2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Harnessing Smoothness to Accelerate Distributed Optimization

Guannan Qu, Na Li

Abstract— **There has been a growing effort in studying the distributed optimization problem over a network. The objective is to optimize a global function formed by a sum of local functions, using only local computation and communication. Literature has developed consensus-based distributed (sub)gradient descent (DGD) methods and has shown that they have the same convergence rate $O(\frac{\log t}{\sqrt{t}})$ as the centralized (sub)gradient methods (CGD) when the function is convex but possibly nonsmooth. However, when the function is convex and smooth, under the framework of DGD, it is unclear how to harness the smoothness to obtain a faster convergence rate comparable to CGD's convergence rate. In this paper, we propose a distributed algorithm that, despite using the same amount of communication per iteration as DGD, can effectively harnesses the function smoothness and converge to the optimum with a rate of $O(\frac{1}{t})$. If the objective function is further strongly convex, our algorithm has a linear convergence rate. Both rates match the convergence rate of CGD. The key step in our algorithm is a novel gradient estimation scheme that uses history information to achieve fast and accurate estimation of the average gradient. To motivate the necessity of history information, we also show that it is impossible for a class of distributed algorithms like DGD to achieve a linear convergence rate without using history information even if the objective function is strongly convex and smooth.**

I. Introduction

Given a set of agents $\mathcal{N} = \{1, 2, \ldots, n\}$, each of which has a local convex cost function $f_i(x) : \mathbb{R}^N \to \mathbb{R}$, the objective of distributed optimization is to find x that minimizes the average of all the functions,

$$\min_{x \in \mathbb{R}^N} f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

using local communication and local computation. The local communication is defined through an undirected communication graph $\mathcal{G} = (V, E)$, where the nodes $V = \mathcal{N}$ and edges $E \subset V \times V$. Agent i and j can send information to each other if and only if i and j are connected in graph $\mathcal{G}$. The local computation means that each agent can only make his decision based on the local function f_i and the information obtained from his neighbors.

This problem has recently received much attention and has found various applications in multi-agent control, distributed state estimation over sensor networks, large scale computation in machine/statistical learning, etc [1]–[3]. As a concrete example, in the setting of distributed statistical learning, x is the parameter to infer, and f_i is the empirical loss function of the local dataset of agent i. Then minimizing f means empirical loss minimization that uses datasets of all the agents.

Guannan Qu and Na Li are affiliated with John A. Paulson School of Engineering and Applied Sciences at Harvard University. Email: gqu@g.harvard.edu, nali@seas.harvard.edu. This work is supported under NSF ECCS 1608509 and NSF CAREER 1553407.

The early work of this problem can be found in [4], [5]. Recently, [6] (see also [7]) proposes a consensus-based distributed (sub)gradient descent (DGD) method where each agent performs a consensus step and then a descent step along the local (sub)gradient direction of f_i. [8] applies a similar idea to develop a distributed dual averaging algorithm. Extensions of these work have been proposed that deal with various realistic conditions, such as stochastic subgradient errors [9], directed or random communication graph [10]–[12], linear scaling in network size [13], heterogeneous local constraints [14], [15]. Overall speaking, these DGD (or DGD-like) algorithms are designated for nonsmooth functions and they achieve the same convergence speed $O(\frac{\log t}{\sqrt{t}})$ [16] as centralized subgradient descent. They can also be applied to smooth functions, but when doing so it either does not guarantee exact convergence when using a fix constant step size [12], [17], or has a convergence rate at most $\Omega(\frac{1}{t^{2/3}})$ when using a diminishing step size [18], slower than the normal Centralized Gradient Descent (CGD) method's $O(\frac{1}{t})$ [19]. Therefore, DGD does not fully exploit the function smoothness and has a slower convergence rate compared with CGD. In fact, we prove in this paper that for strongly convex and smooth functions, it is impossible for DGD-like algorithms to achieve the same linear convergence rate as CGD (Theorem 4). Alternatively, [18], [20] suggest that it is possible to achieve faster convergence for smooth functions, by performing multiple consensus steps after each gradient evaluation. However, it places a larger communication burden. These drawbacks poses the need for distributed algorithms that effectively harness the smoothness to achieves faster convergence, using only *one* communication step per gradient evaluation iteration.

In this paper, we propose a distributed algorithm that can effectively harness the smoothness, and achieve a convergence rate that matches CGD, using only one communication step per gradient evaluation. Specifically, our algorithm achieves a $O(\frac{1}{t})$ rate for smooth convex functions (Theorem 3), and a linear convergence rate ($O(\gamma^t)$ for some $\gamma \in (0, 1)$) for smooth and strongly convex functions (Theorem 1).[1] The convergence rates match the convergence rates of CGD, but with worse constants due to the distributed nature of the problem. Our algorithm is a combination of gradient descent and a novel gradient estimation scheme that utilizes history information to achieve fast and accurate estimation of the average gradient. To show the necessity of history

[1] A recent paper [21] also achieves similar convergence rate results using a different algorithm. However, to the best of out knowledge, our algorithm is the first to theoretically achieve the $O(\frac{1}{t})$ convergence rate for convex smooth functions in terms of objective error. [21] achieves a $O(\frac{1}{t})$ rate in terms of the first order residual instead. In addition, a detailed comparison between our algorithm and [21] will be given in Section III-C.

978-1-5090-1837-6/16/$31.00 ©2016 IEEE

information, we also prove that it is impossible for a class of distributed algorithms like DGD to achieve a linear convergence rate without using history information even if we restrict the class of objective functions to be strongly convex and smooth (Theorem 4).

Moreover, our scheme can be cast as a general method for decentralizing many first-order optimization algorithms, like Nesterov gradient descent [19]. We expect the distributed algorithm will have a similar convergence rate as its centralized counterpart. Some preliminary results on applying the scheme to Nesterov gradient descent can be found in our recent work [22].

The rest of the paper is organized as follows. Section II formally defines the problem and presents our algorithm and results. Section III reviews previous methods, introduces an impossibility result and motivates our approach. Section IV proves the convergence of our algorithm. Lastly, Section V provides numerical simulations and VI concludes the paper.

