Automatica 83 (2017) 37–46

Contents lists available at ScienceDirect

Automatica

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

Exponential convergence of a distributed algorithm for solving linear algebraic equations[☆]

Ji Liu[a], A. Stephen Morse[b], Angelia Nedić[c], Tamer Başar[a]

[a] University of Illinois at Urbana-Champaign, United States
[b] Yale University, United States
[c] Arizona State University, United States

ARTICLE INFO

Article history:
Received 12 October 2015
Received in revised form 16 April 2017
Accepted 18 April 2017

ABSTRACT

In a recent paper, a distributed algorithm was proposed for solving linear algebraic equations of the form $Ax = b$ assuming that the equation has at least one solution. The equation is presumed to be solved by m agents assuming that each agent knows a subset of the rows of the matrix $\begin{bmatrix} A & b \end{bmatrix}$, the current estimates of the equation's solution generated by each of its neighbors, and nothing more. Neighbor relationships are represented by a time-dependent directed graph $\mathbb{N}(t)$ whose vertices correspond to agents and whose arcs characterize neighbor relationships. Sufficient conditions on $\mathbb{N}(t)$ were derived under which the algorithm can cause all agents' estimates to converge exponentially fast to the same solution to $Ax = b$. These conditions were also shown to be necessary for exponential convergence, provided the data about $\begin{bmatrix} A & b \end{bmatrix}$ available to the agents is "non-redundant". The aim of this paper is to relax this "non-redundant" assumption. This is accomplished by establishing exponential convergence under conditions which are the weakest possible for the problem at hand; the conditions are based on a new notion of graph connectivity. An improved bound on the convergence rate is also derived.

1. Introduction

Over the past few decades, there has been considerable interest in developing algorithms for information distribution and computation among agents via local interactions (Jadbabaie, Lin, & Morse, 2003; Tsitsiklis, 1984). Recently, the need for distributed processing has arisen naturally in multi-agent and sensor networks (Dimakis, Kar, Moura, Rabbat, & Scaglione, 2010; Liu, Mou, Morse, Anderson, & Yu, 2011; Olfati-Saber, Fax, & Murray, 2007) because autonomous agents or mobile sensors are physically separated from each other and communication constraints limit the flow of information across a multi-agent or sensor network and consequently preclude centralized processing. As a consequence, distributed computation and decision making problems of all types have arisen naturally; notable examples include consensus, multi-agent coverage problems, the rendezvous problem, localization of sensors in a multi-sensor network, and the distributed management of multi-agent formations. One of the most important numerical computations involving real numbers is solving a system of linear algebraic equations, which has received much attention for a long time, especially in the parallel processing community where the main objective is to solve the system faster or more accurately. It is with these thoughts in mind that we are interested in the problem of solving a system of linear algebraic equations in a distributed manner, introduced in more precise terms as follows.

Consider a network of $m > 1$ autonomous agents which are able to receive information from their "neighbors". Neighbor relationships are characterized by a time-dependent directed graph $\mathbb{N}(t)$ with m vertices and a set of arcs defined so that there is an arc in the graph from vertex j to vertex i whenever agent j is a *neighbor* of agent i. Thus, the directions of arcs represent the directions of information flow. For simplicity, we take each agent to be a neighbor of itself. Thus, $\mathbb{N}(t)$ has self-arcs at all vertices. Each agent i has a real-time dependent state vector $x_i(t)$ taking values in $\mathbb{R}^n$, and we assume that the information agent i receives from neighbor j is only the current state vector of neighbor j. We also assume that agent i knows only a pair of real-valued matrices $(A_i^{n_i \times n}, b_i^{n_i \times 1})$. The problem of interest is to devise local algorithms, one for each agent, which will enable all m agents to iteratively compute the same

☆ Proofs of some results in this paper are not included due to space limitations and can be found in Liu, Morse, Nedić, and Başar (2017). The material in this paper was partially presented at the 53rd IEEE Conference on Decision and Control, December 15–17, 2014, Los Angeles, CA, USA. This paper was recommended for publication in revised form by Associate Editor Antonis Papachristodoulou under the direction of Editor Christos G. Cassandras.

