

CONSENSUS-TRACKING IN DISTRIBUTED NETWORKS BY ONE-HOP AVERAGING

Kevin Topley, Vikram Krishnamurthy*

George Yin†

University of British Columbia
Department of Electrical and Computer Engineering
British Columbia, V6T 1Z4, Canada

Wayne State University
Department of Mathematics
Detroit, MI 48202, USA

ABSTRACT

For a connected network of sensors we consider deriving the linear update weights required by a 1-hop distributed linear averaging algorithm (denoted 1-DLA) such that average-consensus is reached when the sensor nodes simultaneously track, by linear stochastic approximation, a set of distinct Markov chains with time-varying regime. It is found the desired consensus is infeasible for any 1-hop 1-DLA type algorithm in this setting, which includes the consensus filter proposed in [2]. However, assuming a symmetric communication graph we show the average-consensus can be approached with zero asymptotic error by an alternative 1-hop algorithm (denoted 4-DLA) that requires each sensor compute 4 estimates $\{\hat{\pi}, s, s^0, \hat{s}\}$ rather than only $\{s\}$ as required under 1-DLA. We demonstrate a simulation of 4-DLA and explain its advantages compared to alternative multi-hop algorithms.

Index Terms— distributed averaging, stochastic approximation, 1-hop algorithm, sensor network, consensus formation

1. INTRODUCTION

A significant amount of research (e.g. [1]) has explored the properties of the “static” consensus algorithm

$$s_{k+1}^i = s_k^i + \sum_{j=1}^n \mathcal{W}_{ij}(s_k^j - s_k^i), \quad i = 1, \dots, n, \quad (1.1)$$

in regard to having each sensor state-value $s^i \in \mathbf{R}$ reach the average-consensus point $\bar{s}_0 = \frac{1}{n} \sum_{i=1}^n s_0^i$, where $\mathcal{W}^v \in \mathbf{R}^{n \times n}$ represents the communication links of network with n nodes. The sensor communication network can be formally defined as a graph $\mathcal{G} = \{\mathcal{V}(s_0), \mathcal{E}, \mathcal{W}^v\}$ where s_0 denotes a vector of initial values for the set of nodes $\mathcal{V} = \{1, \dots, n\}$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the graph edge set, and \mathcal{W}^v the corresponding weighted adjacency matrix that we assume satisfies the

following constraints,

1. $\mathcal{W}_{ij}^v = 0 \Leftrightarrow (i, j) \notin \mathcal{E}$, (denoted $\mathcal{W}^v \in \mathcal{E}$),
2. $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$,
3. $\bigcap_{i \in \mathcal{V}} \bigcup_{j \in \mathcal{V}} \{j : (i, j) \in \mathcal{E}\} = \mathcal{V}$.

The above 3 conditions are not necessary for the distributed algorithm (1.1) to result in the average-consensus \bar{s}_0 , and indeed many works explore the consensus ability of (1.1) under stochastic or time-varying edge sets, averaging weights, or in the presence of communication noise. Other works employ (1.1) as a means to achieve consensus when each sensor observes a common signal in Gaussian noise, as well as other tracking models (see [2] and references therein).

Main Results. We show here (1.1) can ensure all sensor state-values s_k weakly-converge to the average-consensus regarding the local sensor tracking estimates when assuming each sensor observes an ergodic Markov chain with time-varying regime. To introduce the general scenario of this consensus problem, we consider the consensus-tracking ability of multi-hop algorithms that are more direct than (1.1) but also require an increase in size of communicated data as well as the data-storage capacity of each sensor.

1.1. Multi-Hop Average-Tracking

Let each sensor $i \in \mathcal{V}$ locally observe a unique time-varying parameter $X^i \in \mathbb{R}^S$ and produces a filtered estimate $\hat{X}^i \in \mathbf{R}^S$. The most direct and ubiquitous way to distributively average the \hat{X}^i across n sensor nodes would be for each node to transmit, as frequently as possible, its current estimate to all neighboring nodes, along with a sensor identification number (i.e. their enumeration $i = 1, \dots, n$). The receiving nodes could then relay the same estimate they just received, plus their own current estimate, to all of their neighboring nodes, and so on.

