ARTICLE IN PRESS

JID: NEUCOM [m5G;July 8, 2017;14:43]

Neurocomputing 000 (2017) 1–8

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

On the distributed optimization over directed networks[☆]

Chenguang Xi[a], Qiong Wu[b], Usman A. Khan[a,*]

[a] Department of Electrical and Computer Engineering, Tufts University, 161 College Ave., Medford, MA 02155, USA
[b] Department of Mathematics, Tufts University, 503 Boston Ave., Medford, MA 02155, USA

ARTICLE INFO

Article history:
Received 2 October 2016
Revised 3 March 2017
Accepted 13 June 2017
Available online xxx

Communicated by Choon Ki Ahn

Keywords:
Distributed optimization
Multi-agent networks
Directed graphs
Distributed subgradient descent

ABSTRACT

In this paper, we propose a distributed algorithm, called Directed-Distributed Subgradient Descent (D-DSD), to solve multi-agent optimization problems over *directed* graphs. Existing algorithms mostly deal with similar problems under the assumption of undirected networks, i.e., requiring the weight matrices to be doubly-stochastic. The row-stochasticity of the weight matrix guarantees that all agents reach consensus, while the column-stochasticity ensures that each agent's local (sub)gradient contributes equally to the global objective. In a directed graph, however, it may not be possible to construct a doubly-stochastic weight matrix in a distributed manner. We overcome this difficulty by augmenting an additional variable for each agent to record the change in the state evolution. In each iteration, the algorithm simultaneously constructs a row-stochastic matrix and a column-stochastic matrix instead of only a doubly-stochastic matrix. The convergence of the new weight matrix, depending on the row-stochastic and column-stochastic matrices, ensures agents to reach both consensus and optimality. The analysis shows that the proposed algorithm converges at a rate of $O(\frac{\ln k}{\sqrt{k}})$, where k is the number of iterations.

1. Introduction

Distributed computation and optimization has received significant recent interest in many areas, e.g., distributed machine learning, [2], distributed estimation, [33], cognitive networks, [13], source localization, [23], distributed coordination, [30], and message routing, [19]. The related problems can be posed as the minimization of a sum of objectives, $\sum_{i=1}^{n} f_i(\mathbf{x})$, where $f_i : \mathbb{R}^p \to \mathbb{R}$ is a private objective function at the ith agent. There are two general types of distributed algorithms to solve this problem. The first type is a (sub)gradient based method [4,6,9,12,17,18,24], where at each iteration a (sub)gradient related step is calculated, followed by averaging with neighbors in the network. The main advantage of these methods is computational simplicity. The (sub)gradient based methods are generalized to mirror descent methods [10,11,34] by using the Bregman divergence as distance-measuring function rather than the Euclidean distance. The second type of distributed algorithms are based on augmented Lagrangians, where at each iteration the primal variables are solved to minimize a Lagrangian related function, followed by updating the dual variables accordingly, e.g., the Distributed Alternating Direction Method of Multipliers (D-ADMM), [14,26,31]. The latter type is preferred when agents can solve the local optimization problem efficiently. Most proposed distributed algorithms, [4,6,9,11,12,14,17,18,24,26,31,34], assume undirected graphs. The primary reason behind assuming the undirected graphs is to obtain a doubly-stochastic weight matrix. The row-stochasticity of the weight matrix guarantees that all agents reach consensus, while the column-stochasticity ensures optimality.

In this paper, we propose a (sub)gradient based method solving distributed optimization problem over the *directed* graph, which we refer to as the Directed-Distributed Subgradient Descent (D-DSD). Clearly, a directed topology has broader applications in contrast to undirected graphs and may further result in reduced communication cost and simplified topology design. We start by explaining the necessity of weight matrices being doubly-stochastic in existing (sub)gradient based method, e.g., DSD. In the iteration of DSD, agents will not reach consensus if the row sum of the weight matrix is not equal to one. On the other hand, if the column of the weight matrix does not sum to one, each agent will contribute differently to the network. Since doubly-stochastic matrices may not be achievable in a directed graph, the original methods, e.g., DSD, no longer work. We overcome this difficulty in a directed graph by augmenting an additional variable for each agent to record the state updates. In each iteration of the D-DSD algorithm, we simultaneously construct a row-stochastic matrix and a column-stochastic matrix instead of only a doubly-stochastic

☆ This work has been partially supported by an NSF Award # CCF-1350264.

