{m}(k)\right\|^2$$
$$- 2\alpha_k\mathbf{m}(k)^\top\mathbb{E}[\,\overline{\mathbf{v}}(k)\,|\,\mathcal{F}_k\,] + \alpha_k^2\mathbb{E}[\,\left\|\overline{\mathbf{v}}(k)\right\|^2 \,|\, \mathcal{F}_k],\ \text{a.s.}$$

Taking expectation, using the fact that $\mathbb{E}[\,\overline{\mathbf{v}}(k)\,|\,\mathcal{F}_k\,] = 0$, Assumption 3, and Lemma 3.3, we obtain:

$$\mathbb{E}\left[\left\|\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right\|^2\right] \leq \mathbb{E}\left[\left\|\mathbf{m}(k)\right\|^2\right] + \frac{c_{17}}{(k+1)^2}, \tag{32}$$

for some constant $c_{17} > 0$. Next, using (14), we have for $\mathbf{m}(k)$ the following:

$$\left\|\mathbf{m}(k)\right\|^2 \leq (1+\theta_k)\left\|\mathbf{I} - \frac{\alpha_k}{N}\overline{\mathbf{H}}_k\right\|^2\left\|\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right\|^2$$
$$+ \left(1 + \frac{1}{\theta_k}\right)\frac{\alpha_k^2}{N^2}\left\|\boldsymbol{\epsilon}(k)\right\|^2$$
$$\leq (1+\theta_k)\left(1 - c_{18}\alpha_k\right)^2\left\|\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right\|^2$$
$$+ \left(1 + \frac{1}{\theta_k}\right)\frac{\alpha_k^2}{N^2}\left\|\boldsymbol{\epsilon}(k)\right\|^2,$$

with $c_{18} = \mu/N$, because $\mu\,\mathbf{I} \preceq \overline{\mathbf{H}}_k \preceq L\,\mathbf{I}$. After choosing $\theta_k = \frac{c_{19}}{(k+1)}$ such that $c_{19} < \alpha_0\,c_{18}/2 = \alpha_0\,\mu/(2N)$ and after taking expectation, we obtain:

$$\mathbb{E}[\,\left\|\mathbf{m}(k)\right\|^2\,] \leq \left(1 - \frac{c_{20}}{k+1}\right)\mathbb{E}[\,\left\|\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right\|^2\,] + \frac{c_{21}}{(k+1)^2}, \tag{33}$$

where $c_{20} > \alpha_0\,\mu/(2N) > 1$ (because $\alpha_0 > 2N/\mu$) and c_{21} is a positive constant. Combining (33) and (32), we get:

$$\mathbb{E}\left[\left\|\overline{\mathbf{x}}(k+1) - \mathbf{x}^\star\right\|^2\right] \leq \left(1 - \frac{c_{20}}{k+1}\right)\left\|\overline{\mathbf{x}}(k) - \mathbf{x}^\star\right\|^2$$
$$+ \frac{c_{21}}{(k+1)^2} + \frac{c_{17}}{(k+1)^2}.$$

Invoking Lemma 3.2, the latter inequality implies:

$$\mathbb{E}\left[\|\overline{\mathbf{x}}(k+1)-\mathbf{x}^\star\|^2\right] \le \frac{c_{22}}{(k+1)}, \tag{34}$$

for some constant $c_{22} > 0$. Therefore, for the global average $\overline{\mathbf{x}}(k)$, we have obtained the mean square rate $O\left(\frac{1}{k}\right)$. Finally, we note that,

$$\|\mathbf{x}_i(k)-\mathbf{x}^\star\| \le \|\overline{\mathbf{x}}(k)-\mathbf{x}^\star\| + \left\|\underbrace{\mathbf{x}_i(k)-\overline{\mathbf{x}}(k)}_{\widetilde{\mathbf{x}}_i(k)}\right\|.$$

After using:

$$\|\mathbf{x}_i(k)-\mathbf{x}^\star\|^2 \le 2\,\|\widetilde{\mathbf{x}}_i(k)\|^2 + 2\,\|\overline{\mathbf{x}}(k)-\mathbf{x}^\star\|^2,$$

and taking expectation, it follows that $\mathbb{E}\left[\|\mathbf{x}_i(k)-\mathbf{x}^\star\|^2\right] = O\left(\frac{1}{k}\right)$, for all $i = 1, ..., N$. The proof is complete.

4. Simulation example

This section presents a numerical example on ℓ_2-regularized logistic losses. We consider random networks where links fail independently over iterations and across different links, with probability p_{fail}. The simulation corroborates the derived $O(1/k)$ rate of algorithm (2) over random networks and shows that deterioration in the algorithm performance due to increase of p_{fail} is small.

