Distributed Average Consensus with Time-Varying Metropolis Weights ⋆

Lin Xiao [a], Stephen Boyd [b], Sanjay Lall [c]

[a] *Center for the Mathematics of Information, California Institute of Technology, Pasadena, CA 91125-9300, USA*

[b] *Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510, USA*

[c] *Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305-4035, USA*

Abstract

Given a network of processes where each node has an initial scalar value, we consider the problem of computing their average asymptotically using a distributed, linear iterative algorithm. At each iteration, each node replaces its own value with a weighted average of its previous value and the values of its neighbors. We introduce the Metropolis weights, a simple choice for the averaging weights used in each step. We show that with these weights, the values at every node converge to the average, provided the infinitely occurring communication graphs are jointly connected.

Key words: Average consensus; Distributed computing; Metropolis algorithm.

1 Introduction

We consider a set of interconnected processes (nodes of a network) $\mathcal{N} = \{1, 2, \ldots, n\}$, each with an initial scalar value $x_i(0)$. The goal is to (asymptotically) compute the average value, $(1/n)\sum_{i=1}^n x_i(0)$, at each node. The allowed communication pattern of the processes varies with time, and is specified by the set of active edges $\mathcal{E}(t)$ at time $t = 0, 1, 2, \ldots$. The edges are undirected, with $\{i, j\} \in \mathcal{E}(t)$ meaning that processes i and j can communicate with each other at time t. We focus on distributed, linear iterative algorithms of the following form

$$x_i(t+1) = W_{ii}(t)x_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)x_j(t), \qquad (1)$$

for $i = 1, \ldots, n$. Here $W_{ij}(t)$ is the linear *weight* on $x_j(t)$ at node i, and $\mathcal{N}_i(t) = \{j \mid \{i, j\} \in \mathcal{E}(t)\}$ denotes the set of neighbors of node i at time t. The question is how to choose the weights $W_{ij}(t)$ such that every $x_i(t)$ converges to the average of their initial values; i.e.,

$$\lim_{t\to\infty} x_i(t) = (1/n)\sum_{i=1}^n x_i(0), \qquad i = 1, \ldots, n. \qquad (2)$$

This problem falls into a broader class of distributed *consensus* or *agreement* problems in multi-agent coordination and flocking, which have received a fair amount of attention recently; see, e.g., [12,17,24,10,15,18]. In general consensus or agreement problems, the asymptotic values of $x_i(t)$ must be the same for all i, but need not be the average. This relaxed requirement gives more freedom in choosing the weights. Tsitsiklis et al. [21,22] gave a systematic study of agreement algorithms of the type (1) in an asynchronous distributed environment. The recent work [2] summarizes the key results and establishes some new extensions.

Theoretical development on convergence conditions has been a main focus of research on distributed consensus or agreement problems. On the other hand, it is also very important to construct specific weights in applications, especially weights that allow distributed computation and work with time-varying communication graphs. A well-studied method for choosing weights is the *nearest*

⋆ This paper was not presented at any IFAC meeting. The main result in this paper was presented at The 4th International Conference on Information Processing in Sensor Networks, Los Angeles, April 2005.

Corresponding author: Lin Xiao.

Email addresses: lxiao@caltech.edu (Lin Xiao), boyd@stanford.edu (Stephen Boyd), lall@stanford.edu (Sanjay Lall).

neighbor rule proposed in [23]

$$W_{ij}(t) = \begin{cases} 1/(1+d_i(t)) & \{i,j\} \in \mathcal{E}(t) \text{ or } i=j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $d_i(t) = |\mathcal{N}_i(t)|$ is the degree of node i at time t. This method of choosing weights was analyzed in detail in [12], and several variations were considered, e.g., in [15,18]. We note that the nearest neighbor rule does not preserve the average, and the value of asymptotic agreement depends on the initial values $x_i(0)$, as well as the sequence of time-varying communication graphs $\{(\mathcal{N}, \mathcal{E}(t))\}_{t=0}^{\infty}$.

While preserving the average is not necessary for many coordination and agreement tasks, it is an essential requirement for many others. Examples of such applications include distributed load balancing (with divisible tasks) in parallel computers (e.g., [5,3]) and distributed data fusion in sensor networks (e.g., [19,25,20]). In this note, we introduce the *Metropolis weights*, which preserve the average, are easily computed, and guarantee asymptotic average consensus under mild conditions on the sequence of the time-varying communication graphs.