E-mail addresses: jiliu@illinois.edu (J. Liu), as.morse@yale.edu (A. Stephen Morse), angelia.nedich@asu.edu (A. Nedić), basar1@illinois.edu (T. Başar).

http://dx.doi.org/10.1016/j.automatica.2017.05.004
0005-1098/

solution to the linear equation $Ax = b$ where

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix}_{\bar{n}\times n}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{\bar{n}\times 1}$$

and $\bar{n} = \sum_{i=1}^{m} n_i$. We assume that $Ax = b$ has at least one solution, unless stated otherwise. The algorithm presented in this paper works for both the case when $Ax = b$ has a unique solution and the case when $Ax = b$ has multiple solutions. For the case when $Ax = b$ does not have a solution, the algorithm can be modified to obtain a least squares solution via a centralized initialization step (see Section 5.4).

Recently, a distributed algorithm was proposed in Mou, Liu, and Morse (2013) for the synchronous version of the problem just formulated, and with slight modification, that is for a restricted asynchronous version of the problem in which transmission delays are not taken into account. A more general asynchronous version of the problem in which transmission delays are explicitly taken into account was addressed in Liu, Mou, and Morse (2013).

The synchronous version of the problem considered here can be viewed as a distributed parameter estimation problem (Bolognani, Favero, Schenato, & Varagnolo, 2010; Kar, Moura, & Ramanan, 2012; Xiao, Boyd, & Lall, 2005). One approach to the problem is to reformulate it as a distributed convex optimization problem, which has a rich literature (Boyd, Parikh, Chu, Peleato, & Eckstein, 2010; Chang, Nedić, & Scaglione, 2014; Duchi, Agarwal, & Wainwright, 2012; Gharesifard & Cortés, 2014; Jakovetić, Moura, & Xavier, 2014; Nedić & Olshevsky, 2015; Nedić & Ozdaglar, 2009; Nedić, Ozdaglar, & Parrilo, 2010; Shi, Ling, Wu, & Yin, 2014; Wang & Elia, 2014; Zanella, Varagnolo, Cenedese, Pillonetto, & Schenato, 2012). An alternative approach to the problem is to view it as a constrained consensus problem (Lin & Ren, 2014; Liu, Nedić, & Başar, 2014c; Nedić et al., 2010). A similar problem with more restrictive assumptions has been studied in Lu and Tang (2009a, b). The problem is related to classical parallel algorithms such as Jacobi iterations (Margaris, Souravlas, & Roumeliotis, 2007), so-called "successive over-relaxations" (Young, 1950), and the Kaczmart method (Gordon, Bender, & Herman, 1970). The problem is also related to the problem of estimation on graphs from relative measurements in which A is determined by the underlying graph and noisy measurements are taken into account (Barooah & Hespanha, 2007, 2008, 2009).

The differences and advantages of the algorithm in Mou et al. (2013), compared with those in the literature (Bolognani et al., 2010; Lin & Ren, 2014; Nedić & Ozdaglar, 2009; Nedić et al., 2010; Tron & Vidal, 2011; Xiao et al., 2005) (Gharesifard & Cortés, 2014; Gordon et al., 1970; Kar et al., 2012; Lu & Tang, 2009a, b; Margaris et al., 2007; Young, 1950), have been discussed in Liu et al. (2013), Mou et al. (2013) and Mou, Liu, and Morse (2015). Specifically, the algorithm in Mou et al. (2013)

(1) is applicable to any pair of real matrices (A, b) for which $Ax = b$ has at least one solution,
(2) is capable of finding a solution exponentially fast,
(3) is capable of finding a solution for a time-varying directed graph sequence under appropriate joint connectedness,
(4) is capable of finding a solution using at most an n-dimensional state vector received at each clock time from each of its neighbors,
(5) is applicable without imposing restrictive requirements such as (a) the assumption that each agent is constantly aware of an upper bound on the number of neighbors of each of its neighbors or (b) the assumption that all agents are able to share the same time-varying step size.