We consider empirical risk minimization (7) with the logistic loss in (8) and the regularization functions set to $\Psi_i(\mathbf{x}) = \frac{\kappa}{2}\|\mathbf{x}\|^2$, $i = 1, ..., N$, where $\kappa > 0$ is the regularization parameter that is set to $\kappa = 0.5$.

The number of data points per node is $n_i = 10$. We generate the "true" classification vector $x' = ((\mathbf{x}_1')^\top, x_0')^\top$ by drawing its entries independently from standard normal distribution. Then, the class labels are generated as $b_{ij} = \text{sign}\left((\mathbf{x}_1')^\top \mathbf{a}_{i,j} + x_0' + \epsilon_{ij}\right)$, where ϵ_{ij}'s are drawn independently from normal distribution with zero mean and standard deviation 2. The feature vectors $\mathbf{a}_{i,j}$, $j = 1, ..., n_i$, at node i are generated as follows: each entry of each vector is a sum of a standard normal random variable and a uniform random variable with support $[0,\,5\,i]$. Different entries within a feature vector are drawn independently, and also different vectors are drawn independently, both intra node and inter nodes. Note that the feature vectors at different nodes are drawn from different distributions.

The algorithm parameters are set as follows. We let $\beta_k = \frac{1}{\theta\,(k+1)^{1/2}}$, $\alpha_k = \frac{1}{k+1}$, $k = 0, 1, ...$ Here, θ is the maximal degree across all nodes in the network and here equals $\theta = 6$. Algorithm (2) is initialized with $\mathbf{x}_i(0) = 0$, for all $i = 1, ..., N$.

We consider a connected network $\mathcal{G}$ with $N = 10$ nodes and 23 links, generated as a random geometric graph: nodes are placed randomly (uniformly) on a unit square, and the node pairs whose distance is less than a radius are connected by an edge. We consider the random network model where each (undirected) link in network $\mathcal{G}$ fails independently across iterations and independently from other links with probability p_{fail}. We consider the cases $p_{\text{fail}} \in \{0;\,0.5;\,0.9\}$. Note that the case $p_{\text{fail}} = 0$ corresponds to network $\mathcal{G}$ with all its links always online, more precisely, with links failing with zero probability. Algorithm (2) is then run on each of the described network models, i.e., for each $p_{\text{fail}} \in \{0;\,0.5;\,0.9\}$. This allows us to assess how much the algorithm performance degrades with the increase of p_{fail}. We also include a comparison with the following centralized stochastic gradient method:

$$\mathbf{y}(k+1) = \mathbf{y}(k) - \frac{1}{N(k+1)}\sum_{i=1}^{N} \nabla \ell\left(\,\mathbf{y}(k);\,\mathbf{a}_i(k), b_i(k)\,\right), \tag{35}$$

where $(\mathbf{a}_i(k), b_i(k))$ is drawn uniformly from the set $(\mathbf{a}_{i,j}, b_{i,j})$, $j = 1, ..., n_i$. Note that algorithm (35) makes an unbiased estimate of $\sum_{i=1}^N \nabla f_i(\mathbf{y}(k))$ by drawing a sample uniformly at random from each node's data set. Algorithm (35) is an idealization of (2): it shows how (2) would be implemented if there existed a fusion node that had access to all nodes' data. Hence, the comparison with (35) allows us to examine how much the performance of (2) degrades due to lack of global information, i.e., due to the distributed nature of the considered problem. Note that the step-size in (35) is set to $1/N(k+1)$ for a meaningful comparison with (2), as this is the step-size effectively utilized by the hypothetical global average of the nodes' iterates with (2). As an error metric, we use the mean square error (MSE) estimate averaged across nodes: $\frac{1}{N}\sum_{i=1}^N \|\mathbf{x}_i(k) - \mathbf{x}^\star\|^2$.

In Figure 1, we plot the estimated MSE, averaged across 100 algorithm runs, versus iteration number k in a logarithmic scale. The Figure considers different values of parameter p_{fail}. Note that here the slope of the plot curve corresponds to the sublinear rate of the method; e.g., the -1 slope corresponds to a $1/k$ rate. First, note from the Figure that, for any value of p_{fail}, algorithm (2) achieves on this example (at least) the $1/k$ rate, thus corroborating our theory. Next, note that the increase of the link failure probability only increases the constant in the MSE but does not affect the rate. Interestingly, the loss due to the increase of p_{fail} on this example is small. Figure 1 also shows the performance of the centralized method (35). We can see that the distributed method (2) follows closely the centralized method, except at the few initial iterations.