* Corresponding author.
E-mail addresses: chenguang.xi@tufts.edu (C. Xi), qiong.wu@tufts.edu (Q. Wu), khan@ece.tufts.edu (U.A. Khan).

http://dx.doi.org/10.1016/j.neucom.2017.06.038
0925-2312/© 2017 Elsevier B.V. All rights reserved.

matrix. We give an intuitive explanation of our proposed algorithm and further provide convergence and convergence rate analysis as well.

In the context of directed graphs, related work has considered (sub)gradient based algorithms, [15,16,27–29], by combining subgradient descent and push-sum consensus. The push-sum algorithm, [1,7], is first proposed in consensus problems[1] to achieve average-consensus given a column-stochastic matrix. The idea is based on computing the stationary distribution (the left eigenvector of the weight matrix corresponding to eigenvalue 1) for the Markov chain characterized by the multi-agent network and canceling the imbalance by dividing with the left-eigenvector. The algorithms in [15,16,27–29] follow a similar spirit of push-sum consensus and propose nonlinear (because of division) methods. In contrast, our algorithm follows linear iterations and does not involve any division while providing the same convergence rate as the nonlinear one in e.g., [16]. Finally, the analysis and proofs in our work are completely different than the nonlinear counterparts described here.

The remainder of the paper is organized as follows. In Section 2, we provide the problem formulation and show the reason why DSD fails to converge to the optimal solution over directed graphs. We subsequently present the D-DSD algorithm and the necessary assumptions. The convergence analysis of the D-DSD algorithm is studied in Section 3, consisting of agents' consensus analysis and optimality analysis. The convergence rate analysis and numerical experiments are presented in Sections 4 and 5. Section 6 contains concluding remarks.

Notation: Lowercase bold letters denote vectors and uppercase italic letters denote matrices. We denote by $[\mathbf{x}]_i$ the ith component of a vector $\mathbf{x}$, and by $[A]_{ij}$ the (i, j)th element of a matrix, A. An n-dimensional vector of all ones or zeros is represented by $\mathbf{1}_n$ or $\mathbf{0}_n$. The notation $0_{n\times n}$ represents an $n \times n$ matrix with all elements equal to zero. The inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is $\langle \mathbf{x}, \mathbf{y}\rangle$. We use $\|\mathbf{x}\|$ to denote the standard Euclidean norm.

2. Problem formulation

Consider a strongly-connected network of n agents communicating over a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents, and $\mathcal{E}$ is the collection of ordered pairs, $(i, j), i, j \in \mathcal{V}$, such that agent j can send information to agent i. Define $\mathcal{N}_i^{\text{in}}$ to be the collection of in-neighbors, i.e., the set of agents that can send information to agent i. Similarly, $\mathcal{N}_i^{\text{out}}$ is defined as the out-neighborhood of agent i, i.e., the set of agents that can receive information from agent i. We allow both $\mathcal{N}_i^{\text{in}}$ and $\mathcal{N}_i^{\text{out}}$ to include the node i itself. Note that in a directed graph $\mathcal{N}_i^{\text{in}} \neq \mathcal{N}_i^{\text{out}}$, in general. We focus on solving a convex optimization problem that is distributed over the above network. In particular, the network of agents cooperatively solve the following optimization problem:

$$\text{P1}: \quad \min f(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}),$$

where each $f_i : \mathbb{R}^p \to \mathbb{R}$ is convex, not necessarily differentiable, representing the local objective function at agent i.

Assumption 1. In order to solve the above problem, we make the following assumptions:

(a) The agent graph, $\mathcal{G}$, is strongly-connected.
(b) Each local function, $f_i : \mathbb{R}^p \to \mathbb{R}$, is convex, $\forall i \in \mathcal{V}$.
(c) The solution set of Problem P1 and the corresponding optimal value exist. Formally, we have
$$\mathbf{x}^* \in \mathcal{X}^* = \left\{\mathbf{x} | f(\mathbf{x}) = \min_{\mathbf{y}\in\mathbb{R}^p} f(\mathbf{y})\right\}, f^* = \min f(\mathbf{x}).$$
(d) The (sub)gradient, $\nabla f_i(\mathbf{x})$, is bounded:
$$\|\nabla f_i(\mathbf{x})\| \le D,$$
for all $\mathbf{x} \in \mathbb{R}^p, i \in \mathcal{V}$.

