DISTRIBUTED NESTEROV GRADIENT METHODS FOR RANDOM NETWORKS: CONVERGENCE IN PROBABILITY AND CONVERGENCE RATES

Dušan Jakovetić[1], *João Xavier*[2], *and José M. F. Moura*[3]

[1]University of Novi Sad, BioSense Center, Serbia
[2]Inst. for Systems and Robotics, Instituto Superior Técnico, Technical University of Lisbon, Portugal
[3]Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

We consider distributed optimization where N nodes in a generic, connected network minimize the sum of their individual, locally known, convex costs. Existing literature proposes distributed gradient-like methods that are attractive due to computationally cheap iterations and provable resilience to random inter-node communication failures, but such methods have slow theoretical and empirical convergence rates. Building from the centralized Nesterov gradient methods, we propose accelerated distributed gradient-like methods and establish that they achieve strictly faster rates than existing distributed methods. At the same time, our methods maintain cheap iterations and resilience to random communication failures. Specifically, for convex, differentiable local costs with Lipschitz continuous and bounded derivative, we establish (with respect to the cost function optimality) convergence in probability and convergence rates in expectation and in second moment.

Index Terms— Distributed optimization, convergence rate, random networks, Nesterov gradient, consensus

1. INTRODUCTION

We develop distributed, Nesterov-like, gradient algorithms and establish their convergence and convergence rate guarantees on random networks. We assume a standard N-node random network, e.g., [1, 2], and distributed optimization models, e.g., [3, 1]. The network model assumes a sequence of independent, identically distributed (i.i.d.) $N \times N$ weight matrices $\{W(k)\}$, where $W(k)$ respects the sparsity of the inter-node communication pattern at time k. The matrices $W(k)$ are drawn from the set of symmetric, stochastic matrices with positive diagonals, and the graph that supports the expectation of $W(k)$ is connected. The distributed optimization model assumes that N nodes cooperatively minimize the sum $\sum_{i=1}^{N} f_i(x)$ of their locally known convex costs with respect to the global variable $x \in R^d$; such model encompasses many applications, including distributed inference, e.g., [2], and source localization, e.g., [4], in sensor networks and distributed learning of a linear classifier, e.g., [5]. For this network-optimization model, existing literature develops distributed gradient methods, e.g. [3], which are attractive due to easy-to-implement, computationally cheap iterations and provable resilience to random communication failures. However, these methods have slow theoretical and practical convergence rates.

We design two accelerated distributed gradient methods for random networks, building from the centralized Nesterov gradient algorithm [6]. Our methods enjoy computationally cheap iterations and resilience to communication failures, like the methods in, e.g., [3], but they have strictly faster rates than the methods in [3]. We achieve this when the costs f_i's are convex, differentiable, and have Lipschitz continuous and bounded derivatives. Our two algorithms, termed mD–NG and mD–NC, modify the D-NG and D-NC methods that we previously proposed in [5] for static networks. (Here, mD–NG abbreviates "Modified Distributed Nesterov Gradient," and mD–NC abbreviates "Modified Distributed Nesterov gradient with Consensus iterations.") mD-NG achieves rates $O(\log k/k)$ and $O(\log \mathcal{K}/\mathcal{K})$ in the expected optimality gap at the cost function, where k is the number of per-node gradient evaluations and $\mathcal{K}$ is the number of per-node ($2d$-dimensional) vector communications. mD-NC achieves rates $O(1/k^2)$ and $O(1/\mathcal{K}^{2-\xi})$, where $\xi > 0$ is an arbitrarily small positive number. For comparison, [3] cannot achieve a rate (in a worst-case sense) better than $\Omega(k^{-2/3})$ and $\Omega(\mathcal{K}^{-2/3})$, [5] (See the last paragraph in Section 1 for the meaning of symbols O and Ω.) We further show that, with both mD–NG and mD–NC, optimality gap at the cost function converges to zero in probability, and with mD–NC also almost surely. Finally, for a special case of spatially independent link failures, we find with both methods the rates of convergence in the expected squared optimality gap at the cost (second moment's

The work of the first and second authors was supported by: the Carnegie Mellon|Portugal Program under a grant from the Fundação de Ciência e Tecnologia (FCT) from Portugal; by FCT grants CMU-PT/SIA/0026/2009 and SFRH/BD/33518/2008 (through the Carnegie Mellon|Portugal Program managed by ICTI); by ISR/IST plurianual funding (POSC program, FEDER), and the work of the first and third authors was funded by AFOSR grant FA95501010291 and by NSF grant CCF1011903.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

convergence rate).