If each node contains a vector V of nS elements such that each sensor in the network is allocated uniquely to a fixed set of S elements in V , then upon reception of a locally filtered estimate and the associated sensor identification number, each receiving sensor could simply replace the respective S elements of V . Supposing each sensors local estimate

*Email: ktopley, vikramk@ece.ubc.ca.

†Email: gyin@math.wayne.edu

\hat{X}^i converges (in some optimal sense) to the true parameter X^i , the S -dimensional average \bar{V} across the n subsets of V would then result in the best possible distributed estimate of the current observed parameter averaged across all nodes $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_0^i$. It is clear this algorithm results in an average-consensus modulated by the local sensor estimates, and we classify it as a $(n - 1)$ -hop n -DLA type algorithm since each sensor must *store* n different local estimates, and also each of those estimates may need to be relayed up to a maximum of $(n - 1)$ iterations to reach every node in a network.

The major drawback of this algorithm is that as n increases, both the sensor storage capacity and the number of communicated estimates per iteration must increase in direct proportion. Ideally the size of all data transferred between sensors is kept minimal, thus motivating the extreme opposite of $(n - 1)$ -hop n -DLA, that is 1-hop 1-DLA type algorithms (to which many works have been related, e.g. [2]). 1-hop 1-DLA algorithms require each sensor combine all received estimates into a single average and store only that, thus involving a minimal data transfer yet also requiring, for average-consensus, a more detailed specification of how to update (i.e. linearly weight) each sensors stored estimate with those it receives from its neighbors. For arbitrary $\ell_1, \ell_2 \in [2, \dots, n - 1]$ an ℓ_1 -hop ℓ_2 -DLA algorithm can be seen as a mixture of the extreme cases 1 and n . Values of ℓ closer to n may be beneficial if sensors can readily store and communicate large amounts of data, or when reliance on specific update weights cannot be justified. Conversely, 1-hop 1-DLA is advantageous when the storage capacity of each sensor is low or when the maximum data size that can be communicated per iteration is limited.

We note that for well-designed edge sets, the values (ℓ_1, ℓ_2) may be relatively small compared to n , thus implying ℓ_1 -hop ℓ_2 -DLA could in some cases be an optimal consensus algorithm.

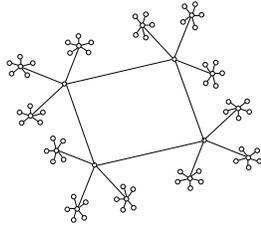


Fig. 1. A graph of 76 nodes and 76 edges. The graph design is such that a 2-hop 4-DLA over the inner loop could achieve average-consensus faster than 1-hop 1-DLA. However, 1-hop 1-DLA is also efficient under the correct edge weights [3].

It would seem, however, that for well-designed edge sets when $\ell_1 = \ell_2 = 1$ the optimal weights for fast consensus ensure a marked increase in speed of convergence (see [3]), thus leading to an interesting problem of efficiency when the

edge set is poorly designed or arbitrary, see Fig.1-2.

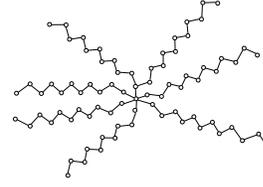


Fig. 2. A graph of 76 nodes and 76 edges. The poor graph design implies 1-hop 1-DLA may be the more practical choice, although it is relatively slow compared to that under Fig.1.

