

Improved least-squares-based combiners for diffusion networks

Jesus Fernandez-Bes*, Luis A. Azpicueta-Ruiz†, Magno T. M. Silva†, and Jerónimo Arenas-García*

*Universidad Carlos III de Madrid
Leganés, 28911 Spain
{jesusfbes, lazpicueta, jarenas}@tsc.uc3m.es

†Universidade de São Paulo
São Paulo 05508-010, Brazil
magno@lps.usp.br

Abstract—Adaptive networks have received great attention during recent years. In diffusion strategies, nodes diffuse their estimations to neighbors, and construct improved estimates by combining all information received by other nodes. When nodes work in heterogeneous conditions, it is reasonable to assign combination weights that take into account the performance of each node; thus, different schemes that implement adaptive combiners have been recently proposed. In this paper, we propose a novel scheme for adaptive combiners which attempts to minimize a least-squares cost function. The novelty in our proposal relies on making the adaptive combiners convex, by projection onto the standard simplex, what result in a numerically more stable implementation. The convergence and steady-state properties of the new scheme are analyzed theoretically, and its performance is experimentally evaluated with respect to existing methods.

I. INTRODUCTION AND PROBLEM FORMULATION

Recently, adaptive networks have gained considerable attention as an efficient way of estimating certain parameters of interest using the information from data collected by sensor nodes distributed over a region (see, e.g., [1], [2] and their references). In many applications, these networks must track the variations in the data statistics, which justifies the need for adaptiveness. In this context, different distributed estimation strategies can be considered. In particular, in diffusion networks, nodes diffuse their estimates to the network, so that each node can combine its own estimation with those received from neighboring nodes. Fig. 1 shows a diffusion network with $N$ nodes, where node $k$ combines the estimations received from neighbors $\mathcal{N}_k = \{1, 2, \ldots, r\}$ with its own estimation. Each node takes a measurement $\{d_k(n), u_k(n)\}$ at each time instant $n$, where $d_k(n)$ represents a desired signal and $u_k(n)$ is a length-$M$ input regressor vector, related through the usual linear regression model $d_k(n) = u_k^T(n)w_k(n-1) + v_k(n)$. Here, $v_k(n)$ is measurement noise while $w_k(n)$ is a length-$M$ column parameter vector. Note that $w_k(n)$ is assumed to be common to all nodes, and the network goal is its estimation.

Different approaches can be followed in order to obtain the combined estimation at each node. Recently, we have proposed a method [3] where each node performs its estimation in two steps: (i) it adapts and preserves its local estimation $\psi_k(n)$ and (ii) combines it with the combined estimates received from the neighboring nodes at the previous iteration $w_{\ell}(n-1)$. Thus, its combined estimation, that will be transmitted to its neighbors diffusing the information, is given by

$$
\{d_k(n), u_k(n)\} \rightarrow \{\psi_k(n)\} = c_{kk}(n)\psi_k(n) + \sum_{\ell \in \mathcal{N}_k} c_{k\ell}(n)w_{\ell}(n-1),
$$

where $\mathcal{N}_k$ represents the neighborhood of node $k$ excluding itself and $c_{k\ell}(n)$, for $k = 1, 2, \ldots, N$ and $\ell \in \mathcal{N}_k$, are the combination weights assigned by node $k$ to the different estimates combined in (1), with $\mathcal{N}_k$ being the neighborhood of node $k$ including itself. It should be evident that, with an adequate selection of $c_{k\ell}(n)$, $w_{\ell}(n)$ can potentially provide an improved estimate of $w_{\ell}(n)$ with respect to $\psi_k(n)$.

We should notice that most papers on diffusion networks assume fixed combination weights, whose values are computed based on the network topology only. However, these static combination rules do not take into account diversity among nodes, or the fact that these may be operating under different signal-to-noise ratio (SNR) conditions, resulting in suboptimal performance when the SNR varies across the network. For this reason, some schemes that implement adaptive combination weights have been recently proposed in [1], [2]. Although these adaptive solutions improve the performance of the networks when compared to static combiners, the learned combination parameters may still be suboptimal during the convergence or when tracking time-varying solutions, especially when different step sizes are used across the network nodes. This occurs due to the fact that some of the assumptions used in the derivations of these adaptive rules hold mainly in stationary scenarios and steady-state conditions.