Notations. Throughout the rest of the paper, n is the number of agents, and N is the dimension of the domain of the f_i's. $i, j \in \{1, 2, \ldots, n\}$ are indices for agents, while $t, k, \ell \in \mathbb{N}$ are indices for iteration steps. We use x^* and f^* to denote the minimizer and the minimal value of f, respectively. If f has multiple minimizers, x^* can be any of them. $\|\cdot\|$ denotes 2-norm for vectors, and Frobenius norm for matrices. $\langle \cdot, \cdot \rangle$ denotes inner product for vectors. $\rho(\cdot)$ denotes spectral radius for square matrices, and $\mathbf{1}$ denotes a n-dimensional all one column vector. All vectors, when having dimension N (the dimension of the domain of the f_i's), will all be regarded as row vectors. As a special case, all gradients, $\nabla f_i(x)$ and $\nabla f(x)$ are interpreted as N-dimensional row vectors. '$\leq$', when applied to vectors of the same dimension, denotes element wise 'less than or equal to'.

II. Problem and Algorithm

A. Problem Formulation

Consider n agents, $\mathcal{N} = \{1, 2, \ldots, n\}$, each of which has a convex function $f_i : \mathbb{R}^N \to \mathbb{R}$. The objective of distributed optimization is to find x to minimize the average of all the functions, i.e.

$$\min_{x \in \mathbb{R}^N} f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x) \tag{1}$$

using local communication and local computation. The local communication is defined through an undirected communication graph $\mathcal{G} = (V, E)$, where the nodes $V = \mathcal{N}$ and edges $E \subset V \times V$. Agent i and j can send information to each other if and only if $(i, j) \in E$. The local computation means that each agent can only make his decision based on the local function f_i and the information obtained from his neighbors.

Throughout the paper, we assume that the set of minimizers of f is non-empty and compact. We denote x^* as one of the minimizers and f^* as the minimal value. We will study the case where each f_i is convex and β-smooth (Assumption 1) and also the case where each f_i is in addition α-strongly convex (Assumption 2).

Assumption 1: $\forall i, f_i$ is convex. In addition, f_i is β-smooth, that is, f_i is differentiable and the gradient is β-Lipschitz continuous, i.e., $\forall x, y \in \mathbb{R}^N$,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq \beta \|x - y\|$$

As a direct consequence, f is also β-smooth.

Assumption 2: $\forall i$, f_i is α-strongly convex, i.e. $\forall x, y \in \mathbb{R}^N$, we have

$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2$$

As a result, f is also α-strongly convex.

B. Algorithm

The algorithm we will describe is a consensus-based distributed algorithm. Each agent weighs their neighbors' information to compute their local decisions. To model the weighting process, we introduce a consensus weight matrix, $W = [w_{ij}] \in \mathbb{R}^{n \times n}$, which satisfies the following properties:[2]

(a) $\forall (i, j) \in E$, $w_{ij} > 0$. $\forall i$, $w_{ii} > 0$. $w_{ij} = 0$ elsewhere.
(b) W is doubly stochastic, i.e. $\sum_{i'} w_{i'j} = \sum_{j'} w_{ij'} = 1$ for all $i, j \in \mathcal{N}$.

As a result, $\exists \sigma \in (0, 1)$ which depends on the spectrum of W, such that for any $\omega \in \mathbb{R}^{n \times 1}$, we have $\|W^t \omega - \mathbf{1}\bar{\omega}\| \leq \sigma^t \|\omega - \mathbf{1}\bar{\omega}\|$ where $\bar{\omega} = \frac{1}{n}\mathbf{1}^T \omega$ (the average of entries in ω) [24]. This 'averaging' property will be frequently used in the rest of the paper.

In our algorithm, each agent i keeps an estimate of the minimizer $x_i(t) \in \mathbb{R}^{1 \times N}$, and another vector $s_i(t) \in \mathbb{R}^{1 \times N}$ which is designated to estimate the average gradient, $\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_i(t))$. The algorithm starts with an arbitrary $x_i(0)$, and with $s_i(0) = \nabla f_i(x_i(0))$. The algorithm proceeds using the following update,

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t) - \eta s_i(t) \tag{2}$$

$$s_i(t+1) = \sum_{j=1}^{n} w_{ij} s_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t)) \tag{3}$$

where $[w_{ij}]_{n \times n}$ are the consensus weights and $\eta > 0$ is a fixed step size. Because $w_{ij} = 0$ when $(i, j) \notin E$, each node i only needs to send $x_i(t)$ and $s_i(t)$ to its neighbors. Therefore, the algorithm can be operated in a fully distributed fashion, with only local communication. Note that the two consensus weight matrices in step (2) and (3) can be chosen differently. We use the same matrix W to carry out our analysis for the purpose of easy exposition.

The update equation (2) is similar to the algorithm in [6] (see also (4) in Section III), except that the subgradient is replaced with $s_i(t)$ which follows the update rule (3). In Section III and IV-B, we will discuss the motivation and the intuition behind this algorithm.

Remark 1: The key of our algorithm is the gradient estimation scheme (3) and it can be used to decentralize

[2]The selection of the consensus weights is an intensely studied problem, see [23], [24].

many other gradient-based algorithms. For example, suppose a centralized algorithm is in the following form,

$$x(t+1) = \mathcal{F}_t(x(t), \nabla f(x(t))$$

where $x(t)$ is the state, $\mathcal{F}_t$ is the update equation. We can write down a distributed algorithm as

$$x_i(t+1) = \mathcal{F}_t(\sum_j w_{ij} x_j(t), s_i(t))$$

$$s_i(t+1) = \sum_j w_{ij} s_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t))$$

Our conjecture is that for a broad range of centralized algorithms, the distributed algorithm obtained as above will have a similar convergence rate as the centralized counterpart. Our ongoing work includes applying the above scheme to other centralized algorithms like Nesterov gradient method. Some of preliminary results are in [22].

C. Convergence of the Algorithm

To state the convergence results, we need to define the following average sequences.

$$\bar{x}(t) = \frac{1}{n}\sum_{i=1}^n x_i(t) \in \mathbb{R}^{1\times N}, \bar{s}(t) = \frac{1}{n}\sum_{i=1}^n s_i(t) \in \mathbb{R}^{1\times N}$$

$$g(t) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i(t)) \in \mathbb{R}^{1\times N}$$

We also define the gradient of f evaluated at $\bar{x}(t)$,

$$h(t) = \nabla f(\bar{x}(t)) \in \mathbb{R}^{1\times N}$$

We summarize our convergence results here.

Theorem 1: Under the smooth and strongly convex assumptions (Assumption 1 and 2), when η is such that the matrix

$$G(\eta) = \begin{bmatrix} (\sigma+\beta\eta) & \beta(\eta\beta+2) & \eta\beta^2 \\ \eta & \sigma & 0 \\ 0 & \eta\beta & \lambda \end{bmatrix}$$

where $\lambda = \max(|1-\alpha\eta|, |1-\beta\eta|)$

has spectral radius $\rho(G(\eta)) < 1$, then $\forall i$, $\|x_i(t) - x^*\|$ (distance to the optimizer), $\|x_i(t) - \bar{x}(t)\|$ (consensus error), and $\|s_i(t) - g(t)\|$ (gradient estimation error) are all decaying with rate $O(\rho(G)^t)$. As a consequence, $f(x_i(t)) - f^*$ (objective error) is decaying with rate $O(\rho(G)^{2t})$.