1.2. 1-Hop 1-DLA Observation and Tracking Model

As a specific observation model for the sensors individual estimates, we assume the sensors observe a set of “fast” Markov chains $\{X^1, \dots, X^n\}$ with aperiodic and irreducible transition matrices $A^i(\theta) = (a_{lj}^i(\theta))$ and a common state-space $\mathcal{S} = \{e_1, \dots, e_S\}$ with finite dimension S ,

$$a_{lj}^i(\theta) = P(X_{k+1}^i = e_j | X_k^i = e_l, \theta_k = \theta). \quad (1.2)$$

The parameters $X^i \in \mathbf{R}^{S \times 1}$ are fast in the sense that their transition matrices remain unscaled on the time-scale $dt = O(\mu)$, in the limit as μ approaches zero. We condition the transition matrices $A^i(\theta)$ on the state of a “slow” Markov chain θ with finite state-space $\mathcal{M}_\theta = \{\theta^1, \dots, \theta^m\}$ and transition matrix,

$$P_\epsilon = I + \epsilon Q, \quad (1.3)$$

where ϵ is a rate parameter of order $O(\mu)$, and Q is the generator of a continuous-time Markov chain θ_t . From (1.3) it is clear that, unlike $A^i(\theta)$, the transition matrix P_ϵ approaches identity as μ approaches zero and thus scales on the order of the time differential dt . For this reason the parameters θ and X^i evolve on two distinct time-scales in the limit as $\mu \rightarrow 0$.

Upon each observation X_k^i we assume sensor i updates its state-value $s^i \in \mathbf{R}^{S \times 1}$ by the linear stochastic approximation (SA) algorithm,

$$s_{k+1}^i = s_k^i + \mu(X_k^i - s_k^i), \quad s_0^i = X_0^i, \quad k = 0, 1, \dots \quad (1.4)$$

This is a “local” algorithm in the sense that each sensor $i \in \mathcal{V}$ operates under (1.4) irrespective of all other sensors $j, j \in \mathcal{V} \setminus \{i\}$. When μ approaches zero the sequence of iterates $\{s_k\}$ will weakly-converge to a solution $s_{(\cdot)}$ of the switching ODE

$$\frac{ds_t}{dt} = -s_t + \pi(\theta_t), \quad t \geq 0, \quad (1.5)$$

where $\pi(\theta_t) = [\pi^1(\theta_t), \dots, \pi^n(\theta_t)] \in \mathbf{R}^{nS \times 1}$ denotes the vector of stationary distributions of the Markov chains X^1, \dots, X^n , conditional on θ . If we do not assume $\pi^i(\theta) = \pi^j(\theta)$ for each pair (i, j) and $\theta \in \mathcal{M}_\theta$, then by (1.4) alone

each sensor i weakly-converges to $\pi^i(\theta)$ and not the average-consensus $\bar{\pi}(\theta) = \frac{1}{n} \sum_{i=1}^n \pi^i(\theta)$, see [4] and references therein.

To achieve the average-consensus $\bar{\pi}(\theta)$ we first consider a 1-hop 1-DLA with a form similar to the consensus filter proposed in [2]. Specifically we suppose that, at each time iteration $t \in \mathbb{N}$, each sensor $i \in \mathcal{V}$ exchanges both their current state-value s_t^i and observation value X_t^i with each of their neighbors, and computes as their updated state-value a weighted average of their current data. Denoting the averaging weight for sensor i and data (s_t^j, X_t^j) is represented by the $(i, j)^{th}$ element of \mathcal{W}^v and \mathcal{W}^o , respectively, where \mathcal{W}^o is an arbitrarily weighted adjacency matrix of the graph $\mathcal{G}^o = \{\mathcal{V}(s_0), \mathcal{E}, \mathcal{W}^o\}$

