

DISTRIBUTED AVERAGE CONSENSUS WITH INCREASED CONVERGENCE RATE

Boris N. Oreshkin, Tuncer C. Aysal and Mark J. Coates

Department of Electrical and Computer Engineering, McGill University

ABSTRACT

The average consensus problem in the distributed signal processing context is addressed by linear iterative algorithms, with asymptotic convergence to the consensus. The convergence of the average consensus for an arbitrary weight matrix satisfying the convergence conditions is unfortunately slow restricting the use of the developed algorithms in applications. In this paper, we propose the use of linear extrapolation methods in order to accelerate distributed linear iterations. We provide analytical and simulation results that demonstrate the validity and effectiveness of the proposed scheme. Finally, we report simulation results showing that the generalized version of our algorithm, when a grid search for the unknown optimum value of mixing parameter is used, significantly outperforms the optimum consensus algorithm based on weight matrix optimization.

Index Terms— distributed signal processing, average consensus, linear prediction.

1. INTRODUCTION

A major drawback of the developed average consensus algorithms [1–3] is the number of iterations required to converge to consensus often refraining the use of them in practical scenarios. Much of the research addressing with consensus algorithm acceleration has been conducted by Boyd *et. al.* [1, 2]. They showed that it is possible to formulate the problem of asymptotic convergence time minimization as a convex semidefinite weight matrix optimization problem. The disadvantages of this approach are twofold. First, the approach is based on a convex optimization paradigm and the time or computational resources necessary to set up the network may be substantial. Second, this approach requires the connectivity pattern to be known in advance and thus assumes that there is a fusion center or some distributed mechanism that is aware of the global network topology. To combat the second problem, iterative optimization using the subgradient algorithm is proposed in [2]. However, the resulting algorithm is extremely demanding in terms of time, computation, and communication. Another approach to weight matrix optimization, called “best constant” [1], is to set the neighboring edge weights to a constant and optimize this constant to achieve maximum possible convergence rate. The subopti-

mality of the best constant weight matrix stems from the fact that all the edge weights are constrained to be the same.

In this paper, we propose accelerating the convergence rate of distributed average consensus by using a convex combination of the values obtained by a linear predictor and the standard consensus operation. Unlike previous methods, we do not burden the nodes with extra computational load since the prediction is linear and its parameters can be calculated offline. We present a general framework for accelerating the consensus, but focus on a special case to gain further insight. For this case, we derive the optimal convex combination parameter that maximizes asymptotic convergence rate. The optimal parameter requires knowledge of the second largest and the smallest eigenvalues of the weight matrix. We therefore derive suboptimal solutions that need much less information and are easily implementable in practical scenarios. Finally, we assess the performance of the general proposed algorithm via simulations.

The remainder of this paper is organized as follows. Section 2 introduces the distributed average consensus problem. Section 3 details the proposed algorithm, along with its properties and the optimal mixing parameter for the special case. It also specifies the achieved improvement in rate of convergence, and describes practical suboptimal solutions. We report the results of numerical simulations testing the proposed algorithms in Section 4. Section 5 concludes the paper.

2. PROBLEM STATEMENT

We define a graph $G = (\mathcal{V}, \mathcal{E})$ as 2-tuple, consisting of a set \mathcal{V} with $|\mathcal{V}| = N$ vertices, where $|\cdot|$ denotes the cardinality, and a set \mathcal{E} of edges. We denote an edge between vertices i and j as an unordered pair $(i, j) \in \mathcal{E}$. We assume connected network topologies and the connectivity pattern of the graph is given by the $N \times N$ adjacency matrix $\Phi = [\Phi_{ij}]$, where $\Phi_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $\Phi_{ij} = 0$ otherwise. Moreover, we denote the neighborhood of the node i by $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The degree of the node i is denoted $d_i \triangleq |\mathcal{N}_i|$.