The following lemma provides a sufficient condition for the step size η to ensure $\rho(G(\eta)) < 1$.

Lemma 2: When $0 < \eta < \eta_0 \triangleq \frac{2}{\beta}\frac{(1-\sigma)^2}{3+\sqrt{13+4\frac{\beta}{\alpha}}}$, $\rho(G(\eta)) < 1$

Remark 2: In experiments we find that $\eta < \frac{1}{3\beta}$ (regardless of α, σ) is usually sufficient for linear convergence. Also, we note that the convergence rate shown in the theorem appears to be conservative compared to numerical experiments.

If we drop the strongly convex assumption, we have the following result.

Theorem 3: Under the smooth assumption (Assumption 1), when η is sufficiently small, we have $f(\bar{x}(t)) - f^* = O(\frac{1}{t})$, and $\forall i$, $\min_{t'\le t} f(x_i(t')) - f^* = O(\frac{1}{t})$

Remark 3: Our algorithm preserves the convergence rate of CGD, in the sense that it has a linear convergence rate when the f_i's are strongly convex and smooth, and a convergence rate of $O(\frac{1}{t})$ when the f_i's are just smooth. However, we note that the linear convergence rate constant $\rho(G)$ is usually worse than CGD; and moreover, in both cases, our algorithm has a worse constant in the big O terms. Moreover, compared to CGD, the choice of step size also depends on the consensus matrix W (Lemma 2).

III. Algorithm development: Motivation

In this section, we will briefly review distributed first-order optimization algorithms that are related to our algorithm and discuss their limitations which motivates our algorithm development. In particular, we will formally provide an impossibility result regarding the limitations. Lastly we will discuss the literature that motivates the idea of harnessing the smoothness from history information.

A. Review of Distributed First-Order Optimization Algorithms

To solve the distributed optimization problem (1), people have developed consensus-based DGD (Distributed (sub)gradient descent) methods, e.g., [6], [8]–[13], [16]–[18], [20], [21], that combine a consensus algorithm and a first order optimization algorithm. For a review of consensus algorithms and first order optimization algorithms, we refer to references [24] and [19], [25], [26] respectively. For the sake of concrete discussion, we focus on the algorithm in [6], where each agent i keeps an local estimate of the solution to (1), $x_i(t)$ and it updates $x_i(t)$ according to,

$$x_i(t+1) = \sum_j w_{ij} x_j(t) - \eta_t g_i(t) \tag{4}$$

where $g_i(t) \in \partial f_i(x_i(t))$ is a subgradient of f_i at $x_i(t)$ (f_i is possibly nonsmooth), and η_t is the step size, and w_{ij} are some properly chosen consensus weights. (4) is essentially performing a consensus step followed by a standard subgradient descent along the local subgradient direction $g_i(t)$. Results in [16] tells that the running best of the objective $f(x_i(t))$ converges to the minimum f^* with rate $O(\frac{\log t}{\sqrt{t}})$ if using a diminishing step size $\eta_t = \Theta(\frac{1}{\sqrt{t}})$. This is the same rate as the centralized subgradient descent algorithm.

When the f_i's are smooth, the subgradient $g_i(t)$ will equal the gradient $\nabla f_i(x_i(t))$. However, as shown in [18], even in this case the convergence rate of (4) can not be better than $\Omega(\frac{1}{t^{2/3}})$. In contrast, the CGD (centralized gradient descent) method,

$$x(t+1) = x(t) - \eta \nabla f(x) \tag{5}$$

converges to the optimum with rate $O(\frac{1}{t})$ if the stepsize η is a small enough constant. Moreover, when f is further strongly convex, CGD (5) converges to the optimal solution with a linear rate. If a fixed step size η is used in DGD (4), though the algorithm runs faster, the method only converges to a neighborhood of the optimizer [12], [17]. This is because even if $x_i(t) = x^*$ (the optimal solution), $\nabla f_i(x_i(t))$ is not necessarily zero.

To fix this problem of non-convergence, it has been proposed to use multiple consensus steps after each gradient descent [18], [20]. One example is provided as follows:

$$\begin{aligned} y_i(t,0) &= x_i(t) - \eta \nabla f_i(x_i(t)) && \text{(6a)} \\ y_i(t,k) &= \sum_j w_{ij} y_j(t,k-1), k = 1,2,\ldots,c_t && \text{(6b)} \\ x_i(t+1) &= y_i(t,c_t) && \text{(6c)} \end{aligned}$$

For each gradient descent step (6a), after c_t consensus steps ($c_t = \Theta(\log t)$ in [18], and $c_t = \Theta(t)$ in [20]), the agents' estimates $x_i(t+1)$ are sufficiently averaged, and it is as if each agent has performed a descent along the average gradient $\frac{1}{n}\sum_i \nabla f_i(x_i(t))$. As a result, algorithm (6) addresses the non-convergence problem mentioned above. However, it places a large communication burden on the agents: the further the algorithm proceeds, the more consensus steps after each gradient step are required. In addition, even if the algorithm already reaches the optimizer $x_i(t) = x^*$, because of (6a) and because $\nabla f_i(x^*)$ might be non-zero, $y_i(t,0)$ will deviate from the optimizer, and then a large number of consensus steps in (6b) are needed to average out the deviation. All these drawbacks pose the need for alternative distributed algorithms that effectively harness the smoothness to achieve faster convergence, using only *one* (or a constant number of) communication step(s) per gradient evaluation.

B. An Impossibility Result

To compliment the preceding discussion, here we provide an impossibility result for a class of distributed first-order algorithms which includes algorithms like (4). We use notation $-i$ to denote the set $\mathcal{N}/\{i\}$. The class of algorithms we consider obeys the following updating rule,

$$x_i(t) = \mathcal{F}(\mathcal{H}(x_i(t-1), x_{-i}(t-1), \mathcal{G}), \eta_t \nabla f_i(x_i(t-1))), \quad \forall i \in \mathcal{N}. \quad (7)$$

Here both $\mathcal{H}$ and $\mathcal{F}$ denote general functions with the following properties. Function $\mathcal{H}$ captures how agents use their neighbors' information, and $\mathcal{H}$ is assumed to be a continuous function on the component $x_j(t)$, $j \in \mathcal{N}$. Note that $\mathcal{H}$ can be interpreted as the consensus step. $\mathcal{F}$ is a function of $\mathcal{H}$ and the scaled gradient direction $\eta_t \nabla f_i(x_i(t-1))$, and $\mathcal{F}$ is assumed to be L-Lipschitz continuous. Note that $\mathcal{F}$ can be interpreted as a first-order update rule, such as the (projected) gradient descent, mirror descent, and some types of proximal algorithms. η_t can be considered as the step size, and we assume it has a limit η^* as $t \to \infty$. We will show that for strongly convex and smooth cost functions, any algorithm belonging to this class will not have a linear convergence rate, which is in contrast to the linear convergence of the centralized methods.

