DISTRIBUTED LMS ESTIMATION OF SCALED AND DELAYED IMPULSE RESPONSES

Kevin T. Wagner

Naval Research Laboratory
Radar Division
Washington, DC 20375, USA

Miloš I. Doroslovački

The George Washington University
Department of Electrical and Computer Engineering
Washington, DC 20052, USA

ABSTRACT

The ability to perform distributed estimation across a neighborhood of nodes when the impulse responses present at each node are related by an unknown time delay and scaling is presented in this work. The time delay estimation is performed by correlating estimates of the unknown impulse response from neighboring nodes. The estimated time delay corresponds to the shift associated with the peak of this correlation. Next combination of neighboring nodes is performed by reweighting node estimates. This reweighting factor is calculated by forming the ratio of the norm of estimated impulse response in a reference node with the norm of the estimated impulse response in a neighboring node. Simulation results demonstrating improved convergence performance for a cooperative network when estimating the time delay and amplitude weights relative to the performance of a non-cooperative network are depicted.

Index Terms— Adaptive filtering, convergence, distributed algorithms, least mean square algorithms.

1. INTRODUCTION

Distributed estimation and adaptation over networks has been an area of active research in recent years [1] [2] [3] [4]. These research efforts have focused on learning a single or cluster [5] [6] of unknown impulse response by cooperation across a network of nodes. The most often assumed condition that the unknown impulse response is the same at each node restricts the class of problems that can be solved using distributed estimation. In this work, the restriction of an identical impulse response at each node is relaxed. Instead, the impulse response being estimated at each node is assumed to have an unknown time-delay and amplitude weighting relative to other nodes in the network. Schemes are presented to estimate these unknown time delays and amplitude weightings that enable cooperation across the network.

In order to incorporate estimates from other nodes, each reference node needs to initially align information sent from neightbor nodes. Several alignment strategies are examined in this work. In particular alignment using correlation between the reference and neighbor weight deviations are examined. After alignment is performed, a methodology estimating the unknown amplitudes is presented. Once the amplitudes are estimated the nodes are reweighted prior to combination. This reweighting and alignment enables the distributed network estimation to achieve improved steady-state convergence relative to non-cooperative networks.

A summary of each section follows next. In section 2, the distributed network estimation framework is presented. Section 3 briefly reviews distributed minimum mean square weight deviation algorithms and the associated notation. Next, sections 4 and 5 describe time delay estimation and amplitude renormalization that occurs in each node, respectively. Finally, simulation results are presented in section 6.

2. DISTRIBUTED ESTIMATION PROBLEM FRAMEWORK

2.1. Network Model

Assume a network with N nodes, where each node has an agent. The neighborhood of any particular agent, k, is denoted by $\mathcal{N}_k$ and consists of all nodes that are connected to node k. Node l is connected to node k if node k receives information from node l. Node k is assumed to be in $\mathcal{N}_k$. Two nodes are said to be two-way connected if they can share information directly with each other. Let $\{a_{kl}(i), a_{lk}(i)\}$ represent the set of combination coefficients between agents k and l at time i. The scalar $a_{lk}(i)$ is used by agent k to scale the estimate it receives from agent l, while $a_{kl}(i)$ is used by agent l to scale the estimate received from agent k. If $a_{lk}(i) \neq 0$ node l is said to be connected to node k. Define

$$\mathbf{a}_k(i) = [a_{1k}(i), a_{2k}(i), \ldots, a_{Nk}(i)]^T$$

where $(\cdot)^T$ represents the transpose operation. If the l^{th} node is not connected to the k^{th} node at time i then $a_{lk}(i) = 0$. It is assumed that $\mathbf{a}_k^T(i)\mathbf{1} = 1$, where $\mathbf{1}$ is the vector of all ones.

2.2. Adaptation Framework

Consider the situation in which the N agents are attempting to estimate N unknown impulse response vectors. Each impulse response has length L. Let the impulse response at the

k^{th} node be represented by $\mathbf{w}_{o,k}$. Each unknown impulse response is a delayed and scaled version of the shape defining vector $\mathbf{w}_o$. For instance, let

$$\mathbf{w}_o = [w_o(1), w_o(2), \ldots, w_o(M)]^T,$$

where M represents the length of the unknown shape defining vector. Then

$$\mathbf{w}_{o,k} = c_k[\mathbf{0}_{\delta_k}^T, \mathbf{w}_o^T, \mathbf{0}_{L-M-\delta_k}^T]^T,$$

where $\mathbf{0}_p$ is a vector of zeros with length p. The delay and scaling at the k^{th} node are unknown apriori and denoted as $\delta_k \in \{0, 1, \ldots, M\}$ and $c_k \in (0, 1]$, respectively. Let $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_N]^T$ and $\mathbf{c} = [c_1, \ldots, c_N]^T$ represent the unknown delays and amplitudes at each node in the network. Each node attempts to estimate the impulse response using the least mean square (LMS) algorithm [7]. Each agent k interrogates the unknown impulse response at time i by sending an input sequence $\mathbf{x}_k(i)$ where $\mathbf{x}_k(i) = [x_k(i), x_k(i-1), \ldots, x_k(i-L+1)]^T$. Let the received response be represented by $d_k(i) = \mathbf{x}_k^T(i)\mathbf{w}_{o,k} + v_k(i)$ where $v_k(i)$ is the measurement noise. The impulse response of the system is estimated with the adaptive filter coefficient vector, $\mathbf{w}_k(i)$, which has length L also. The output of the adaptive filter is given by $y_k(i) = \mathbf{x}_k^T(i)\mathbf{w}_k(i-1)$. The error signal $e_k(i)$ is equal to difference of the measured output, $d_k(i)$ and the output of the adaptive filter $y_k(i)$. Define the weight deviation at time i for the k^{th} node as

$$\mathbf{z}_k(i) = \mathbf{w}_{o,k} - \mathbf{w}_k(i). \tag{1}$$

Each agent employs step size μ_k in its LMS algorithms. The step size μ_k is sufficiently small such that the weight deviation can be considered independent of the input signal for all times and nodes.

2.3. Signal Assumptions

Real-valued signals are assumed in this manuscript. The input signal $\mathbf{x}_k(i)$ is a zero-mean stationary random process with covariance matrix given by $E\{\mathbf{x}_k(i)\mathbf{x}_k^T(i)\} = \mathbf{R}_k$. The noise process is white with variance $E\{v_k^2(i)\} = \sigma_{v,k}^2$, assumed to be independent of the input signal at all nodes, and independent of all measurement noises at nodes $l \neq k$. The input signals are independent of each other at different nodes implying $E\{\mathbf{x}_k(i)\mathbf{x}_l^T(i)\} = \mathbf{0}$ for $l \neq k$.

3. MINIMUM MEAN SQUARE WEIGHT DEVIATION COMBINATION COEFFICIENT ALGORITHM

The implementation of the quadratic programming (QPOCC) and unconstrained optimal combination coefficient (UOCC) algorithms were initially presented in [8] [9]. For the sake of brevity, relevant variables are defined next.

The adapt-then-combine (ATC) diffusion strategy is given by

$$\begin{aligned}\boldsymbol{\psi}_k(i) &= \mathbf{w}_k(i-1) + \mu\mathbf{x}_k(i)[d_k(i) - \mathbf{x}_k^T(i)\mathbf{w}_k(i-1)] \\ \mathbf{w}_k(i) &= \sum_{l\in\mathcal{N}_k} a_{lk}(i)\boldsymbol{\psi}_l(i).\end{aligned} \tag{2}$$

Next the adapted weight deviation (AWD) $\mathbf{y}_k$ is defined as

$$\mathbf{y}_k(i) = \mathbf{w}_{o,k} - \boldsymbol{\psi}_k(i).$$

In [8], the block versions of $\boldsymbol{\psi}_k(i)$ and estimates $\mathbf{y}_k(i)$ denoted by $\boldsymbol{\psi}_k^B(i)$ and $\hat{\mathbf{y}}_k(i)$ are generated every P time steps and exchanged between neighbors at time $i = mP+P-1$ for each node k. Otherwise, when $i \neq mP + P - 1$ each node adapts and stores its local estimate of the unknown impulse response for usage during the block update.

4. IMPULSE RESPONSE DELAY ESTIMATION

In order to exploit information from neighboring nodes, each node must initially align its estimates of the adapted weight devitation, $\hat{\mathbf{y}}_k(i)$, and block updated weights, $\boldsymbol{\psi}_k^B(i)$, with its neighbors' estimates prior to combining these estimates. $\hat{\mathbf{y}}_k(i)$ are needed for calculation of minimum mean square weight deviation combination coefficients, $\mathbf{a}_k(i)$, and $\boldsymbol{\psi}_k^B(i)$ are the objects that are combined. The correlation between the reference node's estimate and its neighbors' estimates is used to perform this alignment. Care must be taken to ensure that the alignment does not degrade the ability of the adaptive filter to track time varying changes in the impulse responses. Next a strategy that builds upon the correlation metric and enables tracking of time varying impulse responses is presented.

Initially, the correlation between the reference node's estimate and its neighbors' estimates are calculated. For arbitrary vectors $\mathbf{p}_k$ and $\mathbf{p}_j$ let

$$r_{\mathbf{p}_k,\mathbf{p}_j}(l) = \sum_{n=\max\{1,1-l\}}^{\min\{L,L-l\}} [\mathbf{p}_k]_n[\mathbf{p}_j]_{n+l} \tag{3}$$

where $l \in \{-L+1, \ldots, L-1\}$ and $[\mathbf{p}_k]_n$ is the n^{th} entry of vector $\mathbf{p}_k$. Define the correlation vector

$$\boldsymbol{r}_{\mathbf{p}_k,\mathbf{p}_j} = [r_{\mathbf{p}_k,\mathbf{p}_j}(-L+1), \ldots, r_{\mathbf{p}_k,\mathbf{p}_j}(L-1)]. \tag{4}$$

Assume that $\boldsymbol{r}_{\hat{\mathbf{y}}_k,\hat{\mathbf{y}}_j}$ and $\boldsymbol{r}_{\boldsymbol{\psi}_k^B,\boldsymbol{\psi}_j^B}$ are calculated using (3) and (4). Next the maximum value of $\boldsymbol{r}_{\hat{\mathbf{y}}_k,\hat{\mathbf{y}}_j}$ is compared to the maximum value of $\boldsymbol{r}_{\boldsymbol{\psi}_k^B,\boldsymbol{\psi}_j^B}$. If $\max\{\boldsymbol{r}_{\hat{\mathbf{y}}_k,\hat{\mathbf{y}}_j}\} > \max\{\boldsymbol{r}_{\boldsymbol{\psi}_k^B,\boldsymbol{\psi}_j^B}\}$ then alignment of node j relative to node k is performed using l, denoted $l_{k,j}^*$, that corresponds to the value $\max\{\boldsymbol{r}_{\hat{\mathbf{y}}_k,\hat{\mathbf{y}}_j}\}$. Otherwise alignment is performed using l corresponding to the value $\max\{\boldsymbol{r}_{\boldsymbol{\psi}_k^B,\boldsymbol{\psi}_j^B}\}$. This logic allows the adaptive filter to track changes in the unknown impulse response.

During the initial stages of convergence the estimates $\boldsymbol{\psi}_k^B$ are close to zero for all nodes as they are initialized to be zero vectors. Therefore, attempting to perform alignment using these estimates results in erroneous alignment and poor convergence performance. To the contrary, the estimates $\hat{\mathbf{y}}_k$ have values that are nearer the unknown impulse response during the initial stages of convergence. Therefore it is preferable to use the estimate $\hat{\mathbf{y}}_k$ in this stage.

Once the adaptive filter starts to converge, the estimates $\hat{\mathbf{y}}_k$ approach zero for all nodes and become a poor choice to perform alignment. At this point the estimates $\boldsymbol{\psi}_k^B$ are a better choice to perform alignment. If the impulse response starts to change, $\hat{\mathbf{y}}_k(i)$ can again become more informative for alignment.

A second strategy to perform alignment attempts to compare the sparsity of the estimates $\boldsymbol{r}_{\hat{\mathbf{y}}_k,\hat{\mathbf{y}}_j}$ and $\boldsymbol{r}_{\boldsymbol{\psi}_k^B,\boldsymbol{\psi}_j^B}$. This methodolgy resulted in poor convergence results and therefore is omitted.

5. AMPLITUDE NORMALIZATION

Prior to calculating the combination coefficients $a_{lk}(i)$ and performing the combination step in (2) each node's estimates of $\boldsymbol{\psi}_l^B(i)$ and $\hat{\mathbf{y}}_l(i)$ need to be normalized relative to the k^{th} node for all $l \in \mathcal{N}_k$. Note that the adapted weights can be approximated in steady state by

$$\boldsymbol{\psi}_l^B(i) \approx \mathbf{w}_{o,l} = c_l[\mathbf{0}_{\delta_k}^T, \mathbf{w}_o^T, \mathbf{0}_{L-M-\delta_k}^T]^T. \tag{5}$$

Therefore it is reasonable to reweight the estimates $\boldsymbol{\psi}_l^B(i)$ and $\hat{\mathbf{y}}_l(i)$ using the ratio

$$\gamma_{kl} = \frac{||\boldsymbol{\psi}_k^B(i)||_2}{||\boldsymbol{\psi}_l^B(i)||_2} \approx \frac{c_k}{c_l} \tag{6}$$

where $||\cdot||_2$ denotes the L^2 norm.

In practice the estimates $\boldsymbol{\psi}_l^B(i)$ can be close to zero in the initial stages of adaptation which may cause very poor reweighting coefficients γ_{kl} which may lead to algorithm instability. Two strategies are proposed to overcome this potential issue. The first strategy consists of increasing the block size P to delay the start of combining until weights are no longer close to zero. The second strategy introduces the following logic:

$$\gamma_{kl} = \begin{cases} \frac{||\hat{\mathbf{y}}_k(i)||_2}{||\hat{\mathbf{y}}_l(i)||_2} \text{ if } ||\hat{\mathbf{y}}_k(i)||_2 > ||\boldsymbol{\psi}_k^B(i)||_2 \\ \frac{||\boldsymbol{\psi}_k^B(i)||_2}{||\boldsymbol{\psi}_l^B(i)||_2} \text{ if } ||\hat{\mathbf{y}}_k(i)||_2 \leq ||\boldsymbol{\psi}_k^B(i)||_2. \end{cases}$$

Once the amplitude and time delays are estimated the AWD and adapted weights are modified to calculate combination coefficients and perform combination in node k as follows:

$$\begin{aligned} \mathbf{y}_{k,l}^*(i) &= \gamma_{kl}\, \mathbf{S}^{l_{k,l}^*}\, \hat{\mathbf{y}}_l(i) \\ \boldsymbol{\psi}_{k,l}^*(i) &= \gamma_{kl}\, \mathbf{S}^{l_{k,l}^*}\, \boldsymbol{\psi}_l^B(i) \end{aligned} \tag{7}$$

where $l_{k,l}^*$ is the estimated relative time delay and $\mathbf{S}$ is the shift matrix given by

$$\mathbf{S} = \begin{cases} \mathbf{S}_+ = \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{L-1\times L-1} & \mathbf{0} \end{bmatrix} \text{ if } l_{k,l}^* > 0 \\ \mathbf{I}_{L\times L} \text{ if } l_{k,l}^* = 0 \\ \mathbf{S}_+^T \text{ if } l_{k,l}^* < 0. \end{cases}$$

6. SIMULATION RESULTS

In this section the arithmetic mean of the MSWD across all nodes in the network was calculated versus iteration for different diffusion strategies. The combination-coefficient methods employed consisted of the non-cooperative network (NCN), unconstrained optimal combination coefficients solution (U-OCC), quadratic programming optimal combination coefficients solution (QPOCC). These methods were implemented within the ATC network diffusion strategy.

The following parameters were used in all simulations. The network had seven nodes and the topology is depicted in Figure 1. Blue lines indicate that both nodes are connected to each other. Red lines indicate that a connection exists only in one direction. The arrow heads on the red lines indicate in which direction information is being sent. The circle around nodes indicate that the nodes are connected to themselves. Each node is labeled and the number in parenthesis indicates the signal to noise ratio in decibels at the measurement point at that node. The impulse response had length fifty. The impulse response is depicted in Figure 2. The initial delay at each node, measurement noise variance and input signal variance were given by

$$\begin{aligned} \boldsymbol{\delta} &= [6, 6, 2, 9, 1, 6, 3]^T, \\ \boldsymbol{\sigma}_v^2 &= [0.001, 0.009, 0.004, 0.002, 0.018, 0.010, 0.023]^T, \\ \boldsymbol{\sigma}_x^2 &= [2.775, 2.085, 0.768, 0.833, 1.712, 1.670, 0.108]^T, \end{aligned}$$

respectively. When used the scaling on each impulse response was given by

$$\mathbf{c} = [0.422, 0.410, 0.648, 0.051, 0.826, 0.667, 0.313]^T.$$

The step size was given by

$$\mu_k = \frac{2}{(100L\sigma_{x,k}^2)}.$$

The block size $P = 150$ was used in Figures 3. In Figure 4 the value $P = 350$ was employed. The steps observed in the convergence curves are due to updates occuring in blocks. We considered white real-valued input signals. A total of 100 Monte Carlo trials were used to generate all figures.

At time $k = 25000$ the impulse response is changed at each node by multipling the initial impulse responses by -1 and delaying each impulse response by an additional 10 samples.

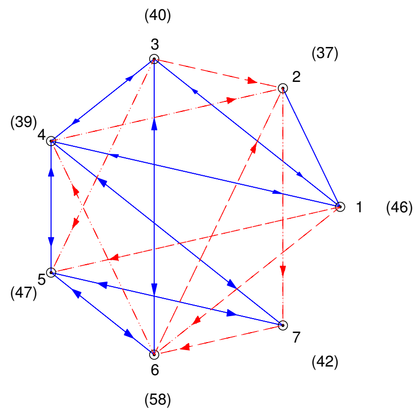

Fig. 1. Network Graph

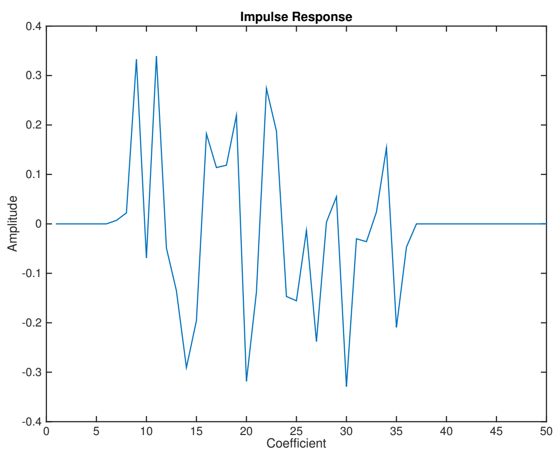

Fig. 2. Impulse Response

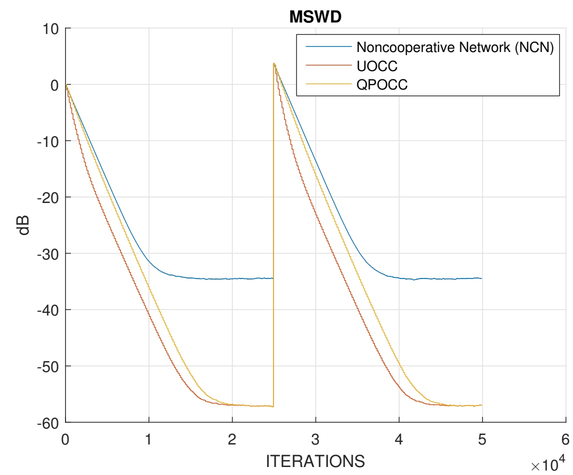

Fig. 3. ATC Distributed Estimation Strategy Learning Curve Comparison Using Maximum of Correlation for Switching

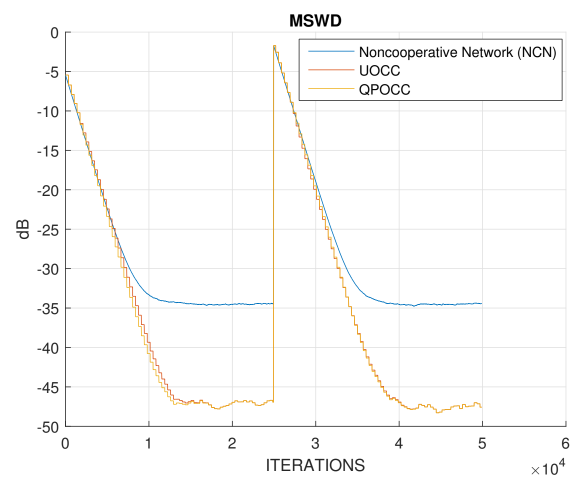

Fig. 4. ATC Distributed Estimation Strategy Learning Curve Comparison Using Maximum of Correlation for Switching and Amplitude Renormalization

In Figure 3 the learning curves when using the maximum of the correlation to estimate the unknown time-delay are depicted. The unknown impulses responses each had $c_l = 1$ for all $l = 1, \ldots, N$. The algorithm using the maximum of the correlation to perform alignment converges and can handle time-varying impulse responses.

In Figure 4 the learning curves when using the maximum of the correlation to calculate the unknown time-delay of each impulse response and amplitude renormalization to remove the effects of the unknown amplitude weights $\mathbf{c}$ are depicted. The transient regime convergence is on par with the NCN algorithm, but the steady-state convergence is greatly improved.

7. CONCLUSION

A novel strategy has developed and shown to handle the case of distributed learning across a network when the impulse response at each node has same shape with an unknown time delay and unknown scaling. In order to take advantage of information across the network, time alignment needs to occur across the separate nodes. A strategy for implementing this time alignment was developed and demonstrated. Prior to combination each estimated impulse response of neighboring nodes should be normalized to the scale of the estimated impulse response of the node performing combinations. The renormalization is performed by forming the ratio of the norm of a reference nodes estimated weights relative to the norm of neighbor nodes estimated weights. The renormalization process enables improved steady-state convergence performance.

8. REFERENCES

[1] A.H. Sayed, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, and Z.J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 155–171, 2013.

[2] F.S. Cattivelli and A.H. Sayed, "Diffusion lms strategies for distributed estimation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[3] I.D. Schizas, G. Mateos, and G.B. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[4] Tinati Mohammad Rastegarnia Amir and Khalili Azam, "Distributed estimation using an improved incremental distributed lms algorithm," *International Science Index*, vol. 3, no. 11, pp. 578–582, 2009.

[5] Jie Chen, C. Richard, and A.H. Sayed, "Multitask diffusion adaptation over networks," *Signal Processing, IEEE Transactions on*, vol. 62, no. 16, pp. 4129–4144, Aug 2014.

[6] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed diffusion-based lms for node-specific adaptive parameter estimation," *Signal Processing, IEEE Transactions on*, vol. 63, no. 13, pp. 3448–3460, July 2015.

[7] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey, fourth edition, 2002.

[8] K.T. Wagner and M.I Doroslovacki, "Combination coefficients for fastest convergence of distributed lms estimation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 7218–7222.

[9] K.T. Wagner and M.I Doroslovacki, "Improving convergence of distributed lms estimation by enabling propagation of good estimates through bad nodes," in *Asilomar Conference on Signals, Systems and Computers (ACSSC)*, Nov in press 2015.