2 Metropolis weights

We define the Metropolis weights on a time varying graph $(\mathcal{N}, \mathcal{E}(t))$ as follows:

$$W_{ij}(t) = \begin{cases} 1/(1+\max\{d_i(t), d_j(t)\}) & \{i,j\} \in \mathcal{E}(t) \\ 1 - \sum_{k \in \mathcal{N}_i(t)} W_{ik}(t) & i=j \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In other words, the weight on each edge is one over one plus the larger degree at its two incident nodes, and the self-weights $W_{ii}(t)$ are chosen so the sum of weights at each node is 1. This method of choosing weights is adapted from the Metropolis algorithm ([13,9]) in the literature of Markov chain Monte Carlo; see also [6,4].

The Metropolis weights are very simple to compute and are well suited for distributed implementation. In particular, each node only needs to know the degrees of its neighbors to determine the weights on its adjacent edges. The weights can computed in any time slot with two rounds of communication between every pair of neighboring nodes. In the first round, each node calculates its degree by counting the number of its (instantaneous) neighbors. In the second round, each node sends its degree information to all its neighbors. The nodes do not need any global knowledge of the communication graph, or even the number of nodes n.

3 Convergence

In this section we state and prove the main convergence result. We will use vector notation to simplify presentation. Let $x(t) = (x_1(t), \ldots, x_n(t))$. The distributed averaging algorithm (1) can be written as

$$x(t+1) = W(t)x(t), \tag{5}$$

where the weight matrix $W(t) \in \mathbf{R}^{n \times n}$ is given by the Metropolis weights (4).

For a finite collection of graphs with a common node set $\mathcal{N}$ and edge sets $\mathcal{E}_k$, $k = 1, \ldots, p$, we call them *jointly connected* if $(\mathcal{N}, \cup_{k=1}^{p} E_k)$ is a connected graph. We note that for a network of n nodes, the number of all possible communication graphs is finite.

Theorem 1 *If the collection of communication graphs that occur infinitely often are jointly connected, then for any $x(0) \in \mathbf{R}^n$, the iteration (5) converges and we have*

$$\lim_{t\to\infty} x(t) = \left((1/n)\mathbf{1}^T x(0)\right)\mathbf{1}, \tag{6}$$

where $\mathbf{1}$ denotes the vector with all components 1.

We make the following remarks on Theorem 1.

- An equivalent, and more concise statement of the convergence condition is that the graph $(\mathcal{N}, \cup_{s\geq t}\mathcal{E}(s))$ is connected for all $t \geq 0$. Similar conditions were established for the convergence of consensus or agreement problems where the common limit may not be the average of the initial values (e.g., [12,2,15,18]).
- Equation (6) is the vector form of (2). With the definition of a t-step transition matrix

  $$\Phi(t) = W(t-1)\cdots W(1)W(0),$$

  we have $x(t) = \Phi(t)x(0)$. Equation (6) holding for all $x(0) \in \mathbf{R}^n$ is equivalent to

  $$\lim_{t\to\infty} \Phi(t) = (1/n)\mathbf{1}\mathbf{1}^T.$$

We prove Theorem 1 by using a convergence result for nonhomogeneous infinite products of *paracontracting matrices* [7], which we explain next. (Theorem 1 will be proved in §3.2.)

3.1 Infinite products of paracontracting matrices

The concept of paracontracting matrices was introduced in [16]. A matrix $M \in \mathbf{R}^{n\times n}$ is called *paracontracting* with respect to a vector norm $\|\cdot\|$ if

$$Mx \neq x \quad \Longleftrightarrow \quad \|Mx\| < \|x\|. \tag{7}$$

It is clear that a symmetric matrix is paracontracting with respect to the Euclidean norm if and only if all its eigenvalues lie in the interval $(-1, 1]$.

For a paracontracting matrix M, let $\mathcal{H}(M)$ denote its fixed-point subspace, i.e., its eigenspace associated with the eigenvalue 1,

$$\mathcal{H}(M) = \{x \in \mathbf{R}^n \mid Mx = x\}.$$

The following is a key result in [7].