Theorem 4: Consider a simple case where $\mathcal{N} = \{1,2\}$, i.e. there are only two agents. Assume the objective functions $f_1, f_2 : \mathbb{R}^N \to \mathbb{R}$ are α-strongly convex and β-smooth. Suppose for any $f_1, f_2, x_1(0), x_2(0)$, $\lim_{t\to\infty} x_i(t) = x^*$ under algorithm (7), where x^* is the minimizer of $f_1 + f_2$. Then there exist $f_1, f_2, x_1(0), x_2(0)$ such that for any $\delta \in (0,1)$ and $T \geq 0$, there exist $t \geq T$, s.t. $\|x_i(t+1) - x^*\| \geq \delta \|x_i(t) - x^*\|$.

Proof: We first show $\eta^* = 0$. Assume the contrary holds, $\eta^* \neq 0$, then for any objective functions f_1, f_2, and any starting point, we have $x_1(t), x_2(t) \to x^*$, which implies $\mathcal{F}(\mathcal{H}(x_1(t), x_2(t)), \eta_t \nabla f_1(x_1(t))) \to x^*$. By the continuity of $\mathcal{F}$ and $\mathcal{H}$ and ∇f_1, we have $x^* = \mathcal{F}(\mathcal{H}(x^*, x^*), \eta^* \nabla f_1(x^*))$. We can choose f_1, f_2 to be simple quadratic functions such that $(x^*, \nabla f_1(x^*))$ can be any point in $\mathbb{R}^N \times \mathbb{R}^N$. Hence, since $\eta^* \neq 0$, we have, for any $x, y \in \mathbb{R}^N$, $x = \mathcal{F}(\mathcal{H}(x,x), y)$. This is impossible, because if we let the objective functions be $f_1(x) = f_2(x) = \frac{\alpha}{2}\|x\|^2$, and we start from $x_1(0) = x_2(0) \neq 0$, we will have the trajectory $x_i(t)$ stays fixed $x_1(t) = x_2(t) = x_1(0) = x_2(0)$, not converging to the minimizer 0. This is a contradiction. Hence, $\eta^* = 0$.

Now we focus on the case $f_1 = f_2 = f$, and $x_1(0) = x_2(0)$. Then $x_1(t)$ always equals $x_2(t)$ and we define $x(t) \triangleq x_1(t) = x_2(t)$. Let $\tilde{\mathcal{F}}(x,y) = \mathcal{F}(\mathcal{H}(x,x), y)$, then $x(t)$ satisfies $x(t+1) = \tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t)))$. For $\tilde{\mathcal{F}}$, we first show that $x = \tilde{\mathcal{F}}(x, 0)$ for any x. This is because if we consider any x^* and a function $f(x) = \frac{\alpha}{2}\|x - x^*\|^2$ (thus x^* is the minimizer of $f_1 + f_2 = 2f$), then the fact of $x(t) \to x^*$ and $\eta_t \nabla f(x(t)) \to 0$ implies that $x^* = \tilde{\mathcal{F}}(x^*, 0)$ (by the continuity of $\mathcal{F}, \mathcal{H}$).

Now we are ready to prove the Theorem. Notice that for any objective functions $f_1 = f_2 = f$, if we start from $x_1(0) = x_2(0) \neq x^*$ (x^* is the minimizer of both f and $f_1 + f_2$), then the generated sequence $x(t) = x_1(t) = x_2(t)$ satisfies

$$\begin{aligned} \|x(t+1) - x^*\| &= \|\tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t))) - x^*\| \\ &\geq \|\tilde{\mathcal{F}}(x(t), 0) - x^*\| \\ &\quad - \|\tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t))) - \tilde{\mathcal{F}}(x(t), 0)\| \\ &\geq \|x(t) - x^*\| - L\eta_t \|\nabla f(x(t))\| \\ &\geq (1 - \eta_t L\beta)\|x(t) - x^*\| \end{aligned}$$

The theorem follows from the fact that $\eta_t L\beta \to 0$. □

C. Harnessing Smoothness via History Information

Motivated by the previous discussion and the impossibility results, we seek for alternative methods to exploit smoothness to develop faster distributed algorithms. Firstly we note that one major reason for the slow convergence of DGD is the decreasing step size η_t. This motivates us to use a constant step size η in our algorithm (2). But we have discussed that constant η will lead to optimization error due to the fact that $\nabla f_i(x_i(t))$ could be very different from the average gradient $g(t) = \frac{1}{n}\sum_i \nabla f_i(x_i(t))$. However, because of smoothness, $\nabla f_i(x_i(t+1))$ and $\nabla f_i(x_i(t))$ would be close (as well as $g(t+1)$ and $g(t)$) if $x_i(t+1)$ and $x_i(t)$ are close, which is exactly the case when the algorithm is coming close to the minimizer x^*. This motivates the second step of our algorithm (3), using history information to get an accurate estimation of the average gradient $g(t)$ which is a better descent direction than ∇f_i. Similar ideas of using history information trace back to [27], in which the previous gradient is used to narrow down the possible values of the current gradient to save communication complexity for a two-agent optimization problem.

A very recent paper [21] proposes an algorithm that achieves similar convergence results as our algorithm. The algorithm in [21] can be regarded as adding an integration type correction term to (4) while using a fixed step size. This correction term also involves history information in a certain way, which is consistent with our impossibility result. The differences of our algorithm with [21] are summarized below. Firstly, the two consensus matrices in [21] need to be symmetric and also satisfy a predefined spectral relationship, while our algorithm has a looser requirement on the consensus matrices. Secondly, without assuming strongly convex, [21] achieves a $O(\frac{1}{t})$ convergence rate in terms of the optimality residuals, which can be loosely defined as $\|\nabla f(x_i(t))\|^2$ and $\|x_i(t)-\bar{x}(t)\|^2$. Our algorithm not only achieves $O(\frac{1}{t})$ for the optimality residuals, but also achieves $O(\frac{1}{t})$ in terms of the objective error $f(x_i(t))-f^*$, which is a more direct measure of optimality. But one downside of our current results is that [21] gives an explicit step size bound that only depends on β and not on W, whereas our step size bound for the strongly convex case (Lemma 2) depends on W, and we do not have an explicit step size bound for the non-strongly convex case (Theorem 3).