The Assumptions 1 are standard in distributed optimization, see related literature, [18], and references therein. Before describing our algorithm, we first recap the DSD algorithm, [17], to solve P1 in an undirected graph. This algorithm requires doubly-stochastic weight matrices. We analyze the influence to the result of the DSD when the weight matrices are *not* doubly-stochastic.

2.1. Distributed subgradient descent

Consider Distributed Subgradient Descent (DSD), [17], to solve P1. Agent i updates its estimate as follows:

$$\mathbf{x}_i^{k+1} = \sum_{j=1}^n w_{ij}\mathbf{x}_j^k - \alpha_k \nabla f_i^k, \tag{1}$$

where w_{ij} is a non-negative weight such that $W = \{w_{ij}\}$ is doubly-stochastic. The scalar, α_k, is a diminishing but non-negative step-size, satisfying the persistence conditions, [8,12]: $\sum_{k=0}^\infty \alpha_k = \infty$, $\sum_{k=0}^\infty \alpha_k^2 < \infty$, and the vector ∇f_i^k is a (sub)gradient of f_i at $\mathbf{x}_i^k$. For the sake of argument, consider W to be row-stochastic but not column-stochastic. Clearly, $\mathbf{1}$ is a right eigenvector of W, and let $\boldsymbol{\pi} = \{\pi_i\}$ be its left eigenvector corresponding to eigenvalue 1. Summing over i in Eq. (1), we get

$$\begin{aligned}\widehat{\mathbf{x}}^{k+1} &\triangleq \sum_{i=1}^n \pi_i \mathbf{x}_i^{k+1},\\ &= \sum_{j=1}^n \left(\sum_{i=1}^n \pi_i w_{ij}\right)\mathbf{x}_j^k - \alpha_k \sum_{i=1}^n \pi_i \nabla f_i(\mathbf{x}_i^k),\\ &= \widehat{\mathbf{x}}^k - \alpha_k \sum_{i=1}^n \pi_i \nabla f_i^k, \end{aligned}\tag{2}$$

where $\pi_j = \sum_{i=1}^n \pi_i w_{ij}, \forall i, j$. If we assume that the agents reach an agreement, then Eq. (2) can be viewed as an inexact (central) subgradient descent (with $\sum_{i=1}^n \pi_i \nabla f_i(\mathbf{x}_i^k)$ instead of $\sum_{i=1}^n \pi_i \nabla f_i(\widehat{\mathbf{x}}^k)$) minimizing a new objective, $\widehat{f}(\mathbf{x}) \triangleq \sum_{i=1}^n \pi_i f_i(\mathbf{x})$. As a result, the agents reach consensus and converge to the minimizer of $\widehat{f}(\mathbf{x})$.

Now consider the weight matrix, W, to be column-stochastic but not row-stochastic. Let $\overline{\mathbf{x}}^k$ be the average of agents estimates at time k, then Eq. (1) leads to

$$\begin{aligned}\overline{\mathbf{x}}^{k+1} &\triangleq \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^{k+1},\\ &= \frac{1}{n}\sum_{j=1}^n \left(\sum_{i=1}^n w_{ij}\right)\mathbf{x}_j^k - \frac{\alpha_k}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^k),\\ &= \overline{\mathbf{x}}^k - \left(\frac{\alpha_k}{n}\right)\sum_{i=1}^n \nabla f_i^k. \end{aligned}\tag{3}$$

Eq. (3) reveals that the average, $\overline{\mathbf{x}}^k$, of agents estimates follows an inexact (central) subgradient descent ($\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^k)$ instead of $\sum_{i=1}^n \nabla f_i(\overline{\mathbf{x}}^k)$) with stepsize α^k/n, thus reaching the minimizer of $f(\mathbf{x})$. Despite the fact that the average, $\overline{\mathbf{x}}^k$, reaches the optima, $\mathbf{x}^*$, of $f(\mathbf{x})$, the optima is not achievable for each agent because consensus can not be reached with a matrix that is not necessary row-stochastic.

[1] See, [5,20–22,25,32], for additional information on average-consensus problems.

Please cite this article as: C. Xi et al., On the distributed optimization over directed networks, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.06.038