We briefly comment on the related literature. Reference [3] proposes standard distributed gradient methods and analyzes them for deterministically time-varying networks, [7] analyzes these methods for asynchronous gossip protocols, and [1] analyzes them for random networks. Reference [8] proposes a different, distributed dual averaging method, and analyzes it on both static and random networks. Reference [9] proposes an accelerated, Nesterov-like, distributed proximal gradient method and analyzes it for deterministically varying networks. In summary, our work contrasts with the literature by *simultaneously* considering: 1) accelerated, Nesterov-like gradient methods, and 2) random networks, and by establishing convergence and convergence rate guarantees for such scenario.

The remainder of the paper is organized as follows. Section 2 introduces the model that we assume and presents our mD–NG and mD–NC distributed algorithms. Section 3 states our convergence results on the cost function's optimality gap: convergence in probability (and almost sure convergence with mD–NC), as well as convergence rates in expectation and in second moment. We conclude in Section 4.

Throughout, we use the following notation. Denote by: $\mathbb{R}^d$ the d-dimensional real space: A_{lm} or $[A]_{lm}$ the (l,m) entry of A; $[a]_{l:m}$ the selection of the l-th, $(l+1)$-th, $\cdots$, m-th entries of vector a; $\|\cdot\|$ the vector (matrix) Euclidean (spectral) norm of its vector (matrix) argument; $\lambda_i(\cdot)$ the i-th largest eigenvalue; $\lceil a \rceil$ the smallest integer greater than or equal to a real scalar a; $\nabla\phi(x)$ the gradient at x of a differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$, $d \geq 1$; and $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ the probability and expectation, respectively. For two positive sequences η_n and χ_n, we have: $\eta_n = O(\chi_n)$ if $\limsup_{n\to\infty} \frac{\eta_n}{\chi_n} < \infty$; $\eta_n = \Omega(\chi_n)$ if $\liminf_{n\to\infty} \frac{\eta_n}{\chi_n} > 0$.

2. MODEL AND ALGORITHMS

Subsection 2.1 introduces the network and optimization models that we assume. Subsection 2.2 presents the mD–NG algorithm, and Subsection 2.3 presents mD–NC.

2.1. Model

Optimization model. Nodes solve the following problem unconstrained:

$$\text{minimize} \; \sum_{i=1}^{N} f_i(x) =: f(x). \tag{1}$$

The function $f_i : \mathbb{R}^d \to \mathbb{R}$ is known only by node i, $\forall i$, and it obeys the following.

Assumption 1 (Optimization model) (a) (Solbability) There exists a solution $x^\star \in \mathbb{R}^d$ such that $f(x^\star) = f^\star := \inf_{x\in\mathbb{R}^d} f(x)$.

(b) (Lipschitz continuous gradient) For all i, f_i is convex, differentiable, and has Lipschitz continuous gradient with constant $L \in [0,\infty)$: $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \;\; \forall x, y \in \mathbb{R}^d$.

(c) (Bounded gradient) There exists a constant $G \in [0,\infty)$ such that, $\forall i$, $\|\nabla f_i(x)\| \leq G$, $\forall x \in \mathbb{R}^d$.

Examples of the f_i's that obey Assumption 1 are logistic, Huber, and fair losses, see [5].

Network model. The inter-node communication pattern at time step k is described by a random $N \times N$ symmetric, stochastic weight matrix $W(k)$, in the sense that $W_{ij}(k) > 0$ if and only if nodes i and j communicate at time step k. Define the undirected graph $\mathcal{G} := (\mathcal{N}, E)$, where $E = \{\{i,j\} : \mathbb{E}[W_{ij}(k)] > 0, \, i < j\}$. In words, $\mathcal{G}$ collects all pairs of nodes that communicate with a non-zero probability. We impose the following assumption.

