Adaptive Consensus Averaging for Information Fusion over Sensor Networks

Mohammad S. Talebi*, Mahdi Kefayati*†, Babak H. Khalaj*, Hamid R. Rabiee*†
*Sharif University of Technology, †Iran Telecommunication Research Center (ITRC), Tehran, Iran.
Email: mstalebi@ee.sharif.edu, kefayati@ce.sharif.edu, khalaj@sharif.edu, rabiee@sharif.edu

***Abstract*— This paper introduces adaptive consensus, a spatio-temporal adaptive method to improve convergence behavior of the current consensus fusion schemes. This is achieved by introducing a time adaptive weighting method for updating each sensor data in each iteration. Adaptive consensus method will improve node convergence rate, average convergence rate and the variance of error over the network. A mathematical formulation of the method according to the adaptive filter theory as well as derivation of the time adaptive weights and convergence conditions are presented. The analytical results are verified by simulation as well.**

I. Introduction

Sensor networks have recently received much attention due to their high potential of the formation of the next generation information gathering and processing systems. The main focus of this paper is on a specific method of distributed sensor fusion for unknown parameter estimation called consensus averaging, and proposes a new spatio-temporal algorithm to improve the convergence behavior of previously proposed methods.

Many schemes for distributed data fusion over sensor networks have been proposed. One simple method is flooding which requires a large amount of data communication, storage memory and book-keeping overhead. Several sophisticated approaches have also been proposed in the context of decentralized detection for a network of mobile agents [1]. Recently, a new class of distributed data fusion is proposed which is used in coordination of agents in a network as well as realization of distributed Kalman filters [2]. The problem of distributed *consensus* or *agreement* among nodes belongs to this class of problems [3] [4]. Xiao *et al.* [5] have proposed a simple iterative method for data fusion in sensor networks based on average consensus, and have applied it to the problem of unknown parameter estimation in the wireless sensor networks. Since the aforementioned method deals only with single-hop transmission, it avoids many overheads incurred in most distributed schemes.

We consider the problem of unknown parameter estimation using a sensor network that consists of n randomly distributed nodes over an area. We assume that each sensor takes a noisy measurement of the unknown parameter, corrupted by additive white gaussian noise (AWGN). In a centralized fusion scheme in sensor networks, each sensor sends its measurement to a center to extract the maximum likelihood estimation of unknown parameter from aggregate measurements of the sensors. This scheme suffers from communication overheads. In a distributed fusion scheme in sensor networks, however, using its own and its neighbors' measurements, each sensor calculates a local estimation of the unknown parameter which iteratively converges to the intended result. The maximum likelihood estimation of the unknown parameter is reduced to an averaging over the measured values over the sensors considering AWGN assumption. Therefore, here we focus on the consensus averaging, and adopts the framework proposed by Xiao *et al.* [5]. In this framework, consensus averaging is achieved by mutual exchange of the information among the neighbor sensors through multiple iterations and updating the data at each sensor according to a weighted sum of the received data. By appropriate choice of weights and after large enough iterations, the sensors' data will converge to the average of initial measurements of all sensors. The main advantage of such a scheme is to avoid multi-hop transmissions.

According to [5], each sensor updates its data as following:

$$x_i(t+1) = W_{ii}x_i(t) + \sum_{j \in N_i} W_{ij}x_j(t) = \mathbf{W}\mathbf{x}(t) \quad (1)$$

where N_i is the set of neighbors of node i and assuming $W_{ij} = 0$ for $j \notin N_i$. After t iterations we have:

$$\mathbf{x}(t) = \mathbf{W}^t \mathbf{x}(0) \quad (2)$$

where each sensor's initial data, $x_i(0)$, is its raw measurement, y_i. In other words, the algorithm initializes with $x_i(0) = y_i$.

In order to compute the average of $\mathbf{x}(0)$ elements, the necessary condition for mis-adjustment free convergence is:

$$\lim_{t\to\infty} \mathbf{W}^t = \frac{1}{n}\mathbf{1}\mathbf{1}^T \qquad (3)$$

So far, several weights have been proposed. Two famous and simple ones are Maximum-degree and Metropolis [5]. Metropolis weight matrix is defined as:

$$W_{ij} = \begin{cases} \frac{1}{1+\max\{|N_i|,|N_j|\}} & j \in N_i \\ 1 - \sum_{i,k\in N_i} W_{ik} & i = j \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

According to the above weights, 1 can be interpreted as a fusion scheme in which each node spatially updates its data according to the difference between its own and the neighbors' data.