The above algorithm together with (1.5) may be expressed by the slow communication dynamics,

$$s_t = \begin{cases} e^{\bar{t}^* - t} (s_{\bar{t}^*} - \pi(\theta_t)) + \pi(\theta_t), & \text{if } t \notin \mathbb{N} \\ (I - \mathcal{L} - \mathcal{D}^o) s_{t-} + \mathcal{W}^o X_k, & \text{if } t \in \mathbb{N} \end{cases} \quad (1.6)$$

where $t_- = \lim_{u \nearrow t} u$, $\bar{t}^* = \max_{t^* \in \mathbb{N}} \{t^* < t\}$, $\mathcal{L} = \mathcal{D}^v - \mathcal{W}^v$, and $\mathcal{D}^p = \text{diag}(\mathcal{W}^p \mathbb{1})$, $p \in \{o, v\}$, and $\mathbb{1}$ denotes a vector of appropriate dimension with unit valued elements. It can be shown that if $L + \mathcal{D}^o$ is positive-definite (PD), then (1.6) implies that as t_-^* increases the sensor state-values s_t approach,

$$\left(\frac{e^1 - I + \mathcal{W}^o}{e^1 - I + \mathcal{L} + \mathcal{D}^o} \right) \pi(\theta_t) = \Lambda_{disc} \pi(\theta_t). \quad (1.7)$$

For this reason, (1.6) may be seen to be equivalent to,

$$s_{k+1} = (I - \mu \mathcal{L} - \mu \mathcal{D}^o) s_k + \mu \mathcal{W}^o X_k, \quad (1.8)$$

in the sense that, as μ approaches zero, the sensors under (1.8) approach

$$\left(\frac{\mathcal{W}^o}{\mathcal{L} + \mathcal{D}^o} \right) \pi(\theta_t) = \Lambda_{cont} \pi(\theta_t) \quad (1.9)$$

which is of the same form as (1.7) with respect to the weight matrices $\{\mathcal{W}^v, \mathcal{W}^o\}$ when $(L + \mathcal{D}^o)$ is PD. For simplicity, however, we refer only to (1.8) due the similar expressions used for both the sensor communication and observation algorithms.

2. AVERAGE-CONSENSUS TRACKING

We first note that since each sensor i privately observes X^i , the estimate $\hat{\pi}_t^i = \pi^i(\theta_t)$ can be achieved locally at each sensor without need of communication. For this reason, each sensor i need not distributively obtain the *true* average-consensus $\bar{\pi}(\theta_t)$, but rather only some average in which all *other* sensor estimates s_t^j , $j \in \mathcal{V} \setminus \{i\}$, are equally weighted. Specifically, if either (1.8) or (1.6) ensures each sensor attains some linear combination ℓ^i of $\{\pi^j(\theta_t), j = 1, \dots, n\}$ of the form

$$\ell^i = \beta_i \pi^i(\theta_t) + \beta^i \sum_{j \neq i} \pi^j(\theta_t), \quad (2.10)$$

for arbitrary constants $(\beta_i, \beta^i \neq 0)$, $i = 1, \dots, n$, then assuming sensor i knows $\{\beta^i, \beta_i, \ell^i, \pi^i(\theta_t), n\}$, this sensor can compute the true value of $\bar{\pi}(\theta_t)$ locally by the simple linear average,

$$\bar{\pi}(\theta_t) = \frac{1}{n} [(1/\beta^i) \ell^i + (1 - \beta_i/\beta^i) \pi^i(\theta_t)]. \quad (2.11)$$

This rationale then renders the following result significant.

Lemma 2.1 *For any connected graph with a symmetric weight matrix \mathcal{W}^v , if $\mathcal{W}^o = -\mathcal{L} + mI$ then $\lim_{m \rightarrow 0} \Lambda_{cont} = \frac{2}{n} \mathbb{1} \mathbb{1}' - I$.*

Proof. Since \mathcal{G} is connected the Laplacian $\mathcal{L} = \mathcal{D}^v - \mathcal{W}^v$ will have an eigendecomposition UJU^{-1} where J is diagonal and has only its first diagonal element equal to zero (that is \mathcal{L} has an isolated eigenvalue at zero [3]). If we set $\mathcal{W}^o = -\mathcal{L} + mI$ then Λ takes the form,

$$\Lambda = U \left(\frac{-J + mI}{J + mI} \right) U^{-1} = UJ^*U^{-1} \quad (2.12)$$