We consider a set of nodes of a network, each with an initial real valued scalar $x_i(0)$, where $i = 1, 2, \dots, N$. Let $\mathbf{1}$ denote the vector of ones. The goal is to develop a distributed iterative algorithm that computes, at every node in the network, the value $\bar{\mathbf{x}} \triangleq (N)^{-1} \mathbf{1}^T \mathbf{x}(0)$. In standard distributed consensus, at each step, each node updates its state with a

linear combination of its own state and the states at its neighbors, $\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t)$ where $\mathbf{x}(t)$ denotes the network state vector and \mathbf{W} is a weight matrix that is constructed to satisfy topology constraint, that is $W_{ij} = 0$ whenever $\Phi_{ij} = 0$. The weight matrix, \mathbf{W} , needs to satisfy the following conditions to ensure asymptotic average consensus [1]:

$$\mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \rho(\mathbf{W} - \mathbf{J}) < 1 \quad (1)$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix $\rho(\mathbf{W}) \triangleq \max_i \{|\lambda_i| : i = 1, 2, \dots, N\}$. Here $\{\lambda_i\}_{i=1}^N$ denote the eigenvalues of \mathbf{W} . In this paper, without loss of generality, we assume that, in the modulus, the second largest eigenvalue of the weight matrix is $\lambda_{(2)}$, i.e., $\lambda_{(2)} > |\lambda_{(N)}|$, where $\lambda_{(i)}$ denotes the i -th ranked eigenvalue.

3. PREDICTOR-BASED DISTRIBUTED AVERAGE CONSENSUS (PBDAC)

We modify the above consensus algorithm to increase its convergence speed in the following way:

$$x_i^W(t) = \mathbf{w}_i\mathbf{x}(t-1), x_i^P(t; k) = \Theta^T \mathbf{z}_i(t), \quad (2a)$$

$$x_i(t) = \alpha x_i^P(t) + (1 - \alpha)x_i^W(t). \quad (2b)$$

where \mathbf{w}_i denotes the i -th row of \mathbf{W} . Here the state value $x_i(t)$ is a convex combination of the values $x_i^W(t)$ and $x_i^P(t; k)$, i.e., $\alpha \in (0, 1)$. Note that $\mathbf{x}^W(t)$ results from the standard consensus procedure. Moreover, $x_i^P(t) \equiv x_i^P(t; k)$ is a k -step prediction of future node states obtained from M previous local node states stacked into the vector $\mathbf{z}_i(t) = [x_i(t-M+1), \dots, x_i(t-1), x_i^W(t)]^T$. It is shown that using linear least squares techniques that optimum predictor weights can be expressed as $\Theta = \mathbf{A}^\dagger \mathbf{t}^{P,k}$, where \mathbf{A}^\dagger is the Moore-Penrose pseudoinverse of \mathbf{A} , and \mathbf{A} and $\mathbf{t}^{P,k}$ have the following form [4],

$$\mathbf{A} \triangleq \begin{bmatrix} 1 & \cdots & 1 & 1 \\ 1 & \cdots & M-1 & M \end{bmatrix}^T \quad \mathbf{t}^{P,k} \triangleq \begin{bmatrix} M+k \\ 1 \end{bmatrix} \quad (3)$$

Notice from (2a) that $x_i^P(t)$ is a linear combination of M previous local consensus values. Thus the consensus acceleration mechanism outlined in equations (2a-2b) is fully local if it is possible to find the optimum value of α in (2b) that without global knowledge. The parameters in Θ can be calculated offline for any M, k pair.

In the general case, the analysis of the proposed algorithm is complicated (we assess the performance of the algorithm at its most general case and at its full potential with simulations in Section 4). In order to gain further insight into the algorithm's performance we analyze an important case when the algorithm (2) is based on one step extrapolator of node state operating on two previous node states, i.e., $k = 1$ and $M = 2$. In this case $\Theta = [-1, 2]^T$ hence $x_i^P(t)$ is expressed as follows:

$$x_i^P(t) = 2x_i^W(t) - x_i(t-1). \quad (4)$$

Substituting (4) into (2b) we get the following expression for $\mathbf{x}(t)$ in matrix form as:

$$\mathbf{x}(t) = \mathbf{W}[\alpha]\mathbf{x}(t-1) \quad (5)$$

where $\mathbf{W}[\alpha]$ is the weight matrix modified by the proposed predictor based distributed average consensus (PBDAC) algorithm:

$$\mathbf{W}[\alpha] \triangleq (1 + \alpha)\mathbf{W} - \alpha\mathbf{I}. \quad (6)$$

The following proposition describes some properties of $\mathbf{W}[\alpha]$. Proofs are omitted due to space constraints but can be found in [4].