Since the degree of each node depends on its relative position to other sensors, the rate of convergence will vary widely over the network. The higher the degree of a node, the more precise would be the estimation of unknown parameter in each iteration and hence the larger will be the rate of convergence. Based on the above discussion, some nodes will suffer from slow convergence and consequently large final error after a certain number of iterations due to low connectivity degree. We call the slow convergence effect due to these reasons the *forgotten node effect*. This effect is due to the fact that in a fusion model like this, the rate of convergence of the nodes depend on the amount of interactions they can have with the network in a certain amount of time. In the case of lowly connected nodes, low convergence rate stems from little number of options the node has for interaction to the network, i.e. little number of neighbors or low connectivity degree.

This paper presents a new method for updating sensor estimations in a spatio-temporal manner to improve not only the overall convergence rate, but also eliminate the forgotten node effect. In other words, our scheme improves overall convergence rate and at the same time reduces error variance among the nodes. This is achieved by updating each sensors' weights through time. We have formulated our method according to the adaptive filter theory concepts and adopted a least mean square (LMS) analysis to analytically derive convergence conditions.

The paper is organized as follows: in section II we formulate adaptive consensus averaging problem and derive its convergence conditions. Simulation results are discussed in section III. Section IV presents our future work and concluding remarks.

II. Adaptive Consensus

A. Adaptive Model

In order to resolve the effect of low connectivity degree on convergence behavior, we propose a method for temporally updating the averaging weights of each node. This can be interpreted according to the adaptive filter concept as well, where we model each node estimation or data as the output of an adaptive filter whose input is the difference between its current estimation and current estimation of its neighbors.

Consider node i with $|N_i|$ neighbors; the temporally adaptive updating equation can be formulated based on 1 as:

$$\begin{aligned} x_i(t+1) &= W_{ii}(t)x_i(t) + \sum_{j\in N_i} W_{ij}(t)x_j(t) & (5)\\ &= x_i(t) + \sum_{j\in N_i} W_{ij}(t)(x_j(t) - x_i(t)) & (6)\end{aligned}$$

assuming that $\sum_j W_{ij} = 1$. According to 6, we establish a *virtual adaptive filter* for each node that relates the difference between $x_i(t+1)$ and $x_i(t)$ by a time adaptive weighted sum of differences of the current data of node i and its neighbors' data. This filter has $|N_i|$ taps, each of which corresponds to one the neighbor of node i whose input data is $x_j(t) - x_i(t)$ for $j \in N_i$.

Now defining $\mathbf{x_i'}$ or simply $\mathbf{x}'$ as the vector of data from neighbors of node i. The desired signal, $x_{di}(t)$, is defined as:

$$x_{di}(t) = \frac{1}{|N_i|+1}\left(x_i(t) + \mathbf{1}^T\mathbf{x}'(t)\right) \qquad (7)$$

The desired signal defined here is an approximate of the actual desired signal which is the average of all the sensors' data. This is the best available estimate of the ultimate average since only neighbors' data is available to each node. et, $x_{di}(t)$ is an unbiased estimation of the actual desired signal since if $x_i(t+1)$ equals $x_{di}(t)$ and analogously to the average of its neighbors' data, the algorithm has converged.

B. Adaptive Consensus Algorithm

In the adaptive consensus, the objective is to minimize the difference between $x_i(t+1)$ and $x_{di}(t)$ which characterizes the error perceived node i at time t. Therefore, considering 6 we have:

$$\begin{aligned} e_i(t) &= x_{di}(t) - x_i(t+1) & (8)\\ &= \frac{1}{|N_i|+1}\left(x_i(t) + \mathbf{1}^T\mathbf{x}'(t)\right) - x_i(t) & \\ &\quad -\mathbf{W}_i^T(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}] & (9)\end{aligned}$$

and the cost function to minimize will be:

$$J_i = E\{e_i^2(t)\} \qquad (10)$$

By taking the gradient of J_i , we have:

$$\nabla J_i = -2E\{e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]\} \qquad (11)$$

Equation 11 can be used in updating weights based on steepest descent method:

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \mu_i E\{e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]\} \quad (12)$$

where μ_i is the step size parameter for node i. Equation 12 requires knowledge of expected value of $\mathbf{x}(t)$; however, each sensor is only aware of its own and the neighbors' data. By using the concept of stochastic gradient, we can replace the term $E\{e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]\}$ with its instantaneous value, $e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]$, similar to LMS type algorithms. Consequently:

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \mu_i e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}] \quad (13)$$

Algorithm 1 presents how 13 is realized on each node.