IV. Convergence Analysis

In this section, we prove our main convergence results Theorem 1, Lemma 2, and Theorem 3. Due to the space limit, we only present the main proof steps here. A detailed proof can be found in [28].

A. Analysis Setup

We first stack the $x_i(t)$, $s_i(t)$ and $\nabla f_i(x_i(t))$ in (2) and (3) into matrices. Define $x(t), s(t), \nabla(t) \in \mathbb{R}^{n\times N}$ as,[3]

$$x(t)=\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, s(t)=\begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_n(t) \end{bmatrix}, \nabla(t)=\begin{bmatrix} \nabla f_1(x_1(t)) \\ \nabla f_2(x_2(t)) \\ \vdots \\ \nabla f_n(x_n(t)) \end{bmatrix}$$

We can compactly write the update rule in (2) and (3) as

$$x(t+1) = Wx(t)-\eta s(t) \quad (8a)$$
$$s(t+1) = Ws(t)+\nabla(t+1)-\nabla(t) \quad (8b)$$

and also $s(0)=\nabla(0)$. We start by introducing two straightforward lemmas. Lemma 5 derives update equations that govern the average sequence $\bar{x}(t)$ and $\bar{s}(t)$. Lemma 6 gives inequalities that are direct consequences of smoothness. The proofs can be found in [28].

Lemma 5: The following equalities hold

(a) $\bar{s}(t+1)=\bar{s}(t)+g(t+1)-g(t)=g(t+1)$
(b) $\bar{x}(t+1)=\bar{x}(t)-\eta\bar{s}(t)=\bar{x}(t)-\eta g(t)$

Lemma 6: Under Assumption 1, the following inequalities hold

(a) $\|\nabla(t)-\nabla(t-1)\| \le \beta\|x(t)-x(t-1)\|$
(b) $\|g(t)-g(t-1)\| \le \beta\frac{1}{\sqrt{n}}\|x(t)-x(t-1)\|$
(c) $\|g(t)-h(t)\| \le \beta\frac{1}{\sqrt{n}}\|x(t)-\mathbf{1}\bar{x}(t)\|$

[3]In section II and III, x and $x(t)$ have been used as the centralized decision variable. Here we abuse the use of notation $x(t)$ without causing any confusion.

B. Why the Algorithm Works: An Intuitive Explanation

We provide our intuition that partially explains why the algorithm (8) can achieve a linear convergence rate for strongly convex and smooth functions. In fact we can prove the following statement.

- Assuming $\|s(t)-\mathbf{1}g(t)\|=O(\mu^t)$ for some $\mu\in(0,1)$, then $\|x(t)-\mathbf{1}x^*\|=O(\gamma^t)$ for some $\gamma\in(0,1)$.
- Assuming $\|x(t)-\mathbf{1}x^*\|=O(\gamma^t)$ for some $\gamma\in(0,1)$, then $\|s(t)-\mathbf{1}g(t)\|=O(\mu^t)$ for some $\mu\in(0,1)$

The proof of the above statement is provided in [28].

The above statement tells that the linear decaying rates of the gradient estimation error $\|s(t)-\mathbf{1}g(t)\|$ and the distance to optimizer $\|x(t)-\mathbf{1}x^*\|$ imply one another. Though this circular argument does not prove the linear convergence rate of our algorithm, it illustrates how the algorithm works: the gradient descent step (8a) and the gradient estimation step (8b) facilitate each other to converge fast in a reciprocal manner. This mutual dependence is distinct from many previous methods, where one usually bounds the consensus error at first, and then use the consensus error to bound the objective error. And there is no mutual dependence between the two. In the next two subsections, we will rigorously prove the convergence.

C. Convergence Analysis: Strongly Convex

We start by introducing a lemma that can be found in standard optimization literature, e.g. [25]. The lemma states that if we perform a gradient descent step with a fixed step size for a strongly convex and smooth function, then the distance to optimizer shrinks by at least a fixed ratio.

Lemma 7: $\forall x\in\mathbb{R}^N$, define $x^+=x-\eta\nabla f(x)$ where $0<\eta<\frac{2}{\beta}$, then

$$\|x^+-x^*\| \le \lambda\|x-x^*\|$$

where $\lambda=\max(|1-\eta\alpha|,|1-\eta\beta|)$

Now we are ready to prove Theorem 1.

Proof of Theorem 1: Our strategy is to bound $\|s(k)-\mathbf{1}g(k)\|$, $\|x(k)-\mathbf{1}\bar{x}(k)\|$, and $\|\bar{x}(k)-x^*\|$ in terms of linear combinations of their past values, and in this way obtain a recursive linear vector inequality, which will imply linear convergence.

Step 1: Bound $\|s(k)-\mathbf{1}g(k)\|$. By the update rules (8b),

$$\begin{aligned} s(k)-\mathbf{1}g(k) &= [Ws(k-1)-\mathbf{1}g(k-1)] \\ &\quad +[\nabla(k)-\nabla(k-1)]-[\mathbf{1}g(k)-\mathbf{1}g(k-1)] \end{aligned}$$

Take the norm, and notice that the column-wise average of $s(k-1)$ is just $g(k-1)$ by Lemma 5(a), using the averaging property of the consensus matrix W, we have

$$\begin{aligned} &\|s(k)-\mathbf{1}g(k)\| \\ &\le \|Ws(k-1)-\mathbf{1}g(k-1)\| \\ &\quad +\Big\|[\nabla(k)-\nabla(k-1)]-[\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\| \\ &\le \sigma\|s(k-1)-\mathbf{1}g(k-1)\| \\ &\quad +\Big\|[\nabla(k)-\nabla(k-1)]-[\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\| \quad (9) \end{aligned}$$

It is easy to verify

$$\begin{aligned}&\Big\|[\nabla(k)-\nabla(k-1)]-[\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\|^2\\&=\|\nabla(k)-\nabla(k-1)\|^2-n\|g(k)-g(k-1)\|^2\\&\le\|\nabla(k)-\nabla(k-1)\|^2\end{aligned}$$

Combine this with (9), and also use Lemma 6 (a), we get

$$\begin{aligned}&\|s(k)-\mathbf{1}g(k)\|\\&\le\sigma\|s(k-1)-\mathbf{1}g(k-1)\|+\beta\|x(k)-x(k-1)\|\end{aligned}\quad(10)$$