Since \mathcal{W}^v is symmetric and thus balanced, taking the limit as m approaches zero implies the diagonal matrix J^* will have one element at 1 and all others at -1 (see for instance [5]). We then have $\Lambda = 2/n \mathbb{1} \mathbb{1}' - I$ which implies a sensor estimate at node i of affine form ℓ^i , with $\beta_i = (2/n - 1)$ and $\beta^i = 2/n$. \square

The above lemma suggests each sensor estimate may asymptotically take the form of (2.10). However, when $\mathcal{W}^o = -\mathcal{L} + mI$, taking the limit as m approaches zero affects the actual derivation of (1.7) or (1.9) as the correct expressions for the sensor steady-state under the original algorithms (1.8) (1.6), as we now show.

Lemma 2.2 *For any connected graph with a symmetric weight matrix \mathcal{W}^v , if $\mathcal{W}^o = -\mathcal{L} + mI$ then $\lim_{m \rightarrow 0}$ the algorithm (1.6) implies all sensor state-values have a steady-state conditional on θ_t ,*

$$\lim_{t \rightarrow \infty} s_t = \frac{1}{n} \mathbb{1} \mathbb{1}' s_0 + \left(\frac{1}{n} \mathbb{1} \mathbb{1}' - I \right) \pi(\theta_t). \quad (2.13)$$

Proof. Assuming $\mathcal{W}^o = -\mathcal{L} + mI$, let $W_\mu = I - \mu \mathcal{L} - \mu mI$ and consider the iteration (1.8) after $t^* = t\mu \in \mathbb{N}$ occurrences for some $t \geq 0$,

$$s_{t^*} = W_\mu^{t^*} s_0 + \sum_{l=0}^{t^*} W_\mu^l \mathcal{W}^o X_{t^*-l}. \quad (2.14)$$

Letting m vanish, we then find in the limit μ approaches zero that the coefficient of s_0 becomes $\exp\{-\mathcal{L}t^*\}$ whereas the observation input becomes

$$(e^{-\mathcal{L}t^*} - I) \pi(\theta_{t^*}) \quad (2.15)$$

since X_{t^*-l} can be estimated $\pi(\theta_{t^*})$ in the limit as μ vanishes. If \mathcal{W}^v is symmetric (and thus balanced) then as t^* increases the term $\exp\{-\mathcal{L}t^*\}$ will approach the average-consensus coefficient $\frac{1}{n} \mathbb{1} \mathbb{1}'$, as shown in [5]. \square

The above lemma implies that an average-consensus can be attained by a 1-hop 4-DLA type algorithm in which each sensor node computes the following estimates,

1. $\hat{\pi}_{k+1} = \hat{\pi}_k + \mu(X_k - \hat{\pi}_k)$,
2. s_{k+1} computed as in (1.6) or (1.8)
3. $s_{k+1}^0 = W_\mu s_k^0$, $s_0^0 = s_0$,
4. $\hat{s}_t = s_t - s_k^0 + \hat{\pi}_k$.

(2.16)

The estimate $\hat{\pi}_k^i$ is distinguished from the stationary distribution $\pi^i(\theta_t)$ not only by using a “hat” but also by omitting its dependence on θ . Also for clarity the first three steps of (2.16) have been written in discrete-time, we note that as μ vanishes the estimates $\hat{\pi}_{(\cdot)}$ converge weakly to solutions of (1.5) and $(s_{(\cdot)}^0, s_{(\cdot)})$ follow accordingly [4].