Algorithm Adaptive Consensus Averaging

Initialization
  $x_i(0) \leftarrow y_i$
Main Loop
While $t <$ `max_iteration` AND `status`$\neq$`converged`
  $x_i(t+1) \leftarrow x_i(t) + \mathbf{W}_i^T(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]$
  $x_{di}(t) \leftarrow \frac{1}{|N_i|+1}\left(x_i(t) + \mathbf{1}^T\mathbf{x}'(t)\right)$
  $e_i(t) \leftarrow x_{di}(t) - x_i(t+1)$
  $\mathbf{W}_i(t+1) \leftarrow \mathbf{W}_i(t) + \mu_i e_i(t)[\mathbf{x}'(t) - x_i(t)\mathbf{1}]$
end while
end

Algorithm 1. Adaptive Consensus Averaging

It must be noted that the main role of adaptive consensus scheme introduced here is improving the convergence by adapting the weights through time. Nevertheless, proper choice of the initial weights affects the convergence properties as well. For example, one can use adaptive consensus in conjunction with Metropolis initial weights to form an adaptive Metropolis scheme.

C. Proof of Convergence

In this section, we determine the range of step-size for which the convergence of 13 is guaranteed. First, we determine the range of step-size for a scenario in which only node i updates its data using proposed scheme and all other nodes use constant weights.

Theorem 1: If the step size parameter is selected so that

$$0 < \mu_i < \frac{2}{\sigma_x^2(|N_i|+1)} \quad (14)$$

holds, then a network with only node i updating adaptively according to 13, will converge in the mean. In the above equation, σ_x^2 is the variance of the input signal.

Proof: for a conventional LMS algorithm to converge in the mean, the step-size parameter should satisfy [6]:

$$0 < \mu_i < \frac{2}{\lambda_{\max}} \quad (15)$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{R} = [r_{ij}]_{|N_i|\times|N_i|}$, which is the correlation matrix of the input data defined as:

$$\mathbf{R} = E\{[\mathbf{x}'(t) - x_i(t)\mathbf{1}][\mathbf{x}'(t) - x_i(t)\mathbf{1}]^T\} \quad (16)$$

It can be shown that:

$$\mathbf{R} = \sigma_x^2(\mathbf{1}\mathbf{1}^T + \mathbf{I}) \quad (17)$$

where $\mathbf{I}$ is the unit matrix. It can be shown that the largest eigenvalue of $\mathbf{R}$ is given by:

$$\lambda = \sigma_x^2(|N_i|+1) \quad (18)$$

Hence we conclude that for the above LMS to converge in the mean, the following condition should hold:

$$0 < \mu_i < \frac{2}{\sigma_x^2(|N_i|+1)} \quad (19)$$

■

Although theorem 1 provides a range for μ_i that guarantees the convergence in the mean, as stated in [6], to achieve convergence in strict sense, the upper bound given in 19 should be restricted to $\frac{1}{|N_i|}$ of its value. Therefore, the strict sense convergence condition is:

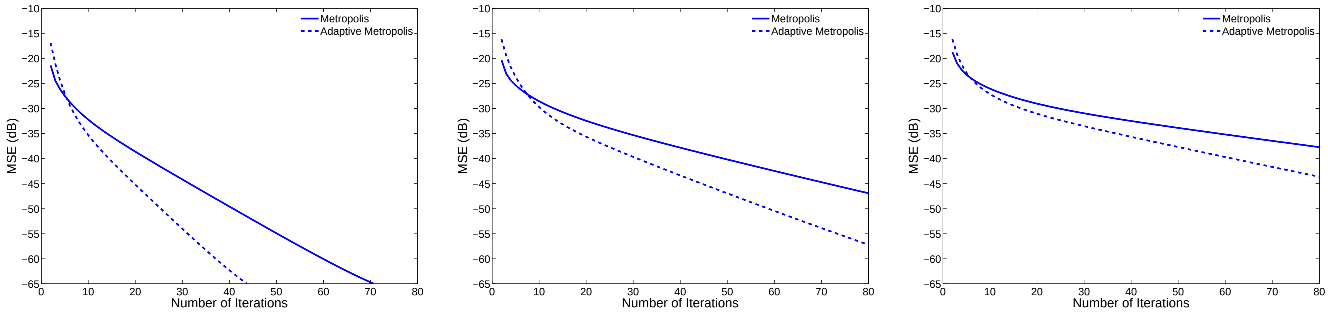
$$0 < \mu_i < \frac{2}{\sigma_x^2|N_i|(|N_i|+1)} \quad (20)$$

It should be noted that although limiting the step-size to upper bound of equation 20, will lead to convergence, as it is shown in the next section, in order to have better performance compared to the constant weights, we should set the step-size parameter to about one order of magnitude less than upper bound given in 20.