Assumption 2 (Random network) We have:

(a) The sequence $\{W(k)\}_{k=1}^{\infty}$ is i.i.d.

(b) Almost surely (a.s.), $W(k)$ are symmetric and stochastic (and hence are doubly stochastic), with strictly positive diagonal entries.

(c) There exists $\underline{w} > 0$ such that, for all $i, j = 1, \cdots, N$, a.s. $W_{ij}(k) \notin (0, \underline{w})$.

(d) The graph $\mathcal{G}$ is connected.

The entries $W_{ij}(k)$, $\{i,j\} \in E$, may take the value zero, but are greater than zero with positive probability; whenever $W_{ij}(k)$ takes a positive value, this value is at least $\underline{w}$.

For future reference, introduce:

$$\overline{\mu} := \left(\lambda_2\left(\mathbb{E}\left[W^2(k)\right]\right)\right)^{1/2}. \tag{2}$$

This quantity measures the speed of consensus (in the mean squared sense) driven by the product $W(k)W(k-1)...W(1)$.

2.2. Algorithm mD–NG

We now present our mD–NG algorithm. Each node, over iterations k, maintains its solution estimate $x_i(k) \in \mathbb{R}^d$, and an auxiliary variable $y_i(k) \in \mathbb{R}^d$. Given the initialization $x_i(0) = y_i(0) \in \mathbb{R}^d$, $x_i(0)$ arbitrary, the update at iteration k, $k = 1, 2, ...$, is:

$$x_i(k) = \sum_{j\in O_i(k)} W_{ij}(k)\, y_j(k-1) - \alpha_{k-1}\nabla f_i(y_i(k-1)) \tag{3}$$

$$y_i(k) = (1+\beta_{k-1})\, x_i(k) - \beta_{k-1} \sum_{j\in O_i(k)} W_{ij}(k)\, x_j(k-1). \tag{4}$$

In (3)–(4), $O_i(k) = \{j : W_{ij}(k) > 0\}$ is the random neighborhood of node i, including node i; the step-size α_k and the sequence β_k, $k = 0, 1, ...$, are given by:

$$\alpha_k = \frac{c}{k+1}, \quad c \leq \frac{1}{2L}, \quad \beta_k = \frac{k}{k+3}. \tag{5}$$

At iteration k, node i broadcasts $x_i(k-1)$ and $y_i(k-1)$ to all its neighbors and receives $x_j(k-1)$ and $y_j(k-1)$ from all current neighbors $j \in O_i(k) - \{i\}$. Upon reception, node i updates $x_i(k)$ via (3) and $y_i(k)$ via (4).

It is instructive to compare the mD–NG algorithm for random networks with its precursor D–NG in [5]. D–NG is proposed for static networks and is the same as mD–NG, except that it replaces the term $\sum_{j \in O_i(k)} W_{ij}(k)\, x_j(k-1)$ in (4) with $x_i(k-1)$. In other words, mD–NG, when compared with D–NG, introduces an additional per-node communication at each iteration k. This allows with mD–NG for structural robustness to random variations in $W(k)$. In contrast with mD–NG, D–NG may diverge when the network is random; we refer to a companion journal paper [10] for details. This interesting difference between mD–NG and D–NG may be, in a certain sense, related to the difference between adapt-then-combine and combine-then-adapt methods studied in [11].

2.3. Algorithm mD–NC

Algorithm mD–NC operates in two time scales, outer iterations k and inner iterations s. There are τ_k inner iterations at the outer iteration k, where we set (recall $\overline{\mu}$ in (2)):

$$\tau_k = \left\lceil \frac{3 \log k + \log N}{-\log \overline{\mu}} \right\rceil. \tag{6}$$

For convenience, we introduce a two index notation $W(k, s)$ for the random weight matrix that describes the communication pattern at inner iteration s and outer iteration k. We order the weight matrices lexicographically in the sequence as $W(k = 1, s = 1), W(k = 1, s = 2), \cdots, W(k = 1, s = \tau_1), \cdots, W(k = 2, s = 1), \cdots$, and let the sequence obey Assumption 2.