Proposition 1 *Suppose \mathbf{W} satisfies the necessary conditions for the convergence of the standard consensus algorithm. Moreover, let $\lambda_{(1)} \geq \lambda_{(2)} \geq \dots \geq \lambda_{(N)}$ denote the eigenvalues associated with eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$ and let $\lambda_{(i)}[\alpha]$ denote the ranked eigenvalues of $\mathbf{W}[\alpha]$.*

(i) *If $\lambda_{(N)} \geq 0$, then $\mathbf{W}[\alpha]$ satisfies the required convergence conditions for all α . If $\lambda_{(N)} < 0$, then $\mathbf{W}[\alpha]$ is a doubly stochastic matrix. Moreover, $\rho(\mathbf{W} - \mathbf{J}) < 1$ if*

$$\alpha < (1 - \lambda_{(N)})^{-1}(1 + \lambda_{(N)}). \quad (7)$$

(ii) *$\mathbf{W}[\alpha]$ has the same eigenvectors and its eigenvalues are related to the eigenvalues of \mathbf{W} via the relationship:*

$$\lambda_{(i)}[\alpha] = (1 + \alpha)\lambda_{(i)} - \alpha \quad (8)$$

for any α and i .

In the following, we consider optimization of α to maximize the asymptotic (worst-case) rate of convergence for algorithmic structure (2) and the $M = 2, k = 1$ case.

Theorem 1 *PBDAC with $M = 2$ and $k = 1$ has the fastest asymptotic worst-case convergence rate if*

$$\alpha = \alpha^* \triangleq \arg \min_{\alpha} \rho(\mathbf{W}[\alpha] - \mathbf{J}) = \frac{\lambda_{(N)} + \lambda_{(2)}}{2 - \lambda_{(N)} - \lambda_{(2)}}. \quad (9)$$

To see to what extent the proposed algorithm (2) yields performance improvement over the conventional consensus, we consider the ratio of the spectral radius of corresponding matrices that gives the lower bound on performance improvement $\gamma[\alpha] \triangleq \rho(\mathbf{W} - \mathbf{J})/\rho(\mathbf{W}[\alpha] - \mathbf{J})$.

Proposition 2 *In the optimal case, i.e., when $\alpha = \alpha^*$, the performance improvement factor is given by*

$$\gamma[\alpha^*] = \frac{\lambda_{(2)}(2 - \lambda_{(2)} - \lambda_{(N)})}{\lambda_{(2)} - \lambda_{(N)}}. \quad (10)$$

Although (9) provides an expression for optimum mixing factor resulting in fastest asymptotic convergence rate, the calculation of this optimum value requires knowledge of the second and the last eigenvalues of matrix \mathbf{W} . Therefore it is of interest to derive suboptimum choices for α that result in less performance gain but require considerably less information at the node level.

Proposition 3 *The PBDAC has asymptotic worst-case convergence rate faster than that of conventional consensus if the value of mixing parameter satisfies: $0 < \alpha \leq \alpha^*$.*

To ensure that $\gamma[\alpha] > 1$, and remove the need of weight matrix information, we proceed to derive bounds on α^* satisfying the range defined by Proposition 3. The mentioned constraints indicate that α^* needs to be lower-bounded, which implies that we need to derive a lower bound for $\lambda_{(2)} + \lambda_{(N)}$.

Proposition 4 *If \mathbf{W} satisfies the convergence conditions and its eigenspectrum is a convex function of the eigenvalue index then $\lambda_{(2)} + \lambda_{(N)} \geq \xi$, where*

$$\xi \triangleq \frac{2(\text{tr}(\mathbf{W}) - 1)}{(N - 1)} \quad (11)$$

and $\text{tr}(\cdot)$ denotes the trace of its argument.

Proposition 4 provides an upper bound for the mixing parameter α in terms of the trace of weight matrix \mathbf{W} :

$$\alpha \leq \frac{\xi}{2 - \xi} \triangleq \Lambda(\xi). \quad (12)$$

Centralized calculation of $\text{tr}(\mathbf{W})$ is a far less complicated operation than computing eigenvalues to find the optimum mixing parameter. Moreover, it can be calculated in a distributed fashion. However, the convexity assumption appears to be strong and unnecessary in many cases and the upper bound still requires knowledge of the diagonal elements of \mathbf{W} .

We now consider the special, but important, case of random geometric graphs, which can act as good topological models of wireless sensor networks, one of the promising application domains for consensus algorithms. For this case, we show that there exists an asymptotic upperbound $\Lambda_\infty(\xi)$ for α that can be calculated off-line. The random geometric graph is defined as follows: N nodes (vertices) are distributed in an area \mathcal{D} according to a point process with known spatial distribution $p_{x,y}(x, y)$. Two nodes i and j are connected, i.e. $\Phi_{ij} = 1$, if the Euclidean distance $r_{i,j}$ between them is less than some predefined connectivity radius r_c . The indicator function $I\{r_{i,j}^2 \leq r_c^2\} = 1$ whenever $r_{i,j}^2 \leq r_c^2$ holds.