Step 2: Bound $\|x(k)-\mathbf{1}\bar{x}(k)\|$. Consider update rule (8a) and use Lemma 5(b) and the property of W, we have

$$\begin{aligned}\|x(k)-\mathbf{1}\bar{x}(k)\|\le\sigma\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|\\+\eta\|s(k-1)-\mathbf{1}g(k-1)\|\end{aligned}\quad(11)$$

Step 3: Bound $\|\bar{x}(k)-x^*\|$. Notice by Lemma 5(b), the update rule for $\bar{x}(k)$ is that

$$\bar{x}(k)=\bar{x}(k-1)-\eta h(k-1)-\eta[g(k-1)-h(k-1)]$$

Since the gradient of f at $\bar{x}(k)$ is actually $h(k)$, therefore, by Lemma 7 and Lemma 6(c), we have

$$\begin{aligned}&\|\bar{x}(k)-x^*\|\\&\le\lambda\|\bar{x}(k-1)-x^*\|+\eta\|g(k-1)-h(k-1)\|\\&\le\lambda\|\bar{x}(k-1)-x^*\|+\eta\frac{\beta}{\sqrt{n}}\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|\end{aligned}\quad(12)$$

where $\lambda=\max(|1-\eta\alpha|,|1-\eta\beta|)$.

Step 4: Bound $\|x(k)-x(k-1)\|$. Notice that by smoothness

$$\|h(k-1)\|=\|\nabla f(\bar{x}(k-1))\|\le\beta\|\bar{x}(k-1)-x^*\|$$

Combine the above and Lemma 6(c), we have

$$\begin{aligned}&\|s(k-1)\|\\&\le\|s(k-1)-\mathbf{1}g(k-1)\|\\&\quad+\|\mathbf{1}g(k-1)-\mathbf{1}h(k-1)\|+\|\mathbf{1}h(k-1)\|\\&\le\|s(k-1)-\mathbf{1}g(k-1)\|+\beta\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|\\&\quad+\beta\sqrt{n}\|\bar{x}(k-1)-x^*\|\end{aligned}$$

Hence

$$\begin{aligned}&\|x(k)-x(k-1)\|\\&=\|Wx(k-1)-x(k-1)-\eta s(k-1)\|\\&=\|(W-I)(x(k-1)-\mathbf{1}\bar{x}(k-1))-\eta s(k-1)\|\\&\le2\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|+\eta\|s(k-1)\|\\&\le\eta\|s(k-1)-\mathbf{1}g(k-1)\|\\&+(\eta\beta+2)\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|+\eta\beta\sqrt{n}\|\bar{x}(k-1)-x^*\|\end{aligned}\quad(13)$$

Step 5: Derive a recursive inequality. We combine the previous four steps into a big recursive inequality. Plug (13) into (10), we have

$$\begin{aligned}\|s(k)-\mathbf{1}g(k)\|\le(\sigma+\beta\eta)\|s(k-1)-\mathbf{1}g(k-1)\|\\+\beta(\eta\beta+2)\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|\\+\eta\beta^2\sqrt{n}\|\bar{x}(k-1)-x^*\|\end{aligned}\quad(14)$$

Combine (14), (11) and (12), we get

$$\overbrace{\begin{bmatrix}\|s(k)-\mathbf{1}g(k)\|\\\|x(k)-\mathbf{1}\bar{x}(k)\|\\\sqrt{n}\|\bar{x}(k)-x^*\|\end{bmatrix}}^{\triangleq z(k)\in\mathbb{R}^3}\le\overbrace{\begin{bmatrix}(\sigma+\beta\eta)&\beta(\eta\beta+2)&\eta\beta^2\\\eta&\sigma&0\\0&\eta\beta&\lambda\end{bmatrix}}^{\triangleq G(\eta)\in\mathbb{R}^{3\times3}}\cdot\overbrace{\begin{bmatrix}\|s(k-1)-\mathbf{1}g(k-1)\|\\\|x(k-1)-\mathbf{1}\bar{x}(k-1)\|\\\sqrt{n}\|\bar{x}(k-1)-x^*\|\end{bmatrix}}^{\triangleq z(k-1)\in\mathbb{R}^3}\quad(15)$$

where '$\le$' means element wise less than or equal to. Since $z(k)$ and $G(\eta)$ have nonnegative entries, we can actually expand (15) recursively, and get

$$z(k)\le G(\eta)^kz(0)$$

This directly leads to the statements of the theorem. □

We now prove Lemma 2.

Proof of Lemma 2: Since $\eta_0<\frac{1}{\beta}$, in the rest of the proof we will let $\lambda=1-\alpha\eta$. Define $\gamma(\eta)=\rho(G(\eta))$ over $\eta\in[0,\eta_0)$. When $\eta=0$, $G(0)$'s eigenvalues are σ, σ and 1. We now investigate how the eigenvalue 1 is perturbed if we slightly increase η from 0. Let the characteristic equation of $G(\eta)$ be $p(y)=0$. This equation defines a implicit map from η to y on a neighborhood of $\eta=0$. From the equation we can calculate $\gamma'(0)=\frac{dy}{d\eta}|_{\eta=0,y=1}=-\alpha<0$. Therefore, there exists a $\eta_1\in(0,\eta_0)$ such that $\gamma(\eta)<1$ on interval $(0,\eta_1)$. Assume there exists a $\eta'\in(0,\eta_0)$ s.t. $\gamma(\eta')\ge1$, by continuity of γ there exists a $\eta''\in[\eta_1,\eta']$ such that $\gamma(\eta'')=1$. Since $G(\eta'')$ is a nonnegative matrix, by Perron-Frobenius Theorem [29], 1 is a eigenvalue of $G(\eta'')$, i.e. $\det(I-G(\eta''))=0\Rightarrow$

$$\eta''\big[\beta^2(\alpha+\beta)\eta''^2+\alpha\beta(3-\sigma)\eta'-\alpha(1-\sigma)^2\big]=0$$

Solve this equation for η'', we get three solutions, one is negative, one is 0, and the last one is

$$\eta''=\frac{2\alpha(1-\sigma)^2}{\sqrt{\Delta}+\alpha\beta(3-\sigma)}$$

where $\Delta=\alpha^2\beta^2(3-\sigma)^2+4\alpha(\alpha+\beta)\beta^2(1-\sigma)^2$. It is easy to verify such a $\eta''>\eta_0$. But $\eta_0>\eta'\ge\eta''$. This is a contradiction. Therefore $\gamma(\eta)<1$ on interval $(0,\eta_0)$. □

D. Convergence Analysis: Non-strongly Convex Case

Here we present the main proof steps for the convergence of the algorithm given non-strongly convex functions.