III. Simulation Results and Analysis

In this section, the performance of the adaptive consensus scheme is analyzed through simulation and compared with its static counterpart. We considered a network of 100 randomly placed nodes over $[0,1]\times[0,1]$ field. In order to study the effect of average connectivity degree, we study the results for various range of sights (RoS), D, which defines the range below which nodes are considered connected. Mean Square Error (MSE) for $D = 0.15, 0.2$ and 0.25 are analyzed here. Each sensor takes a scalar noisy measurement $y_i = \theta + \nu_i$, where θ is unknown parameter and ν_is are AWGN samples with distribution $N(0, \sigma^2)$.

Figure 1(a)-(c) show the simulation results for various D. For a specific D, each figure depicts the average mean square error behavior versus time, that is, the average error over all the nodes is depicted. Results for adaptive consensus with Metropolis initial weights, i.e. adaptive Metropolis, are presented here. To give a comparative



(a) $D = 0.25$ (b) $D = 0.2$ (c) $D = 0.15$

Fig. 1. Average MSE vs. iteration for various range of sights; $D = 0.25, 0.2, 0.15$

sense, we studied static Metropolis weights as well. It must be noted that the essence of adaptive consensus is its spatio-temporal updating method, hence, the initial weights can be freely chosen saving that they do not result in divergence of the algorithm, that is adaptive max-degree scheme can be derived similarly. However, proper choice of the initial weights, which corresponds to the weights for only spatial updating has a great impact on the convergence behavior of the system.

We study the performance of improvements of our proposed method in terms of average error and error variance over the network. Table I and II presents average error and error variance respectively over the network after 80 iterations for various D in dB for both Metropolis and adaptive Metropolis methods as well as their ratio. According to table II, adaptive Metropolis has fairly lower error variance in comparison with its static counterpart. This means that the error perceived by each node error does not differ greatly over the network and therefore, the forgotten node effects which stems from low connectivity degree will not be crucial in adaptive consensus method.

TABLE I

AVERAGE FINAL ERROR AFTER 80 ITERATIONS (dB).

D	Metropolis	Adaptive Metropolis	Ratio
0.15	-37.7193	-43.6313	5.9119
0.2	-46.9184	-57.2002	10.2818
0.25	-68.1872	-71.0293	2.8421

TABLE II

VARIANCE OF FINAL SQUARE ERROR AFTER 80 ITERATIONS (dB).

D	Metropolis	Adaptive Metropolis	Ratio
0.15	-76.4316	-89.2097	12.7781
0.2	-97.4774	-116.0034	18.5260
0.25	-142.1162	-175.1353	33.0191

IV. CONCLUSION AND FUTURE WORK

This paper introduced the adaptive consensus scheme, an adaptive method to improve convergence behavior of distributed consensus averaging methods. The proposed scheme not only improves the convergence rate, but also the variance of the error over the network. This result is achieved by introducing time adaptive weights which are derived through an LMS analysis. We are considering a more detail analysis of the convergence behavior of the adaptive consensus scheme as well as the effect of mobility on the convergence rate for our future work.

ACKNOWLEDGMENTS

This work is supported by Iran Telecommunication Research Center (ITRC) and Sharif Advanced Information and Communication Technology Center (AICTC).

REFERENCES

[1] J. N. Tsitsiklis. “Decentralized Detection,” *in Advances in Signal Processing*, Vol. 2, JAI Press, 1993, pp. 297-344.

[2] R. Olfati-Saber. “Distributed Kalman Filter with Embedded Consensus Filters,” *Proceedings of the joint CDC-ECC ’05 Conference*, Dec. 2005.

[3] Mortada Mehyar, Demetri Spanos, John Pongsajapan, Steven Low and Richard Murray. “Distributed Averaging on Asynchronous Communication Networks,” *Appears in Proceedings of IEEE Conference on Decision and Control*, 2005, Seville, Spain.

[4] R. Olfati-Saber and R.M. Murray. “Consensus problems in networks of agents with switching topology and time-delays,” *Automatic Control, IEEE Transactions on* , vol.49, no.9pp. 1520- 1533, Sept. 2004.

[5] L. Xiao, S. Boyd and S. Lall. “A scheme for robust distributed sensor fusion based on average consensus,” *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 63-70, 2005.

[6] Simon Hakin. “Adaptive Filter Theory,” Prentice Hall PTR, 2002.