The mD–NC method is summarized in Algorithm 1. It uses a constant step-size $\alpha \leq 1/(2L)$. Each node i maintains, over outer iterations k, its solution estimate $x_i(k) \in \mathbb{R}^d$ and an auxiliary variable $y_i(k) \in \mathbb{R}^d$. At each outer iteration k, nodes run a consensus algorithm with τ_k (inner) iterations; each inner iteration requires, per node, a $2d$-dimensional broadcast transmission to all neighbors (See Algorithm 1 for details.) Hence, major differences between mD–NG and mD–NC are that: 1) mD–NG uses a diminishing step-size, while mD–NC uses a constant step-size; and 2) mD–NG effectively has one consensus round per k, while mD–NC has multiple (τ_k) consensus rounds per k.

We close this Section by noting that, with both our methods, we assumed a certain global knowledge by all nodes, acquired beforehand in a network training period. Specifically, with mD–NG, all nodes know the gradient's Lipschitz constant L to set the step-size in (5); with mD–NC, nodes know L to set the step-size $\alpha \leq 1/(2L)$, and the number of nodes N and the quantity $\mu := \left(\lambda_2\left(\mathbb{E}[W^2(k)]\right)\right)^{1/2}$, to set τ_k in (6). We can modify our methods and relax these prior knowledge requirements such that the methods still provable converge, at rates that are close to the ones presented in this paper; for details, we refer to [10].

Algorithm 1 mD–NC

```
1: Initialization: Node $i$ sets $x_i(0)=y_i(0) \in \mathbb{R}^d$; and $k = 1$.
2: Node $i$ calculates $x_i^{(a)}(k) = y_i(k-1) - \alpha \nabla f_i(y_i(k-1))$.
3: (Consensus) Nodes run average consensus on $\chi_i(s,k)$, initialized by $\chi_i(s=0,k) = \left(x_i^{(a)}(k)^\top, x_i(k-1)^\top\right)^\top$:

       $\chi_i(s,k) = \sum_{j \in O_i(k)} W_{ij}(k,s) \chi_j(s-1,k), s = 1, 2, \cdots, \tau_k,$

   with $\tau_k$ in (6), and set $x_i(k) := [\chi_i(s=\tau_k,k)]_{1:d}$ and $x_i^{(b)}(k-1) := [\chi_i(s=\tau_k,k)]_{d+1:2\,d}$. (Here $[a]_{l:m}$ is a selection of $l$-th, $l+1$-th, $\cdots$, $m$-th entries of vector $a$.)
4: Node $i$ calculates $y_i(k) = (1+\beta_{k-1}) x_i(k) - \beta_{k-1}\, x_i^{(b)}(k-1)$.
5: Set $k \mapsto k+1$ and go to step 2.
```

3. CONVERGENCE ANALYSIS

In this Section, we characterize with both mD–NG and mD–NC the optimality gap $f(x_i) - f^\star \geq 0$ at any node i, with respect to the number of per-node gradient evaluations k and per-node ($2d$-dimensional) communications $\mathcal{K}$. Note that, with mD–NG, we have that $k = \mathcal{K}$, i.e., one per-node communication corresponds to one per-node gradient evaluation. With mD–NC, τ_k (see (6)) per-node communications correspond to one per-node gradient evaluation; it is easy to show that $\mathcal{K} = O(k \log k)$.

We first state our results on the convergence rate in the expected optimality gaps with mD–NG and mD–NC. In subsequent results, ξ denotes an arbitrarily small positive number.

Theorem 1 (Convergence rates in expectation) Let Assumptions 1 and 2 hold. Then, at any node i, the expected optimality gap $\mathbb{E}\left[f(x_i)\right] - f^\star$ is:

(a) With mD–NG: $O(\log k/k)$ and $O(\log \mathcal{K}/\mathcal{K})$;

(b) With mD–NC: $O(1/k^2)$ and $O(1/\mathcal{K}^{2-\xi})$.

A proof of Theorem 1, as well as explicit constants in the established rates, can be found in [10]. Theorem 1 indicates that the convergence rates do not depend on the underlying random network statistics. However, the convergence constants actually depend on $\overline{\mu}$ in (2). From Theorem 1, we can see that mD–NC achieves faster theoretical rates than mD–NG. Typically, in simulations, mD–NG actually converges faster for practical accuracies, see [10].