Theorem 2 ([7]) *Consider a set of paracontracting matrices $\{W_1, \ldots, W_r\}$. Let $\{i(t)\}_{t=0}^{\infty}$, with $1 \leq i(t) \leq r$, be a sequence of integers, and denote by $\mathcal{J}$ the set of all integers that appear infinitely often in the sequence. Then for any $x(0) \in \mathbf{R}^n$, the sequence of vectors*

$$x(t+1) = W_{i(t)} x(t), \qquad t = 0, 1, 2, \ldots,$$

has a limit $x^* \in \bigcap_{i \in \mathcal{J}} \mathcal{H}(W_i)$.

Intuitively, each paracontracting matrix preserves vectors in its fixed-point subspace, and is contractive for all other vectors. If some paracontracting matrices occur infinitely often in the iterative process, then the limit can only be in the intersection of their fixed-point subspaces. For more background on the convergence of infinite matrix products, see the book [8] and references therein.

3.2 Proof of Theorem 1

To use the result of Theorem 2, we need to prove the following two lemmas.

Lemma 1 *The Metropolis weight matrix $W(t)$ is paracontracting with respect to the Euclidean norm.*

Proof. By the definition (4), $W(t)$ is a (symmetric) stochastic matrix and $W_{ii}(t) \geq 1/n$ for all i. Therefore $W(t) - (1/n)I$ is a nonnegative matrix with row and column sums equal to $(n-1)/n$. This implies that all eigenvalues of $W(t) - (1/n)I$ (all real) have absolute value no larger than $(n-1)/n$ (e.g., [11, §8.1]). Thus all eigenvalues of the symmetric matrix $W(t)$ lie in the interval $[-(n-2)/n, 1]$. This means that $W(t)$ is paracontracting with respect to the Euclidean norm. ∎

Lemma 2 *If a collection of graphs $\mathcal{G}_1, \ldots, \mathcal{G}_p$ are jointly connected, then their corresponding Metropolis weight matrices $W_1, \ldots, W_p$ satisfy*

$$\bigcap_{i=1}^{p} \mathcal{H}(W_i) = \operatorname{span}\{\mathbf{1}\}. \tag{8}$$

Proof. By the definition (4), the Metropolis weight matrices are symmetric and stochastic, so we have $\mathbf{1} \in \mathcal{H}(W_i)$ for $i = 1, \ldots, p$. Therefore

$$\operatorname{span}\{\mathbf{1}\} \subset \bigcap_{i=1}^{p} \mathcal{H}(W_i). \tag{9}$$

Notice that if $W_i x = x$ for $i = 1, \ldots, p$, then we have $(1/p)\sum_{i=1}^{p} W_i x = x$. Therefore

$$\bigcap_{i=1}^{p} \mathcal{H}(W_i) \subset \mathcal{H}\left((1/p)\sum_{i=1}^{p} W_i\right). \tag{10}$$

By assumption, the graphs $\mathcal{G}_1, \ldots, \mathcal{G}_p$ are jointly connected. This implies that the matrix $(1/p)\sum_{i=1}^{p} W_i$ is symmetric, stochastic and *irreducible*. This means that it has a simple eigenvalue 1 with associated eigenvector $\mathbf{1}$ (e.g., [11, §8.4]). In other words,

$$\mathcal{H}\left((1/p)\sum_{i=1}^{p} W_i\right) = \operatorname{span}\{\mathbf{1}\}. \tag{11}$$

Putting the equations (9), (10) and (11) together, we get the desired result (8). ∎

Now we are ready to prove Theorem 1. By Theorem 2, the iteration (5) converges and we have

$$\lim_{t \to \infty} x(t) = c\mathbf{1}$$

where c is a constant depending on the initial condition $x(0)$ and possibly on the sequence of matrices $\{W(t)\}_{t=0}^{\infty}$. However, by the definition (4), every matrix $W(t)$ preserves the average. Thus the constant c can only be $(1/n)\mathbf{1}^T x(0)$, which is independent of the sequence of weight matrices. This finishes the proof for Theorem 1.