We consider weight matrices \mathbf{W} constructing according to a rule of the following form:

$$W_{ij} = \begin{cases} I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j), & i \neq j \\ 1 - \sum_{j \in \mathcal{N}_i} W_{ij}, & i = j \end{cases} \quad (13)$$

where $\mathcal{L}(d_i, d_j)$ is some function of the local connectivity degrees d_i and d_j of nodes i and j . For such a graph and weight matrix, the following proposition provides an asymptotic upper bound on the value of mixing parameter α in terms of the expectation of $I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j)$ for the popular Max-degree weights [1].

Proposition 5 *Consider the Max-degree weight design scheme where $\mathcal{L}(d_i, d_j) \triangleq N^{-1}$ for $(i, j) \in \mathcal{E}$ and $W_{ii} = 1 - \mathcal{N}_i/N$, then, we have*

$$\xi_\infty \triangleq \lim_{N \rightarrow \infty} \xi \xrightarrow[N \rightarrow \infty]{a.s.} 2(1 - E\{\zeta\}) = 2(1 - p) \quad (14)$$

where $E\{\zeta\} = p$ is the probability that two nodes in the network are connected.

The value p can be analytically derived for a given connectivity radius and distribution of the nodes [4]. Substituting this into (12) gives

$$\alpha \leq \frac{1 - p}{p}. \quad (15)$$

This provides an intuition indicating that, for highly (sparsely) connected graphs, i.e., large (small) p , a small (large) α is favored. In turn, a small (large) α implies more (less) weight is associated with the predictor output.

4. NUMERICAL EXAMPLES

We consider a set of $N = 50$ nodes uniformly distributed on the unit square. The nodes establish bidirectional links to each other if the Euclidean distance between them is smaller than the connectivity radius, $\sqrt{\log N/\bar{N}}$. Initial node measurements are generated as $\mathbf{x} = \theta + \mathbf{n}$ where $\theta = 1$ and \mathbf{n} is Gaussian distributed with $\sigma = 1$. Then, we regularize the data such that $\bar{\mathbf{x}} = 1$. First of all, we compare the convergence time results of the algorithm we propose for the theoretically analyzed $M = 2$ and $k = 1$ case, against the algorithms presented in [1, 2]. Let us introduce the following notation: MD and MH are the standard consensus algorithms based on maximum degree and Metropolis-Hastings weight matrices [1, 2], respectively; OPT is the optimum and BC the best constant algorithm from [1, 2]; MD-OM, MD-SM, and MD-SAM denote the PBDAC using the maximum degree weight matrix and optimum α , suboptimum α chosen according to the trace bound (12), and suboptimum α chosen according to the asymptotic bound (15), respectively; MH-OM and MH-SM denote the PBDAC using Metropolis-Hastings weight matrix, optimum α and suboptimum α chosen according to trace bound (12), respectively. In this notation M is an integer number showing how many past samples are used in the predictor.

Figure 1 compares the convergence times of the algorithms. Here the algorithm (2) is simulated for the case that was analyzed analytically in Section 3, i.e., $M = 2$ and $k = 1$. It is clear from Fig. 1 that although our algorithm is extremely simple and does not require any global optimization, it performs almost as well as the optimum algorithm from [1, 2]. It outperforms the best constant algorithm when used in conjunction with the Metropolis-Hastings weight matrix. When Max-degree is utilized in the proposed algorithm, its asymptotic convergence time is very similar to that of the optimized

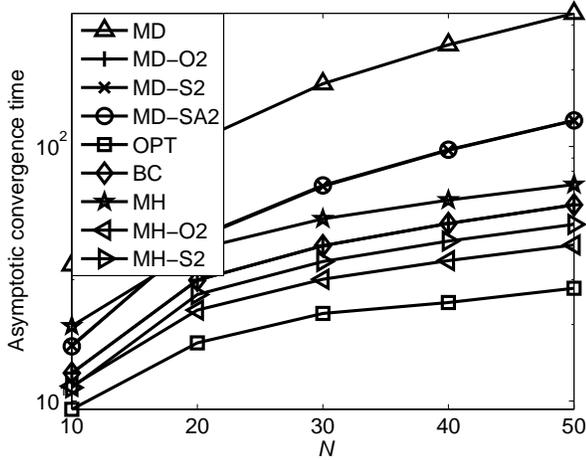


Fig. 1. Asymptotic convergence time for, MD: \triangle , MD-O2: $+$, MD-S2: \times , MD-SA2: \circ , BC: \diamond , OPT: \square , MH: \star , MH-O2: \triangleleft , MH-S2: \triangleright .

best constant approach from [2]. Moreover, the solution obtained using Proposition 5 performs similarly to the suboptimal solution derived using knowledge of the trace of \mathbf{W} .