5. Conclusion

We considered a distributed stochastic gradient method for smooth strongly convex optimization. Through the analysis of the considered method, we established for the first time the order optimal $O(1/k)$ MSE convergence rate for the assumed optimization setting when the underlying network is randomly varying and the noise distribution has unbounded support. Simulation example on ℓ_2-regularized logistic losses corroborates the established results.

References

[1] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing, Signal*

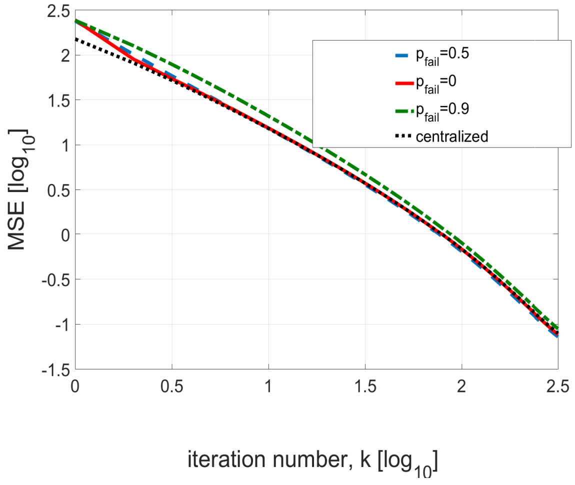

Fig. 1: Estimated MSE versus iteration number k for algorithm (2) with link failure probability $p_{\rm fail} = 0$ (red, solid line); 0.5 (blue, dashed line); and 0.9 (green, dash-dot line). The Figure also shows the performance of the centralized stochastic gradient method in (35) (black, dotted line).

Processing in Gossiping Algorithms Design and Applications, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[2] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: A mathematical approach to motion coordination algorithms*. Princeton University Press, 209.

[3] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari, “Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3914–3929, 2015.

[4] I. Lobel and A. E. Ozdaglar, “Distributed subgradient methods for convex optimization over random networks,” *IEEE Trans. Automat. Contr.*, vol. 56, no. 6, pp. 1291–1306, Jan. 2011.

[5] I. Lobel, A. Ozdaglar, and D. Feijer, “Distributed multi-agent optimization with state-dependent communication,” *Mathematical Programming*, vol. 129, no. 2, pp. 255–284, 2011.

[6] D. Jakovetic, J. Xavier, and J. M. F. Moura, “Convergence rates of distributed Nesterov-like gradient methods on random networks,” *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 868–882, February 2014.

[7] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” *Journal of Machine Learning Rsearch*, vol. 14, p. 567599, 2013.

[8] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, p. 772790, Aug. 2011.

[9] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks,” *IEEE Sig. Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[10] M. Huang, S. Dey, G. N.Nair, and J. H. Manton, “Stochastic consensus over noisy networks with markovian and arbitrary switches,” *Automatica*, vol. 46, no. 10, pp. 1571–1583, Oct. 2010.

[11] 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, 2007.

[12] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise,” *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[13] K. Tsianos and M. Rabbat, “Distributed strongly convex optimization,” *50th Annual Allerton Conference onCommunication, Control, and Computing*, Oct. 2012.

[14] Z. J. Towfic, J. Chen, and A. H. Sayed, “Excess-risk of distributed stochastic learners,” *IEEE Transactions on Information Theory*, vol. 62, no. 10, Oct. 2016.

[15] D. Yuan, Y. Hong, D. W. C. Ho, and G. Jiang, “Optimal distributed stochastic mirror descent for strongly convex optimization,” *Automatica*, vol. 90, pp. 196–203, April 2018.

[16] N. D. Vanli, M. O. Sayin, and S. S. Kozat, “Stochastic subgradient algorithms for strongly convex optimization over distributed networks,” *IEEE Transactions on network science and engineering*, vol. 4, no. 4, pp. 248–260, Oct.-Dec. 2017.

[17] A. Nedic and A. Olshevsky, “Stochastic gradient-push for strongly convex functions on time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[18] K. Tsianos and M. Rabbat, “Stochastic proximal gradient consensus over random networks,” *IEEE Transactions on Signal Processing*, vol. 65, no. 11, June 2017.

[19] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, “Distributed zeroth order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach,” 2018, available at https://www.dropbox.com/s/kfc2hgbfcx5yhr8/MainCDC2018KWSA.pdf.

[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, June 2006.

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