3.3 On convergence rate

Under the assumption of Theorem 1, the convergence of the sequence $x(t)$ may not be geometric. In other words, there may not exist a constant $0 < \gamma < 1$, such that

$$\|x(t) - c\mathbf{1}\|_2 \leq \gamma^t \|x(0) - c\mathbf{1}\|_2, \qquad t = 0, 1, 2, \ldots \tag{12}$$

where $c = (1/n)\mathbf{1}^T x(0)$. In particular, as an upper bound of γ, the *joint spectral radius* of the matrices $W_k - (1/n)\mathbf{1}\mathbf{1}^T$, $k = 1, \ldots, r$, may not be strictly less than one (see, e.g., [2]).

To have geometric convergence, we have to impose more conditions on how often joint connectedness occurs. For example, if there exists an integer $T > 0$ such that the graph $(\mathcal{N}, \cup_{t \leq \tau < t+T} \mathcal{E}(\tau))$ is connected for all t, then it can be shown that there exist a γ strictly less than one such that (12) holds.

4 Concluding remarks

In this note we have used a result on infinite products of paracontracting matrices to prove the convergence of the node values using Metropolis weights. For this particular rule of choosing weights, convergence can also be established by applying techniques developed in [21,22]. The main idea is to show that the sequences $\max_i x_i(t)$ and $\min_i x_i(t)$ are monotone decreasing and increasing, respectively, and their difference converges to zero. The proof relies on the fact that the matrices are stochastic; thus nonnegativity of the weights are essential to obtain contraction properties by taking convex combinations. See [1, §7.3.1] for a simplified version and [2] for a overview of the technique. For symmetric weight matrices, however, the technique of paracontracting matrices can be more powerful in certain respects. For example, it may establish convergence in cases where some of the edge weights are negative. As shown in [24], for a fixed graph, the fastest convergence is often obtained when some of the weights are negative. We could imagine that similar things can happen for time-varying graphs.

The simple model for average consensus used in this paper does not include communication delays. If there is no specific requirement on the final value of agreement, communication delays can be readily handled in the framework of asynchronous and distributed computing developed in [21,22,1]. In this case, the final value of agreement in general depends on the initial condition, the sequence of communication graphs, as well as the sequences of communication delays at every node. Inclusion of communication delays makes it very hard to preserve the average, and more sophisticated interprocess protocols are needed for distributed average consensus. One promising approach in this direction is the recent work on *consensus propagation* [14].

References

[1] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[2] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005.

[3] J. Boillat. Load balancing and Poisson equation in a graph. *Concurrency: Practice and Experience*, 2:289–313, 1990.

[4] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review, problems and techniques section*, 46(4):667–689, 2004.

[5] G. Cybenko. Load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

[6] P. Diaconis and L. Saloff-Coste. What do we know about the Metropolis algorithms. *Journal of Computer and System Sciences*, 57:20–36, 1998.

[7] L. Elsner, I. Koltracht, and M. Neumann. On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data. *Linear Algebra Appl.*, 130:65–82, 1990.

[8] D. Hartfiel. *Nonhomogeneous Matrix Products*. World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.

[9] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[10] Y. Hatano and M. Mesbahi. Agreement over random networks. In *Proceedings of 43rd IEEE Conference on Decision and Control*, pages 2010–2015, Atlantis, Paradise Island, Bahamas, December 2004.

[11] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[12] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions Automatic Control*, 48(6):988–1001, June 2003.

[13] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[14] C. Moallemi and B. Van Roy. Consensus propagation. Submitted for publication, 2006.

[15] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50:169–182, 2005.

[16] S. Nelson and M. Neumann. Generalization of the projection method with applications to SOR theory for Hermitian positive semidefinite linear systems. *Numerische Mathematik*, 51:123–141, 1987.

[17] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[18] W. Ren and R. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50:655–661, 2005.

[19] D. Scherber and H. Papadopoulos. Locally constructed algorithms for distributed computations in ad-hoc networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 11–19, Berkeley, CA, April 2004. ACM Press, New York.

[20] D. Spanos, R. Olfati-Saber, and R. Murray. Distributed sensor fusion using dynamic consensus. In *Proceedings of The 16th IFAC World Congress*, Prague, July 2005. Elsevier.

[21] J. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, Massachusetts Institute of Technology, 1984.

[22] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, September 1986.

[23] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, 1995.

[24] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[25] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, pages 63–70, Los Angeles, California, USA, April 2005.