Our proposal in [3], [4] updates the combiners using a least-squares (LS) rule. Compared to the solutions of [1],
[2], preliminary simulation results show that our approach outperforms the state-of-the-art scheme of [1] during the convergence and in tracking situations, but may present a slightly worse steady-state performance in stationary scenarios due to gradient noise and numerical instabilities inherent to the LS adaptation. This could be confirmed through a preliminary theoretical analysis of our scheme, obtained recently in [4].

In this paper, we extend our work of [3], [4] in two ways:

- At each adaptation step, we project the adaptive combiners to the standard simplex to obtain a convex combination. Although not strictly necessary, most of the works in this field consider convex combinations (i.e. \( \sum_{\ell \in N_k} c_{\ell k}(n) = 1 \) and \( c_{\ell k}(n) \geq 0, \forall \ell, k \)). In our case, adding convexity constraints results in a more stable implementation.

- We present a more refined theoretical analysis of the network MSD during convergence and steady-state operation. This analysis gives rise to a more accurate model than that of [4], mainly in the transient.

The experimental section of the paper provides a wide evaluation of the new adaptive rule, testing its ability to reduce the steady-state MSE level with respect to the existing LS-based scheme under different conditions at each node: Different adaptation characteristics, SNRs, power spectral densities, etc.

II. CONVEX ADAPTIVE COMBINERS BASED ON LS UPDATES

In this section, we briefly review the scheme of [3], [4] for the LS adaptation of network combiners \( c_{\ell k}(n) \), and introduce a minor modification to guarantee convex combiners at every iteration. The scheme of [3], [4] proposes a LS cost function, whose minimization leads to the following expression

\[
\tilde{e}_k(n) = P_k^{-1}(n)z_k(n), \quad \text{with} \quad c_{kk}(n) = 1 - 1^T\tilde{e}_k(n).
\]

Here, 1 is an all-one column vector with the same length of \( e_k(n) \), which is a vector containing the combination weights \( c_{\ell k}(n), \ell \in N_k \), while \( P_k(n) \) is a square symmetric matrix and \( z_k(n) \) a column vector with components

\[
[P_k(n)]_{m, \ell} = \sum_{i=1}^{n} \beta(n, i)\tilde{y}_{k,b}^{(i)}(i)\tilde{y}_{k,b}^{(\ell)}(i)
\]

\[
z^{(\ell)}_k(n) = \sum_{i=1}^{n} \beta(n, i)\bar{e}_k(i)\tilde{y}_{k,b}^{(\ell)}(i),
\]

for \( m, \ell = 1, \cdots, n_k \), with \( n_k \) the number of neighbors of node \( k \) (excluding itself), and where \( b^{(n)}_k \) is the index of the \( n \)th neighbor of node \( k \) and

\[
\tilde{y}_{k,b}^{(i)}(i) = u_k^T(i)[w_{b}^{(i)}(i) - 1 - \psi_k(i)].
\]

Finally, we assume that \( \beta(n, i) \) is a rectangular window, i.e., with value equal to one if \( n-i < L \), where \( L \) is the window length, and 0 elsewhere. More details about this scheme can be found in [3], [4].

Although the scheme we have just reviewed provides appropriate adaptation of the combiners during convergence and in tracking situations, we found in our previous works that it resulted in a slightly worse steady-state performance with respect to other methods. Looking closely at the causes for this degradation, we found that feedback loops in the diffusion of weights over the network can lead matrices \( P_k(n) \) close to singularity. When this occurs, the combiners tend to show very large and/or negative values.

In order to overcome the problem above, in this paper we force nodes to implement convex combinations of the received estimates and its local estimates, i.e., in addition to affinity constraints used in [3], [4] we force all combination weights to remain positive (and therefore, also smaller than one). Since convexity constraints are difficult to incorporate in recursive LS algorithms, we will simply modify the combiners at each iteration with a simple mechanism: Negative weights are first set to zero, and then all remaining weights are rescaled so that their sum remains one.