Proof of Theorem 3: For the ease of technical exposition, we pick a particular configuration at the initial setp. All nodes agree upon a vector $x_0\in\mathbb{R}^{1\times N}$ first, i.e., $x_i(0)=x_0$. Then, each agent calculates $\nabla f_i(x_i(0))$ and sets $s_i(0)=\frac{1}{n}\sum_{i=1}^n\nabla f_i(x_i(0))$. This might need some initial coordination. Note that the theorem is still true if the algorithm uses a general initial setting as described before (Section II-B).

We divide the proof into 3 steps. Step 1 will be devoted to bound what we call 'relative consensus error' (16). After this is done, as will be shown in step 2 and 3, our proof

will follow almost the same as the proof of the $O(\frac{1}{t})$ rate for CGD [25].

Step 1: Bound relative consensus error. In this step, we prove the inequality (16), which is essentially upper bounding the consensus error by the gradient $\|h(k)\|$ (thus the name 'relative consensus error')

$$\|x(k) - \mathbf{1}\bar{x}(k)\| \leq \frac{1}{2}\frac{\sqrt{n}}{\beta}\|h(k)\| \tag{16}$$

We give the proof idea here. (16) is true when $k = 0$. Suppose (16) is true for $k-1$. Then for k, roughly speaking, due to the consensus matrix W we have $\|x(k) - \mathbf{1}\bar{x}(k)\| \leq \tilde{\sigma}\|x(k-1) - \mathbf{1}\bar{x}(k-1)\|$, for some $\tilde{\sigma} \in (0,1)$, while by controlling η to be small enough, $\|h(k)\| \geq \mu\|h(k-1)\|$ for some $\mu > \tilde{\sigma}$. In other words, by controlling η, at every step the consensus error $\|x(k) - \mathbf{1}\bar{x}(k)\|$ decays faster than $\|h(k)\|$. This finishes the induction.

Step 2: Follow the proof of CGD. We can show, when $\eta < \frac{1}{\beta}$, by smoothness and Lemma 5(b),

$$\begin{aligned}
&f(\bar{x}(k+1)) \\
&\leq f(\bar{x}(k)) + \langle h(k), \bar{x}(k+1) - \bar{x}(k)\rangle + \frac{\beta}{2}\|\bar{x}(k+1) - \bar{x}(k))\|^2 \\
&= f(\bar{x}(k)) - \eta\langle h(k), g(k)\rangle + \frac{\beta\eta^2}{2}\|g(k)\|^2 \\
&= f(\bar{x}(k)) + (\frac{\beta\eta^2}{2} - \eta)\|h(k)\|^2 + \frac{\beta\eta^2}{2}\|g(k) - h(k)\|^2 \\
&\quad + (\beta\eta^2 - \eta)\langle h(k), g(k) - h(k)\rangle \\
&\leq f(\bar{x}(k)) + (\frac{\beta\eta^2}{2} - \eta)\|h(k)\|^2 + \frac{\beta\eta^2}{2}\|g(k) - h(k)\|^2 \\
&\quad + (\eta - \beta\eta^2)\frac{\|h(k)\|^2 + \|g(k) - h(k)\|^2}{2} \\
&= f(\bar{x}(k)) - \frac{\eta}{2}\|h(k)\|^2 + \frac{\eta}{2}\|g(k) - h(k)\|^2 \\
&\leq f(\bar{x}(k)) - \frac{\eta}{2}\|h(k)\|^2 + \frac{\eta}{2}\frac{\beta^2}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \qquad (17)
\end{aligned}$$

where the last inequality is from Lemma 6(c). Now plug (16) into (17), we get

$$\begin{aligned}
f(\bar{x}(k+1)) &\leq f(\bar{x}(k)) - \frac{3}{8}\eta\|h(k)\|^2 &\text{(18a)}\\
&\leq f(\bar{x}(0)) - \frac{3}{8}\eta\sum_{\ell=0}^{k}\|h(\ell)\|^2. &\text{(18b)}
\end{aligned}$$

Since f is lower bounded, the above inequality directly shows that $\sum_{\ell=0}^{\infty}\|h(\ell)\|^2 < \infty$, i.e. $\|h(\ell)\|$ is square summable. Next, by (18a), we have $f(\bar{x}(k)) \leq f(\bar{x}(0))$, therefore $\bar{x}(k)$ belongs to f's $f(\bar{x}(0))$-level set. Since the set of minimizers of f is compact, by [26] Proposition B.9, f's level sets are compact. Therefore, $\|\bar{x}(k)\|$ is upper bounded, and so is $\|\bar{x}(k) - x^*\|$. Let $\|\bar{x}(k) - x^*\| \leq C$, and let $\delta_k = f(\bar{x}(k)) - f^*$. By convexity,

$$\delta_k \leq \langle \nabla f(\bar{x}(k)), \bar{x}(k) - x^*\rangle \leq C\|h(k)\|$$

Plug the above into (18a), we have $\delta_{k+1} \leq \delta_k - \frac{3}{8}\eta\frac{1}{C^2}\delta_k^2$, which is equivalent to (without loss of generality we assume $\delta_k \neq 0, \forall k$)

$$\frac{1}{\delta_{k+1}} - \frac{1}{\delta_k} \geq \frac{3}{8}\eta\frac{1}{C^2}\frac{\delta_k}{\delta_{k+1}} \geq \frac{3}{8}\eta\frac{1}{C^2}.$$

Then it immediately follows that $\delta_k = O(\frac{1}{k})$, i.e. $f(\bar{x}(k)) - f^* = O(\frac{1}{k})$.