A direct corollary of Theorem 1, through an application of Markov inequality, is the convergence in probability of mD–NG and mD–NC. Furthermore, using the technique in, e.g., ([12], Subsection IV–A), it can be shown that mD–NC also converges almost surely (a.s.)

Corollary 2 (Convergence in probability and a.s. convergence) Let Assumptions 1 and 2 hold. Then, at any node i:

(a) With mD–NG: $\mathbb{P}\left(f(x_i) - f^\star > \epsilon\right) \to 0$ as $k \to \infty$;

(b) With mD–NC: $\mathbb{P}\left(f(x_i) - f^\star > \epsilon\right) \to 0$ as $k \to \infty$, and $\mathbb{P}\left(\lim_{k\to\infty}(f(x_i) - f^\star) = 0\right) = 1$.

Finally, we establish with both methods convergence rates in the second moment, for a special case when the link failures are spatially independent. A proof of the Theorem below can be found in [10].

Theorem 3 (Convergence rates in second moment) Let Assumptions 1 and 2 hold. Further, assume that the random variables $W_{ij}(k)$ that correspond to different links $\{i,j\} \in E$ are mutually independent. Then, at any node i, the expected squared optimality gap $\mathbb{E}\left[(f(x_i) - f^\star)^2\right]$ is:

(a) With mD–NG: $O\left(\frac{\log^2 k}{k^2}\right)$ and $O\left(\frac{\log^2 \mathcal{K}}{\mathcal{K}^2}\right)$.

(b) With mD–NC: $O\left(\frac{1}{k^4}\right)$ and $O\left(\frac{1}{\mathcal{K}^{4-2\xi}}\right)$.

We interpret Theorem 3 for mD–NG, while a similar interpretation is in place for mD–NC also. Theorem 1 says that $\epsilon_k := (f(x_i(k)) - f^\star)\frac{k}{\log k}$ is, in expectation, upper bounded by a certain constant $C > 0$, for all k. Theorem 3 strengthens the latter claim by saying that the second moment of ϵ_k is also upper bounded by a constant $C' > 0$.

4. CONCLUSION

We considered distributed optimization over random networks, where the goal for each node is to minimize the sum of locally known nodes' convex costs. We design two distributed Nesterov-like gradient methods, referred to as mD–NG and mD–NC, and we characterize for both methods the optimality gap at the cost function at any node i, with respect to the number of per-node gradient evaluations k and per-node communications $\mathcal{K}$. Specifically, we show with both methods: 1) convergence in probability (and also almost sure convergence for mD–NC); 2) convergence rates in expectation; and 3) convergence rates in the second moment for spatially independent link failures.

5. REFERENCES

[1] I. Lobel and A. Ozdaglar, "Convergence analysis of distributed subgradient methods over random networks," in *46th Annual Allerton Conference onCommunication, Control, and Computing*, Monticello, Illinois, September 2008, pp. 353 – 360.

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

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

[4] M Rabbat and R Nowak, "Distributed optimization in sensor networks," in *IPSN 2004, 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, California, USA, April 2004, pp. 20 – 27.

[5] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *conditionally accepted to IEEE Trans. Autom. Contr.*, Jan. 2013, available at: http://arxiv.org/abs/1112.2972.

[6] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate O$(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983, (in Russian).

[7] S. Sundhar Ram, A. Nedic, and V.V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *CDC '09, 48th IEEE International Conference on Decision and Control*, Shanghai, China, December 2009, pp. 3581 – 3586.

[8] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence and network scaling," *IEEE Trans. Aut. Contr.*, vol. 57, no. 3, pp. 592–606, March 2012.

[9] I.-A. Chen and A. Ozdaglar, "A fast distributed proximal gradient method," in *Allerton Conference on Communication, Control and Computing*, Monticello, IL, October 2012.

[10] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Convergence rates of distributed Nesterov-like gradient methods on random networks," *to appear in IEEE Transactions on Signal Processing*, available at: http://arxiv.org/abs/1308.0916.

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

[12] Alireza Tahbaz-Salehi and Ali Jadbabaie, "On consensus in random networks," in *44th Annual Allerton Conference on Communication, Control, and Computing*, Allerton House, Illinois, USA, September 2006, pp. 1315–1321.