III. STATISTICAL ANALYSIS

The analysis is based on the Assumptions A1-A4 shown in Appendix. As a figure of merit of the performance of each node, we use the local mean square deviation (MSD), i.e., MSD\( _k(n) = \mathbb{E}[\|\bar{w}_k(n)\|^2] \), where \( \bar{w}_k(n) \equiv w_o(n) - w_k(n) \) is the weight-error vector for the combined estimates of each node. The performance of the whole network is measured by the Network MSD, given as MSD\( _N(n) = \frac{1}{N} \sum_{k=1}^{N} \text{MSD}_k(n) \).

We should also define the weight-error vector for the local estimates of each node, i.e. \( \psi_k(n) \equiv w_o(n) - \bar{w}_k(n) \). In the sequel, we obtain an analytical expression for MSD\( _k(n) \).

Subtracting both sides of (1) from \( w_o(n) \) and using Assumption A1, we obtain

\[
\bar{w}_k(n) - [1-c_{kk}(n)]q(n) = c_{kk}(n)\psi_k(n) + \sum_{\ell \in N_k} c_{\ell k}(n)\bar{w}_\ell(n-1).
\]

Multiplying both sides of (6) by their transposes, taking expectations on both sides of the resulting expression, and using Assumptions A1-A4, we arrive at

\[
\text{MSD}_k(n) \approx \mathbb{E}\left\{c_{kk}(n)^2\right\} \mathbb{E}[ar{w}_k(n)\bar{w}_k^T(n)]
\]

\[
+ \sum_{\ell \in N_k} \mathbb{E}\left\{c_{\ell k}(n)c_{mk}(n)\right\} \mathbb{E}[ar{w}_k(n-1)\bar{w}_\ell^T(n-1)]
\]

\[
+ 2 \mathbb{E}\left\{c_{kk}(n)c_{\ell k}(n)\right\} \mathbb{E}[\bar{w}_k(n)\bar{w}_\ell^T(n-1)]
\]

\[
- \mathbb{E}[(1-c_{kk}(n))^2]\mathbb{E}[\bar{w}_k^T(n-1)\bar{w}_k(n-1)]
\]

where we have defined the cross-covariance matrices of the local, combined and local-combined weight-error vectors, i.e.,

\[
\text{S}_{\ell m}(n) \equiv \mathbb{E}\{\bar{w}_\ell(n)\bar{w}_m^T(n)\},
\]

\[
\text{W}_{\ell m}(n) \equiv \mathbb{E}\{\hat{\bar{w}}_\ell(n)\hat{\bar{w}}_m^T(n)\},
\]

\[
\text{X}_{\ell m}(n) \equiv \mathbb{E}\{\hat{\psi}_\ell(n)\hat{\psi}_m^T(n)\},
\]
and the normalized step size $\bar{\eta}_k \triangleq \frac{\eta_k}{\sigma^2_{\epsilon_k}}$, where $\sigma^2_{\epsilon_k}$ represents the variance of the input signal of node $k$, with $k = 1, 2, \cdots N$ and $\text{Tr}[:]$ the trace of a matrix.

To complete the analysis, we must obtain analytical expressions for $E\{c_{ik}(n)c_{mk}(n)\}$, $S_{l\ell m}(n)$, $W_{l\ell m}(n)$, and $X_{l\ell m}(n)$. Obtaining a theoretical model for $E\{c_{ik}(n)c_{mk}(n)\}$ is a difficult task given its dependency on the combined estimation vectors. For this reason, we model only their optimal values. Thus, using (7) the approximation

$$E\{c_{ik}(n)c_{mk}(n)\} \approx c_{ik,o}(n)c_{mk,o}(n), \quad i, k, m \in N_k$$

where $c_{ik,o}(n)$ stands for the optimal value of $c_{ik}(n)$, we can obtain a theoretical value of the minimum achievable MSD.

Following derivations similar to those of Section II, it is straightforward to show that

$$E_k, a(n) = P^{-1}_{k,a}(n)\bar{\eta}_{k,a}(n),$$

where $P_{k,a}(n) = E\{\bar{y}_k(n)\bar{y}_k^T(n)\}$ and $\bar{\eta}_{k,a}(n) = E\{\epsilon_k(n)\bar{y}_k(n)\}$, with $\bar{y}_k(n) = [\bar{y}_{k1,1}, \cdots, \bar{y}_{k1,n}]^T$. Noticing that

$$\tilde{y}_k(p) = u_k^T(n)\left[\bar{y}_k(n) - \bar{w}_k(p)\right],$$

it is possible to obtain analytical expressions for $P_{k,a}(n)$ and $\bar{\eta}_{k,a}(n)$ that are shown in Table I, where $f, g = 1, 2, \cdots, N_k$, and column vector $\bar{b}_k$ is again used for notational convenience. Recursions for the cross-covariance matrices are also provided in Table I. The proofs of these recursions will be shown elsewhere due to the lack of space.

To avoid an ill-conditioning of matrix $P_{k,a}(n)$ in the calculus of the model, it was regularized by loading its main diagonal with a small constant $10^{-10}$. Since nonnegative combination weights provide a more stable recursion without loss of performance, we also consider this constraint in the model. Additionally, we considered the following initializations for cross-covariance matrices: $S_{l\ell m}(\cdot - 1) = W_{l\ell m}(\cdot - 1) = W_{o}(\cdot - 1)w_{o}^T(\cdot - 1)$, and $X_{l\ell m}(\cdot - 1) = I$ (the identity matrix).

To verify the accuracy of the analysis, we have carried out a preliminary simulation with a very simple network topology composed of three interconnected nodes running normalized least mean-squares (NLMS) algorithms with step sizes $\mu_1 = \mu_3 = 0.1$ and $\mu_2 = 1$. We assume the stationary case, where $w_o$ is a vector of length $M = 50$ with random components taken from a uniform distribution between $-1$ and $1$. To study the ability of the algorithms to converge, an abrupt change is introduced at $n = 10^4$. Input regressors and observation noises are independent across the network nodes. Furthermore, the regressors $u_k(n)$ follow a multidimensional Gaussian distribution with zero mean and covariance matrix equal to the identity ($I$). The observation noise $v_k(n)$ is generated independently of $u_k(n)$ and also follows a Gaussian distribution with zero mean and variances selected to get a $\text{SNR}$ in the interval $[20, 30]$ dB.

Fig. 2 shows the network MSD for the algorithm with adaptive combination weights of [6] (named here ACW), for our previous scheme with LS combiners (ADN-LS) [3], and for the proposed ADN-LS scheme including the convexity constraints on the combination weights, denominated Improved ADN-LS (IADN-LS), whose performance is compared with the theoretical MSD for a network with three nodes with different step sizes. In ADN-LS, a rectangular window of $L = 500$ samples is considered.

![Network MSD for different algorithms](image)

**Fig. 2. Simulated (averaging over 500 independent realizations) and theoretical MSD for a network with three nodes with different step sizes. In ADN-LS, a rectangular window of $L = 500$ samples is considered.**

**Fig. 3. Simulated (averaging over 500 independent realizations) and theoretical MSD for a network with three nodes with different step sizes. In ADN-LS, a rectangular window of $L = 500$ samples is considered.**

As expected, the MSD from the real network exceeds to some extent the values predicted by the theoretical model, mostly because of gradient noise introduced by the adoption of the combination weights. Nevertheless, the deviation is not very significant and, more importantly, the model predicts the qualitative behavior of the network MSD and the time instants where MSD has roughly converged to $-20$ dB and for the transition from $-20$ to $-30$ dB.

**IV. NUMERICAL EXPERIMENTS**

In this section, we provide empirical evidence to assess the performance of our proposal in stationary and tracking scenarios considering the network depicted in Fig. 3, with the same topology used in [3] and [6]. Regarding the signals involved, input regressors $u_k(n)$ and observation noise $v_k(n)$ are independent across the network nodes. The observation noise $v_k(n)$ is generated following a zero-mean Gaussian with a variance selected to obtain a different $\text{SNR}$ at each node. We will consider two different input signals $u_k(n)$: 1) a white input signal following a multidimensional Gaussian with zero mean and covariance matrix equal to the identity; 2) An input signal generated as a first-order AR model with
TABLE I

| \(\mathbf{P}_{k,0}(n)\), \(\mathbf{z}_{k,0}(n)\), \(\mathbf{S}_{\ell m}(n)\), \(\mathbf{W}_{\ell m}(n)\), and \(\mathbf{X}_{\ell m}(n)\). |

\[
\begin{align*}
\mathbf{P}_{k,0}(n) &= \text{Tr} \left\{ \mathbf{S}_{kk}(n-1) + \mathbf{W}_{\ell m}(n-1) - \mathbf{X}_{\ell m}(n-1) \right\} \\
\mathbf{z}_{k,0}(n) &= \text{Tr} \left\{ \mathbf{S}_{kk}(n-1) - \mathbf{X}_{\ell m}(n-1) \right\} \\
\mathbf{S}_{\ell m}(n) &= \mathbf{S}_{\ell m}(n-1) - \mathbf{S}_{\ell m}(n-1) \mathbf{R}_{mm} - \mathbf{S}_{\ell m}(n-1) + \mathbf{S}_{\ell m}(n-1) \mathbf{R}_{mm} + \mathbf{Q}, \quad (\ell \neq m) \\
\mathbf{S}_{\ell m}(n) &= \mathbf{S}_{\ell m}(n-1) - \mathbf{S}_{\ell m}(n-1) \mathbf{R}_{dd} + \mathbf{R}_{dd} \mathbf{S}_{\ell m}(n-1) + \mathbf{S}_{\ell m}(n-1) \mathbf{R}_{dd} + \mathbf{R}_{dd} \mathbf{S}_{\ell m}(n-1) \mathbf{R}_{dd} \\
\mathbf{X}_{\ell m}(n) &= \mathbf{E}(c_{mm}(n)) \mathbf{S}_{\ell m}(n) + \mathbf{E}(c_{mm}(n)) \mathbf{X}_{\ell m}(n-1) + \mathbf{E}(c_{mm}(n)) \mathbf{Q} \\
\mathbf{W}_{\ell m}(n) &= \mathbf{E}(c_{\ell m}(n)c_{mm}(n)) \mathbf{S}_{\ell m}(n) + \sum_{p \in N_{L}} \sum_{r \in N_{m}} \mathbf{E}(c_{\ell m}(n)c_{mm}(n)) \mathbf{W}_{pr}(n-1) + \mathbf{E}(c_{\ell m}(n)c_{mm}(n)) \mathbf{X}_{\ell m}(n-1) \\
&+ \sum_{p \in N_{L}} \sum_{r \in N_{m}} \mathbf{E}(c_{\ell m}(n)c_{mm}(n)) \mathbf{X}_{\ell m}(n-1) \mathbf{R}_{mm} + \mathbf{E}(1 - c_{\ell m}(n)) \mathbf{Q},
\end{align*}
\]

![Network topology for the experiments. Shaded nodes are adapted with \(\mu_k = 1\) and the rest with \(\mu_k = 0.1\).](image1)

![Performance of the proposed IADN-LS, the ADN-LS, the ACW and the no combining scheme in a stationary scenario. (a) Considering white input signals. (b) With colored input signals.](image2)

transfer function \(0.6/(1 - 0.8z^{-1})\), fed with i.i.d. Gaussian noise with unit variance.

We compare the performance of both LS methods (ADN-LS and IADN-LS) with the ACW algorithm in terms of Network MSD, averaging all provided results over 500 independent realizations. In adaptation step of all the diffusion algorithms the nodes use an NLMS rule with two different step sizes \(\mu_k = 1\) and \(\mu_k = 0.1\), as seen in Fig. 3. LS algorithms have as free parameter the window length \(L\). Algorithm ACW has also a free parameter \(\nu\) in the combination rule that was empirically set to \(\nu = 0.01\), value that obtains the best steady-state error.

Focusing on the stationary scenario, the algorithms pursue to estimate a common vector of parameters \(w_o\) of length \(M = 50\) that were generated uniformly in the range \([-1, 1]\). In order to evaluate their capability to converge, we introduce an abrupt change in \(w_o\) in the middle of the simulation, remaining a similar SNR at each node.

Fig. 4 (a) shows the performance of the algorithms for the stationary scenario with white input \(u_k(n)\). The length of the squared window in both ADN-LS and IADN-LS is \(L = 2500\). As can be seen, both LS algorithms converge faster than ACW, and both three schemes achieve a very similar steady-state error. Interestingly, imposing convexity in the combination weights (algorithm IADN-LS), gives rise to a more stable performance allowing to use windows of this length without instabilities. Fig. 4 (b) shows the Network MSD considering colored input signals \(u_k(n)\). In this case, the improvement of LS-algorithms in convergence is even more significant (see temporal axis). Moreover, it is important to remark again the
the no combining scheme in a tracking scenario with white input signals. (a) $\text{Tr}(Q) = 5 \cdot 10^{-6}$. (b) $\text{Tr}(Q) = 0.02$.

stability of the new rule compared to ADN-LS of [3].

The performance has also been evaluated for a tracking situation where the vector of parameters of interest $w_\ell(n)$ varies according to the tracking model of [5, Eq. (7.2.8)] and also used in [3, 4]. We have considered a scenario of fast changes for $w_\ell(n)$ characterized by $\gamma = 0.99$, $Q = \sigma_q^2 I$, with $\sigma_q$ selected for two different velocities of changes in $w_\ell(n)$: $\text{Tr}(Q) = 0.02$, $\text{Tr}(Q) = 5 \cdot 10^{-6}$. In this case, we consider white input signals $u_k(n)$ and we have selected $L = 500$.

As it can be seen in Fig. 5, the fastest adaptation rate of our rule becomes more and more important (in terms of both convergence and steady-state error) when the change rate of $w_\ell(n)$ increases. Again, the higher stability of the improved rule with convex combiners not only removes the ‘bump’ happening in ADN-LS but also improves the overall performance.

V. CONCLUSIONS

In this paper, we have proposed a novel scheme for the adaptation of combination weights in diffusion networks. In our scheme combination weights are obtained as the solution of a LS problem, and after that, we impose a convexity constraint on these combination weights. In this way, we obtain a more stable implementation comparing with previous affine LS rules. The performance of the new rule has been theoretically and experimentally studied by means of experiments with different kinds of input signals, SNRs and scenarios. These experiments show a significant improvement with respect to previous works.

APPENDIX

The assumptions used in the statistical analysis are:

A1- Random-walk model: the optimal solution varies in a nonstationary environment as $w_\ell(n) = w_\ell(n-1) + q(n)$, where $q(n)$ is a zero-mean independent and identically distributed (i.i.d.) vector with positive-definite autocorrelation matrix $Q = E\{q(n)q^T(n)\}$, independent of the initial conditions $q_0, w_0(n)$ and of $\{u_k(n'), v_k(n')\}$ for all $k$ and $n'$ [5].

A2- The regressors are zero-mean with covariance matrix $R_{kk} = E\{u_k(n)u_k^T(n')\}$. The data regression is spatially independent, i.e., $R_{kk} = E\{u_k(n)u_k^T(n')\} = 0$, $k \neq \ell$. Furthermore, the noise process $v_k(n)$ is assumed to be temporally white and spatially independent, i.e. $E\{v_k(n)v_k(n')\} = 0$, for all $n \neq n'$ and $E\{v_k(n)v_k(n')\} = 0$, for all $n, n'$ whenever $k \neq \ell$. It is further assumed to be independent (not only uncorrelated) of $u_k(n')$, so that $E\{v_k(n)u_k(n')\} = 0$, for all $k, \ell, n$, and $n'$. Under Assumption A2, $q_0, q_0(n-1)$ is independent of $v_k(n)$ and of $u_k(n)$, for all $k$ and $\ell$. This condition is a part of the widely used independence assumptions in adaptive filter theory [5].

A3- The adaptation of the combination weights is slow when compared to the adaptation of the local and combined estimates. Therefore, the correlation between combination parameters and local and combined estimates can be disregarded. This assumption follows from observations: simulations show that the combination weights converge slowly compared to variations in the input $u_k(n)$ and thus to variations on the local and combined estimates.

A4- The number of coefficients $M$ is large enough for each element of $u_k(n)u_k^T(n)$ to be approximately independent from $\sum_{l=0}^{M-1}|u_k(n-l)|^2$. Furthermore, the regressors $u_k(n), k = 1, 2, \ldots, N$ are formed by a tapped-delay line with Gaussian entries [7].

REFERENCES