See Section II in Mou et al. (2015) for details. To the best of our knowledge, there is no distributed convex optimization algorithm which simultaneously satisfies all the above properties. We provide a comparison with competing algorithms in the following table.

Paper	Convergence rate	Neighbor graph
This paper	Exponentially fast	Time-varying, directed
Bolognani et al. (2010)	Exponentially fast	Time-varying, undirected
Nedić et al. (2010)	Exponentially fast	Time-invariant, complete
Nedić and Olshevsky (2015)	$O(\ln t/\sqrt{t})$	Time-varying, directed
Carli, Notarstefano, Schenato, and Varagnolo (2015)	(locally) exponentially fast	Time-invariant, directed
Chang, Hong, Liao, and Wang (2016)	Exponentially fast	Time-invariant, star
Zhang and Kwok (2014)	$O(1/t)$	Time-invariant, star
Zanella et al. (2012)	Not explicit	Time-invariant, undirected

From the table, it can be seen that only the algorithm presented in this paper can solve the problem exponentially fast for time-varying, directed, neighbor graphs. It is worth noting that the idea in Bolognani et al. (2010) can solve the problem for time-varying, directed, neighbor graphs by using double linear iterations which are specifically tailored to the distributed averaging problem when unidirectional communications exist (Liu & Morse, 2012); but the downside of this idea is that the amount of data to be communicated between agents does not scale well as the number of agents increases.

Continuous-time distributed algorithms for the problem in this paper have also received some attention lately; see Anderson, Mou, Morse, and Helmke (2016), Liu, Chen, Başar, and Nedić (2016), Shi and Anderson (2016) and Yang and Tang (2015).

From the preceding discussion, a significant advantage of the algorithm in Mou et al. (2013) over the other existing ones is its capability to solve the problem exponentially fast even when the underlying neighbor graph is directed and time-varying, using only an n-dimensional state vector transmitted between neighboring agents at each clock time. Accordingly, our aim in this paper is to analyze the algorithm proposed in Mou et al. (2013), and particularly to determine the weakest graph-theoretic condition under which the algorithm can solve the distributed linear equation problem exponentially fast. We emphasize exponential convergence because it is robust against certain types of perturbation, analogous to exponential stability of linear systems (Rugh, 1996); it will be clear shortly that the system determined by the algorithm in Mou et al. (2013) is a discrete-time linear time-varying system.

In this paper, we focus on the synchronous version of the problem, but the results derived can be straightforwardly extended to asynchronous versions using the analysis tools in Liu et al. (2013). In Mou et al. (2015), a necessary and sufficient graph-theoretic condition was obtained under a "non-redundant" assumption. Roughly speaking, the set of m agents is non-redundant if a distributed solution to $Ax = b$ cannot be obtained by any proper subset of the full set of m agents; otherwise, the set is redundant. The formal definition is given as follows.

We say that agents with labels in $\mathcal{V} = \{i_1, i_2, \ldots, i_q\} \subset \{1, 2, \ldots, m\}$ are *redundant* if any solution to the equations $A_i x = b_i$ for all i in the complement of $\mathcal{V}$, is a solution to $Ax = b$. To derive an algebraic condition for redundancy, suppose that z is a solution to $Ax = b$. Write $\bar{\mathcal{V}}$ for the complement of $\mathcal{V}$ in $\{1, 2, \ldots, m\}$. Then, any solution w to the equations $A_i x = b_i$, $i \in \bar{\mathcal{V}}$, must satisfy $w - z \in \bigcap_{i\in\bar{\mathcal{V}}} \ker A_i$. Thus, agents with labels in $\mathcal{V}$ will