Step 3: Prove the rest of the statements. $\forall i$, by smoothness and (16)

$$\begin{aligned}
&f(x_i(k)) - f^* \\
&\leq [f(\bar{x}(k)) - f^*] + \langle h(k), x_i(k) - \bar{x}(k)\rangle \\
&\quad + \frac{\beta}{2}\|x_i(k) - \bar{x}(k)\|^2 \\
&\leq [f(\bar{x}(k)) - f^*] + (\frac{1}{2}\frac{\sqrt{n}}{\beta} + \frac{1}{8}\frac{n}{\beta})\|h(k)\|^2
\end{aligned}$$

The first term is $O(\frac{1}{k})$. Since $\|h(k)\|$ is square summable, it is easy to verify $\min_{k' \leq k}\|h(k')\|^2 = O(\frac{1}{k})$. Therefore,

$$\min_{k' \leq k} f(x_i(k')) - f^* = O(\frac{1}{k})$$

which concludes our proof. □

V. Numerical Experiments

We simulate our algorithm and compare it with other algorithms. We choose $n = 100$ agents and the graph is generated using the Erdos-Renyi model [30] with connectivity probability 0.3. The weight matrix W is chosen using the Laplacian method [21]. The algorithms we compare include DGD (4) with a vanishing step size and with a fixed step size, the algorithm proposed in [21] (with $\tilde{W} = \frac{W+I}{2}$), and CGD with a fixed step size. Step sizes are optimized for each algorithm separately. For the functions f_i, we test three cases: i) The functions f_i are square losses for linear regression, i.e. $f_i(x) = \sum_{m=1}^{M_i}(\langle u_{im}, x\rangle - v_{im})^2$ where $u_{im} \in \mathbb{R}^N$ are the features and $v_{im} \in \mathbb{R}$ are the observed outputs, and $\{(u_{im}, v_{im})\}_{m=1}^{M_i}$ are M_i data samples for agent i. ii) The functions f_i are the loss functions for logistic regression [31], i.e. $f_i(x) = \sum_{m=1}^{M_i}\left[\ln(1 + e^{\langle u_{im}, x\rangle}) - v_{im}\langle u_{im}, x\rangle\right]$ where $u_{im} \in \mathbb{R}^N$ are the features and $v_{im} \in \{0, 1\}$ are the observed labels, and $\{(u_{im}, v_{im})\}_{m=1}^{M_i}$ are M_i data samples for agent i; iii) The functions f_i are smooth and convex but $\nabla^2 f$ is zero at the optimum x^*. In details, we choose $N = 1$ and $\forall x \in \mathbb{R}$, $f_i(x) = u(x) + b_i x$, where b_i are randomly chosen that satisfy $\sum_i b_i = 0$, and $u(x) = \frac{1}{4}x^4$ for $|x| \leq 1$, and $u(x) = |x| - \frac{3}{4}$ for $|x| > 1$. Case I and case II satisfy Assumption 1 and 2, while case III only satisfies Assumption 1. In case I and II, we plot the average objective error, i.e. $\frac{1}{n}\sum_i f(x_i(t)) - f^*$. Case III is intended to test the sublinear convergence rate $\frac{1}{t}$ of the algorithm (Theorem 3), therefore we also plot $t \times (\frac{1}{n}\sum_i f(x_i(t)) - f^*)$ to check if the objective error decays as $O(\frac{1}{t})$. The results are shown in Figure 1, 2 and 3. In all cases, DGD with vanishing step size has a slow convergence rate, and DGD with fixed step size has a error. In case I and II, our algorithm and [21] can achieve linear convergence rate, but both slower than CGD. In case III, our algorithm and [21] can both achieve $O(\frac{1}{t})$ rate.

VI. Conclusion

In this paper, we have proposed a method that can effectively harness smoothness to speed up distributed optimization. The method features a novel gradient estimation

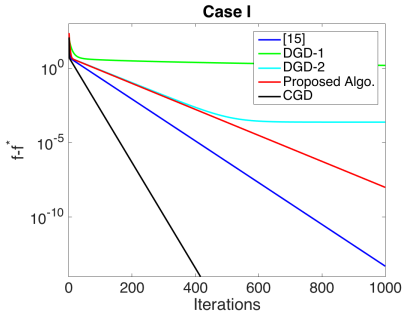

Fig. 1: Simulation results for case I. Green is DGD (4) with vanishing step size; cyan is DGD (4) with fixed step size; blue is the algorithm in [21]; red is our algorithm; black is CGD.

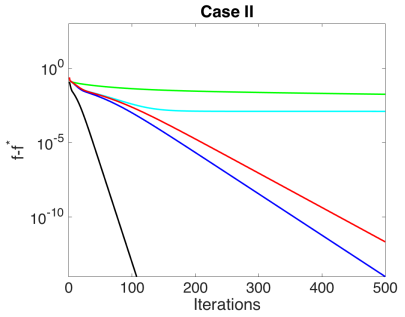

Fig. 2: Simulation results for case II.

scheme. Future work includes giving a better characterization of the step size and the convergence rate, as well as applying the gradient estimation scheme to other first order optimization algorithms.

REFERENCES

[1] B. Johansson, "On distributed optimization in networked systems," 2008.

[2] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, 2010.

[3] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.

[4] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," in *1984 American Control Conference*, 1984, pp. 484–489.

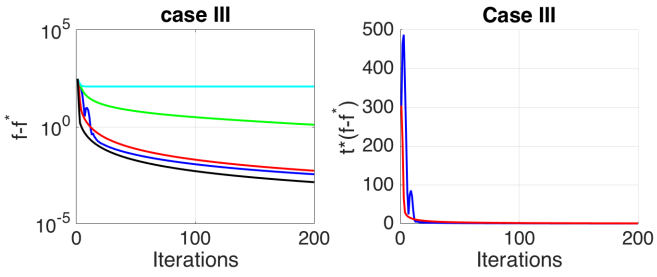

Fig. 3: Simulation results for case III. Left: objective error. Right: t times objective error.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

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

[7] I. Lobel and A. Ozdaglar, "Convergence analysis of distributed subgradient methods over random networks," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 353–360.

[8] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *Automatic control, IEEE Transactions on*, vol. 57, no. 3, pp. 592–606, 2012.

[9] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[10] A. Nedic and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv preprint arXiv:1406.2075*, 2014.

[11] ——, "Distributed optimization over time-varying directed graphs," *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 601–615, 2015.

[12] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 754–771, 2011.

[13] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2014.

[14] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 151–164, 2012.

[15] 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.

[16] I.-A. Chen *et al.*, "Fast distributed first-order methods," Master's thesis, Massachusetts Institute of Technology, 2012.

[17] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv preprint arXiv:1310.7063*, 2013.

[18] D. Jakovetic, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1131–1146, 2014.

[19] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[20] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 601–608.

[21] 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.

[22] G. Qu and N. Li, "Fast distributed nesterov gradient descent," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*. IEEE, 2016.

[23] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[24] 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.

[25] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[26] D. P. Bertsekas, "Nonlinear programming," 1999.

[27] J. N. Tsitsiklis and Z.-Q. Luo, "Communication complexity of convex optimization," *Journal of Complexity*, vol. 3, no. 3, pp. 231–243, 1987.

[28] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *arXiv preprint arXiv:1605.07112*, 2016.

[29] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.

[30] P. Erdos and A. Renyi, "On random graphs i," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[31] (2012) Logistic regression. [Online]. Available: http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf