Optimal Combination Rules for Adaptation and Learning over Networks

Sheng-Yuan Tu and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles, CA 90095
E-mail: {shinetu, sayed}@ee.ucla.edu

***Abstract*—Adaptive networks, consisting of a collection of nodes with learning abilities, are well-suited to solve distributed inference problems and to model various types of self-organized behavior observed in nature. One important issue in designing adaptive networks is how to fuse the information collected from the neighbors, especially since the mean-square performance of the network depends on the choice of combination weights. We consider the problem of optimal selection of the combination weights and motivate one combination rule, along with an adaptive implementation. The rule is related to the inverse of the noise variances and is shown to be effective in simulations.**

***Index Terms*—Adaptive networks, diffusion adaptation, relative-variance combination rule, self-organization, distributed processing.**

I. Introduction

Adaptive networks consist of a collection of spatially distributed nodes that are linked together through a connection topology. The nodes cooperate with each other through local interactions to solve distributed inference problems in real-time. The diffusion of information across the network results in improved adaptation and learning relative to non-cooperative networks. Adaptive networks are well-suited to perform decentralized information processing [1], [2] and to model self-organized behavior encountered in nature, such as animal flocking behavior [3]–[5].

Each node in an adaptive network relies on the fusion of information collected from its local neighbors. Several combination rules have been proposed in the literature, especially in the context of consensus-based iterations [6]–[9], such as the maximum-degree rule and the Metropolis rule. However, these schemes focus on convergence behavior and ignore the variation in noise (and signal-to-noise ratio) profile across the nodes, which can result in performance degradation [10]. Therefore, designing combination rules that take into account the variation in noise profile over the network is an important task.

In this paper, we incorporate the noise profile into the design of the combination weights. Some earlier work in this regard appeared in [2], which relied on the formulation and solution of an optimization problem. However, the optimization problem was nonlinear and non-convex, and its solution was pursued numerically. In this paper, we introduce an approximation and formulate a convex optimization problem that can be solved in closed-form and lead to good performance. The solution, nevertheless, requires knowledge of the second-order statistics of the noise. We subsequently introduce an adaptive implementation for adjusting the combination weights by relying on instantaneous data approximations. In this way, besides the standard adaptation layer to solve the desired distributed estimation, each node also runs a second adaptation layer to adjust its combination weights in real-time.

II. Diffusion Adaptation

A. Algorithm Description

Consider a collection of N nodes distributed over a spatial domain. Two nodes are said to be neighbors if they can share information. The set of neighbors of node k, including k itself, is called the neighborhood of k and is denoted by $\mathcal{N}_k$. At every time instant i, every node k has access to a scalar measurement $d_k(i)$ and a row regression vector $u_{k,i}$ of size M, both arising from realizations of zero-mean random processes, $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$; note that we use boldface letters to refer to random quantities and normal font to refer to their realizations. The available measurements are assumed to be related to some unknown column vector w° of size M as follows:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^\circ + \boldsymbol{v}_k(i) \tag{1}$$

where $\boldsymbol{v}_k(i)$ denotes noise and is assumed to be a zero-mean white random process with power $\sigma_{v,k}^2$ and is independent of all other variables.

The nodes seek to estimate a parameter w° that minimizes the following cost function:

$$J^{glob}(w) = \sum_{k=1}^{N} E|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2 \tag{2}$$

where E denotes the expectation operator. The objective of the network is to estimate w° in a distributed manner and in real-time. Several diffusion adaptation schemes for solving (2) in this manner were developed in [1], [2]. One such scheme is the so-called Adapt-then-Combine (ATC) diffusion algorithm

This work was supported in part by NSF grants CCF-1011918 and CCF-0942936.

[2]. It operates as follows. We select an $N \times N$ matrix A with nonnegative entries $\{a_{l,k}\}$ satisfying:

$$\mathbf{1}^T A = \mathbf{1}^T \text{ and } a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \tag{3}$$

where $\mathbf{1}$ is the vector with all its entries equal to one. The entry $a_{l,k}$ refers to weight used for the data exchanged over the link connecting node l (source) to node k (destination). The ATC algorithm consists of two steps. The first step (4) involves local adaptation, where node k use its own data $\{d_k(i), u_{k,i}\}$ to update the weight estimate at node k from $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$. The second step (5) is a combination step where the intermediate estimates $\{\psi_{l,i}\}$ from the neighborhood are combined through the coefficients $\{a_{k,l}\}$ to obtain the updated weight estimate $w_{k,i}$. The algorithm is described as follows:

$$\psi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^*[d_k(i) - u_{k,i} w_{k,i-1}] \tag{4}$$

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} \psi_{l,i} \tag{5}$$

where μ_k is the positive step-size used by node k. Naturally, the choice of the weighting matrix A affects the performance of the network. Before formulating an optimization problem for designing A, we note that the matrix is not generally required to be symmetric (i.e., the weight that node k assigns to data arriving from node l does not need to be equal to the weight that node l assigns to data arriving from node k). It was noted earlier in [2] that non-symmetric choices for A lead to better network mean-square performance in estimating w°.

B. Mean-Square Performance

The mean-square performance of the ATC algorithm was studied in detail in [2] by applying the energy conservation approach of [11]. We summarize the results below in preparation for our proposal of a combination rule. Let the error vector for node k be denoted by:

$$\tilde{\boldsymbol{w}}_{k,i} = w^\circ - \boldsymbol{w}_{k,i} \tag{6}$$

The network MSD is defined as the following average steady-state measure:

$$\text{MSD} \triangleq \lim_{i \to \infty} \frac{1}{N} \sum_{k=1}^{N} E\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \tag{7}$$

We collect all weight error vectors and step-sizes parameters across the network into global vectors and matrices:

$$\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\} \tag{8}$$

$$\mathcal{M} = \text{diag}\{\mu_1 I_M, \ldots, \mu_N I_M\} \tag{9}$$

where the notation col$\{\cdot\}$ denotes the vector that is obtained by stacking its arguments on top of each other, and the notation diag$\{\cdot\}$ denotes a diagonal matrix formed from its arguments. We also define the extended weighting matrix:

$$\mathcal{A} = A \otimes I_M \tag{10}$$

where the symbol $\otimes$ denotes the Kronecker product of two matrices. Then, some algebra shows that the global error vector (8) evolves according to the relation:

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I - \mathcal{M}\boldsymbol{R}_i)\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T \mathcal{M} \boldsymbol{g}_i \tag{11}$$

where the identity matrix in (11) has dimensions $NM \times NM$ and

$$\boldsymbol{R}_i = \text{diag}\left\{\boldsymbol{u}_{1,i}^* \boldsymbol{u}_{1,i}, \ldots, \boldsymbol{u}_{N,i}^* \boldsymbol{u}_{N,i}\right\} \tag{12}$$

$$\boldsymbol{g}_i = \text{col}\{\boldsymbol{u}_{1,i}^* \boldsymbol{v}_1(i), \ldots, \boldsymbol{u}_{N,i}^* \boldsymbol{v}_N(i)\} \tag{13}$$

Under the assumption that all regressors $\{\boldsymbol{u}_{k,i}\}$ are spatially and temporally independent and that the step-sizes are sufficiently small, the MSD of the network can be shown to be [2]

$$\boxed{\begin{aligned} \text{MSD} &\approx \frac{1}{N}\text{vec}(Y^T)^T (I - F)^{-1} \text{vec}(I_{NM}) \\ &= \frac{1}{N} \sum_{j=0}^{\infty} \text{Tr}\left[X^j Y (X^*)^j\right] \end{aligned}} \tag{14}$$

where

$$F \approx X^T \otimes X^* \tag{15}$$

$$X = \mathcal{A}^T(I - \mathcal{M}R) \tag{16}$$

$$Y = \mathcal{A}^T \mathcal{M} G \mathcal{M} \mathcal{A} \tag{17}$$

$$R = E\boldsymbol{R}_i = \text{diag}\left\{R_{u,1}, \ldots, R_{u,N}\right\} \tag{18}$$

$$G = E\boldsymbol{g}_i \boldsymbol{g}_i^* = \text{diag}\{\sigma_{v,1}^2 R_{u,N}, \ldots, \sigma_{v,N}^2 R_{u,N}\} \tag{19}$$

with $R_{u,k} = E\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}$. For the second equality in (14), we used the following equalities for arbitrary matrices $\{U, W, \Sigma\}$:

$$\text{vec}(U\Sigma W) = (W^T \otimes U)\text{vec}(\Sigma) \tag{20}$$

$$\text{Tr}(\Sigma W) = \text{vec}(W^T)^T \text{vec}(\Sigma) \tag{21}$$

Note from expression (14) that the noise level at any node influences the network performance via the combination matrix A. In the next section, we optimize over A.

III. Combination Weights

In [2], the selection of the optimal combination weights was formulated as the following optimization problem:

$$\boxed{\begin{aligned} &\min_{A} \quad \text{MSD} \\ &\text{subject to} \\ &\qquad \mathbf{1}^T A = \mathbf{1}^T, \quad a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \end{aligned}} \tag{22}$$

However, expression (14) is not convex in A, and the optimal solution to (22) was sought numerically in [2]. We follow a different approximate approach that leads to a closed-form solution and performs equally well.

Although not necessary, in this article we illustrate the procedure by considering the case when the regressors $\{\boldsymbol{u}_{k,i}\}$ have the same covariance matrix, say, $R_{u,k} = R_u$ for all k.

We also assume that all nodes use the same step-size (i.e., $\mu_k = \mu$ for all k). Then, the MSD expression (14) reduces to:

$$\text{MSD} \approx \frac{\mu^2}{N}\sum_{j=0}^{\infty}\text{Tr}[R_u(I-\mu R_u)^{2j}]\text{Tr}[(A^T)^{j+1}VA^{j+1}] \tag{23}$$

where we introduce the matrix:

$$V = \text{diag}\{\sigma^2_{v,1}, \ldots, \sigma^2_{v,N}\} \tag{24}$$

A. Approximate Optimal Solution

Even though the MSD is simplified to (23), the MSD is still not convex in A. To proceed, we observe that the factor $\text{Tr}[R_u(I-\mu R_u)^{2j}]$ in (23) decays exponentially fast with j. We therefore choose to focus on the first term of the summation (corresponding to $j = 0$) and ignore the other terms. The simulations further ahead indicate that this approximation performs well. We therefore replace (22) with the simpler optimization problem:

$$\boxed{\begin{array}{l} \displaystyle\min_{A} \quad \text{Tr}(A^TVA) \\ \text{subject to} \\ \qquad \mathbf{1}^TA = \mathbf{1}^T, \quad a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \end{array}} \tag{25}$$

Note that the kth diagonal entry of A^TVA is given by

$$[A^TVA]_{k,k} = \sum_{l=1}^{N}\sigma^2_{v,l}a^2_{l,k} \tag{26}$$

and the value of this diagonal entry is only affected by the combination weights used by node k, i.e., the $\{a_{l,k}\}$. Therefore, the optimization problem (25) can be decoupled into N separate optimization problems of the form:

$$\boxed{\begin{array}{l} \displaystyle\min_{\{a_{l,k}\}_{l=1}^{N}} \quad \sum_{l=1}^{N}\sigma^2_{v,l}a^2_{l,k},\ k = 1, \ldots, N \\ \text{subject to} \\ \qquad \displaystyle\sum_{l=1}^{N}a_{l,k} = 1, \quad a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \end{array}} \tag{27}$$

and the solution is given by:

$$a_{l,k} = \begin{cases} \frac{\sigma^{-2}_{v,l}}{\sum_{j\in\mathcal{N}_k}\sigma^{-2}_{v,j}}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

We refer to this combination rule as the *relative-variance combination rule*. In this way, node k combines the intermediate estimates $\{\psi_{l,i}\}$ from its neighbors in (5) in proportion to the inverses of the noise variances. The result is physically meaningful. Nodes with smaller noise variance will be given larger weights. In comparison, the following relative-degree-variance rule was used in [2]:

$$a_{l,k} = \begin{cases} \frac{n_l\sigma^{-2}_{v,l}}{\sum_{j\in\mathcal{N}_k}n_j\sigma^{-2}_{v,j}}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

where n_l denotes the degree of node l, i.e., the number of its neighbors including itself.[1]

We remark that had we allowed for different $R_{u,k}$ across the nodes, the earlier derivation would lead us instead to an expression similar to (28) where the terms $\sigma^{-2}_{v,m}$ in the numerator and denominator would appear replaced by the terms $[\text{Tr}(R_{u,m})\sigma^2_{v,m}]^{-1}$. We leave the details for future work.

B. Adaptive Implementation

To evaluate the combination weights (28), we still need to have information about the noise variances across the network. These quantities are usually not available beforehand, or they may even vary with time. Therefore, an adaptive implementation is desirable, where the individual nodes learn their combination weights (28) in real-time using instantaneous data. One adaptive solution for selecting the combination weights was proposed earlier in [10]; albeit using a different design criterion. We compare the approach of [10] with our proposed solution further ahead in the simulations.

Referring to the ATC recursions (4)-(5), as the algorithm approaches steady-state, and for sufficiently small step-sizes, the estimate $w_{k,i-1}$ approaches w°. Therefore, using model (1), we can write:

$$\boldsymbol{\psi}_{k,i} \approx w^\circ + \mu\boldsymbol{u}^*_{k,i}\boldsymbol{v}_k(i) \tag{30}$$

It follows that

$$E\|\boldsymbol{\psi}_{k,i} - w^\circ\|^2 \approx \mu^2\sigma^2_{v,k}\text{Tr}(R_u) \tag{31}$$

That is, the expression on the left is a scaled multiple of the noise variance $\sigma^2_{v,k}$. Using the instantaneous approximation

$$E\|\boldsymbol{\psi}_{k,i} - w^\circ\|^2 \approx \|\psi_{k,i} - w_{k,i-1}\|^2 \tag{32}$$

we can motivate an algorithm for estimating noise variances in real-time. Let $\sigma^2_{l,k}(i)$ denote a (scaled) estimate of the noise variance $\sigma^2_{v,l}$ at node k at time i. Then we estimate it as follows:

$$\sigma^2_{l,k}(i) = (1-\nu_k)\sigma^2_{l,k}(i-1) + \nu_k\cdot\|\psi_{l,i} - w_{k,i-1}\|^2 \tag{33}$$

where ν_k is a positive step-size (smaller than one). We see that under expectation, expression (33) becomes

$$\begin{aligned} E\boldsymbol{\sigma}^2_{l,k}(i) &= (1-\nu_k)E\boldsymbol{\sigma}^2_{l,k}(i-1) + \nu_k\cdot E\|\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{k,i-1}\|^2 \\ &\approx (1-\nu_k)E\boldsymbol{\sigma}^2_{l,k}(i-1) + \nu_k\mu^2\sigma^2_{v,l}\text{Tr}(R_u) \end{aligned} \tag{34}$$

Hence, we obtain

$$\lim_{i\to\infty}E\boldsymbol{\sigma}^2_{l,k}(i) \approx \mu^2\sigma^2_{v,l}\text{Tr}(R_u) \tag{35}$$

That is, the estimate $\sigma^2_{l,k}(i)$ converges on average to the scaled multiple of $\sigma^2_{v,l}$, so that (28) is replaced by:

$$a_{l,k}(i) = \begin{cases} \frac{\sigma^{-2}_{l,k}(i)}{\sum_{j\in\mathcal{N}_k}\sigma^{-2}_{j,k}(i)}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

[1] We remark that a typo appears in the above expression in Table III in [2], where the noise variances appear written in the table instead of their inverses.

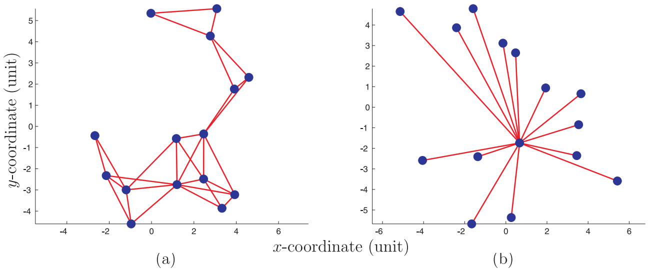

Fig. 1. Two network topologies (a) random topology and (b) star topology.

IV. Simulation Results

In this section, we simulate the mean-square performance under different combination rules. We compare the relative-variance rule (28) with the relative-degree-variance rule (29) and the uniform rule:

$$a_{l,k} = \begin{cases} 1/n_k, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

where every node k simply averages the estimates from its neighbors. We also compare with the alternative adaptive implementation from [10]. In the simulations, the noise at each node is uniformly generated between $[-35, -5]$ (dB), the regressors have the same covariance matrix, $R_{u,k} = I$, and the step-sizes are set to $\mu_k = 0.1$ and $\nu_k = 0.2$ for all k.

The network consists of 15 nodes and its topology is randomly generated (see Fig. 1(a)). The simulation results are depicted in Fig. 2. We observe that the relative-variance combination rule and the relative-degree-variance combination rule have similar performance in this case. We also observe that the adaptive implementation (36) converges to the relative-variance rule and outperforms the adaptive method of [10]; this is because expression (16) in [10] relies on an approximation for a certain covariance matrix $Q_{k,l}$ — this approximation is not used in our construction (28).

To further compare the relative-variance and relative-degree variance combination rules, we consider an extreme case of degree distribution: the star topology, where one center node connects to all the other nodes and the other nodes only connect to the center node (see Fig. 1(b)). The MSD is shown in Fig. 3, and in this case the proposed relative-variance combination rule outperforms the others rules.

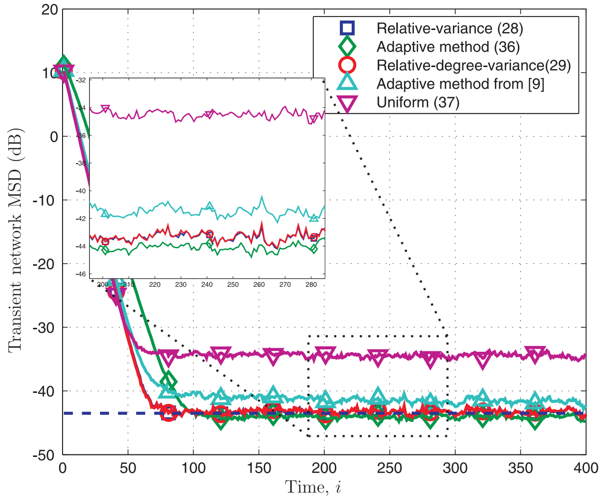

Fig. 2. Transient network MSD over a randomly connected network. The dashed line indicates the theoretical steady-state MSD with relative-variance combination rule.

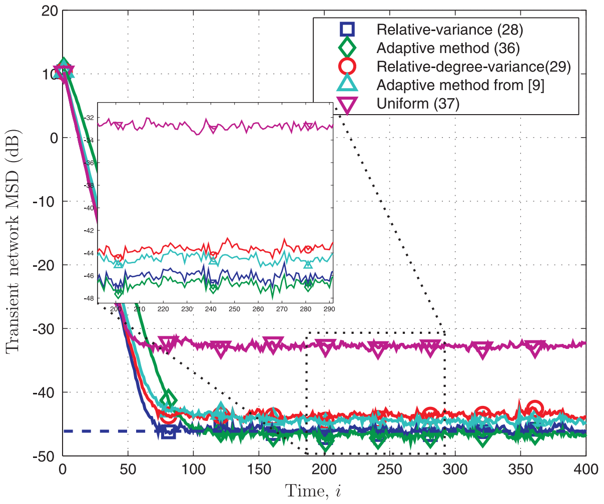

Fig. 3. Transient network MSD over a star-topology network. The dashed line indicates the theoretical steady-state MSD with relative-variance combination rule.

References

[1] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, pp. 3122-3136, July 2008.

[2] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035-1048, Mar. 2010.

[3] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, pp. 2038-2051, May 2011.

[4] S. Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Selected Topics on Signal Processing*, vol. 5, pp. 649-664, Mar. 2011.

[5] J. Li and A. H. Sayed, "Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing," *EURASIP Journal on Advances in Signal Processing*, 2011.

[6] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Proc. Int. Symp. Information Processing Sensor Networks (IPSN)*, pp. 63-70, Los Angeles, CA, Apr. 2005.

[7] P. Yang, R. A. Freeman, and K. M. Lynch, "Optimal information propagation in sensor networks," *Proc. Robotics and Automation*, pp. 3122-3127, Orlando, FL, May 2006.

[8] D. Jakovetic, J. Xavier, and J. M. F Moura, "Weight optimization for consenus algorithms with correlated switching topology," *IEEE Trans. on Signal Processing*, vol 58, pp. 3788-3801, July 2010.

[9] S. Sardellitti, M. Giona, and S. Barbarossa,"Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Trans. on Signal Processing*, vol. 58, pp. 826-842, Feb. 2010.

[10] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean-squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 9, pp. 4795-4810, Sep. 2010.

[11] A. H. Sayed, *Adaptive Filters*, NJ. Wiley, 2008.