3. SIMULATIONS OF 4-DLA

Letting the communication network be given by a cyclical graph with 9 nodes and unit edge weights, we demonstrate the 4 estimates required of the 4-DLA algorithm (2.16) result in the desired average-consensus $\bar{\pi}(\theta)$. The conditioning parameter θ switches twice during the simulation ($t_1 = 2900\mu$, $t_2 = 5400\mu$, where we take $\mu = 10^{-5}$). The first plot (Fig.3) presents an example of the estimates $\{s_t, (s_t - s_t^0)\}$ based respectively on steps 2 and 3 of (2.16). We note, as expected, the sensor state-value trajectories converge exponentially to their respective steady-states. It is also apparent the initial values s_0 have been set approximately twice as large as the first set of observed values (these values can be seen in Fig.4 as the asymptotic local sensor tracks $\hat{\pi}^i(\theta_t)$, covering the range (0 – 100) for the initial θ value). The linear combination $(s_t - s_t^0)$ is found in Fig.3 to attain a steady-state with affine form ℓ^i (denoted by the horizontal lines). The estimates s_t (based on step 2 alone) remain uninformative due to the residual initial term.

In Figure 4 the local sensor tracks $\hat{\pi}_t^i$ and the final estimate \hat{s}_t are found to converge to the local observed stationary measure and average-consensus $\bar{\pi}(\theta_t)$, respectively.

4. CONCLUSION

We proposed a viable 1-hop averaging algorithm (4-DLA) under which all sensors connected within a distributed network may form an average-consensus when slowly communicating their current estimates and locally observed parameter values. This task can be seen infeasible under a 1-hop 1-DLA algorithm such as the distributed consensus filter. Comparisons with a proposed class of multi-hop averaging algorithms were considered to clarify some advantages of 1-hop DLA. Simulated sensor state-value trajectories were shown to illustrate

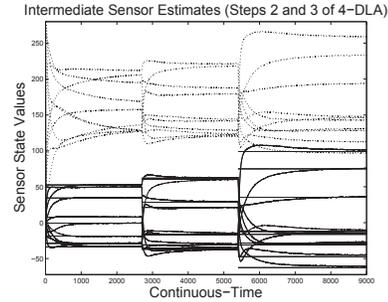


Fig. 3. The estimates $(s_t - s_t^0)$ (solid) and s_t (dotted). The horizontal lines indicate asymptotic steady-states with the required affine form ℓ^i .

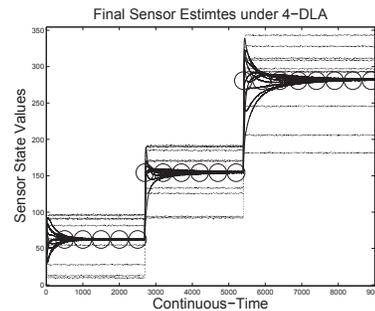


Fig. 4. The final estimates \hat{s} (solid lines) of (2.16). The average-consensus $\bar{\pi}(\theta_t)$ is denoted by the horizontal row of circles, and the local estimates $\hat{\pi}_t^i$ are dotted lines.

how the slowly time-varying average-consensus is reached by 4-DLA.

5. REFERENCES

- [1] M. Porfiri and D. Stilwell, Consensus Seeking Over Weighted Directed Graphs, *IEEE Trans. Automat. Control*, vol. 52(9), 2007, pp.1767-1773.
- [2] R. Olfati-Saber and J.F. Shamma, "Consensus Filters for Sensor Networks and Distributed Sensor Fusion", *Proc. of 44th IEEE Conf. on Deci. and Control*, Spain, 2005.
- [3] S. Boyd and L. Xiao, Fast Linear Iterations for Distributed Averaging, *Syst. and Control Let.*, vol.53, 2004.
- [4] K. Topley, V. Krishnamurthy, and G. Yin, "Consensus Formation in a Switched Markovian Dynamical System", *Proc. of Conf. on Dec. and Cont.*, 2008.
- [5] R. Murray and R. Olfati-Saber, Consensus Problems in Networks of Agents with Switching Topology and Time-Delays, *IEEE Trans. Automat. Control*, vol.49(9), 2004, pp.1520-1532.