Figure 2 shows the evolution with iteration of the mean-squared error (MSE)

$$\text{MSE}(t) = \frac{1}{N} \sum_{i=1}^N (x_i(t) - \bar{x})^2 \quad (16)$$

for PBDAC and (optimized) standard consensus algorithms. The PBDAC approaches employ Metropolis–Hastings weight matrices. The intent is to depict the potential of PBDAC, so in order to obtain the results for our algorithm when $M = 3$ and $k = 1$ we used a 0.1 grid for the unknown parameter α and evaluated the MSE of our algorithm at every value of α during each of the trials. For each trial, we selected the α that achieved the minimum MSE at iteration 50. The results are ensemble average of 500 trials. We draw two conclusions from Fig. 2: (i) the performance of our algorithm with $M = 2$ and $k = 1$ is very close to the optimum in terms of step-wise MSE decay; and (ii) our algorithm with $M = 3$ and $k = 1$ has the potential to significantly outperform the worst-case optimum algorithm [1, 2] in terms of step-wise MSE decay. The performance depicted in Fig. 2 is not currently attainable for the $M = 3$ case, because we do not have a method for identifying the optimum α . However, the results underline the enormous potential for improvement if an optimum mixing parameter for the case $M = 3$ can be specified analytically.

We have also conducted simulations evaluating the performance of the algorithm for varying $M \in \{2, 3, \dots, 6\}$ and $k \in \{1, 2, 3\}$. The best results are obtained for the case shown here, *i.e.*, $M = 3$ and $k = 1$. Varying k has little effect on the proposed algorithm’s performance. The results, not depicted here, can be found in [4].

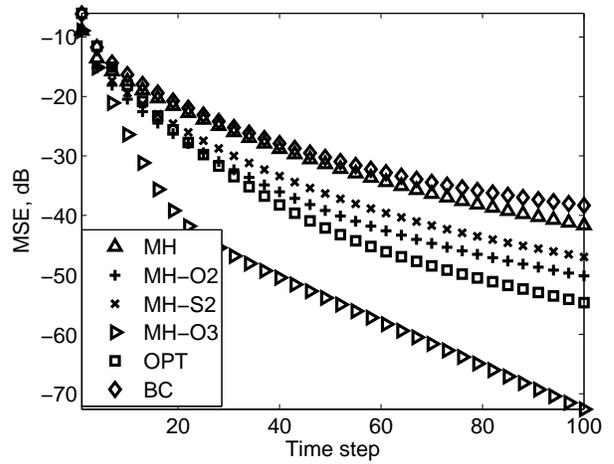


Fig. 2. MSE versus time step for, MH: \triangle , MH-O2: $+$, MH-S2: \times , MH-O3: \triangleleft , BC: \diamond , and OPT: \square .

5. CONCLUDING REMARKS

We have presented a general, predictor-based framework to improve the rate of convergence of distributed average consensus algorithms. In contrast to previous acceleration approaches, the proposed algorithm is simple, fully linear, and parameters can be calculated offline. To gain further insight into the proposed algorithm, we focused on a special case, presenting theoretical and simulation results. In its most simple case, the proposed algorithm outperforms the optimal best constant algorithm from [1] and performs almost as well as the worst-case-optimal design algorithm of [2]. Simulation studies shows that the proposed algorithm has the potential to outperform significantly the worst-case-optimal algorithm, but in order for this potential to be realized, we must devise a scheme for specifying an optimal (or close-to-optimal) α value for the case $M > 2$, and this is the subject of our current research efforts.

6. REFERENCES

- [1] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
- [3] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Distributed sensor fusion using dynamic consensus,” in *Proceedings of the 16th IFAC World Congress*, 2005.
- [4] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, “Accelerated distributed average consensus via localized node state prediction,” ECE Dept., McGill University, Tech. Rep., Oct. 2007, www.tsp.ece.mcgill.ca/Networks/Publications.html.