DIFFUSION LMS OVER MULTITASK NETWORKS WITH NOISY LINKS

Roula Nassif (1), *Cédric Richard* (1), *Jie Chen* (2), *André Ferrari* (1), *Ali H. Sayed* (3)

(1) Université de Nice Sophia-Antipolis, France
(2) Northwestern Polytechnical University, Xi'an, China
(3) University of California, Los Angeles, USA

ABSTRACT

Diffusion LMS is an efficient strategy for solving distributed optimization problems with cooperating agents. In some applications, the optimum parameter vectors may not be the same for all agents. Moreover, agents usually exchange information through noisy communication links. In this work, we analyze the theoretical performance of the single-task diffusion LMS when it is run, intentionally or unintentionally, in a multitask environment in the presence of noisy links. To reduce the impact of these nuisance factors, we introduce an improved strategy that allows the agents to promote or reduce exchanges of information with their neighbors.

1. INTRODUCTION

Distributed optimization allows to address inference problems in a decentralized manner over networks, where nodes are allowed to exchange information with their neighbors to improve their local estimates. In *single-task* networks, all nodes are interested in estimating the same parameter vector. Among the existing cooperation rules for single-task problems, we are interested in diffusion strategies [1–4] since they are scalable, robust, and enable continuous learning.

In *multitask* networks, nodes are grouped into clusters and each cluster is interested in estimating its own parameter vector, that is, each cluster has its own task. Recent studies on diffusion strategies over multitask networks have focused on two scenarios. In a first scenario, it is assumed that nodes know which cluster they belong to, and multitask diffusion strategies were derived to exploit intra-cluster and inter-cluster information exchanges in a meaningful way [5–9]. In a second scenario, nodes do not know the cluster they belong to. Several research efforts have focused on analyzing the performance of diffusion strategies when they are run, intentionally or unintentionally, in a multitask environment. It is shown in [10], for example, that the diffusion iterates converge to a Pareto optimal solution when the optimization problem consists of a sum of individual costs with possibly different minimizers. It is further shown in [11] that, when the tasks are sufficiently similar to each other, the single-task diffusion LMS can still perform better than noncooperative strategies. To avoid poor results resulting from cooperation between neighbors with sufficiently different objectives, extended diffusion strategies with a clustering step are proposed in [11–14] to enable agents to identify which neighbors belong to the same cluster and which neighbors should be ignored.

Usually, the exchange of raw data and local estimates between nodes may be corrupted by noisy communication links. Useful results dealing with the consequences of noisy communications on diffusion LMS behavior are presented in [15–18] for single-task environments. In this paper, we are interested in studying the degradation in the mean and mean-square error performance that would result from running the same diffusion LMS algorithm over a multitask network in the presence of noisy communication links. The analytical results reveal the influence of each nuisance factor on the dynamics of the network, on the biases in the weight estimates, and on the mean-square error performance. Since the mean-square error depends on the combination coefficients, we also show how these coefficients can be adjusted efficiently during the learning process in order to enable agents to cooperate only with neighbors sharing the same objective, and to simultaneously reduce the effect of exchanging information through noisy links.

Notation. Normal font letters denote scalars. Boldface lowercase letters denote column vectors. Boldface uppercase letters denote matrices. The operator $\otimes$ refers to the Kronecker product and $\text{col}\{\cdot\}$ stacks the column vectors entries on top of each other. The set $\mathcal{N}_k$ denotes the neighbors of node k, $\mathcal{C}(k)$ denotes the cluster to which node k belongs, and $\mathcal{C}_i$ is the i-th cluster.

2. DIFFUSION LMS IN THE PRESENCE OF NOISY LINKS

Consider a connected network of N nodes. At each time instant i, node k collects a zero-mean scalar measurement $d_k(i)$ and a zero-mean $L\times 1$ regression vector $\boldsymbol{x}_k(i)$ with a positive covariance matrix denoted by $\boldsymbol{R}_{x,k} = \mathbb{E}\,\boldsymbol{x}_k(i)\boldsymbol{x}_k^\top(i)$. These data are assumed to be related to an $L \times 1$ unknown vector $\boldsymbol{w}_k^o$ via the linear model:

$$d_k(i) = \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k^o + z_k(i), \tag{1}$$

where $z_k(i)$ is a zero-mean measurement noise of variance $\sigma_{z,k}^2$. The noise process is assumed to be temporally white and spatially independent. The problem is to estimate $\boldsymbol{w}_k^o$ at each node k. To solve this problem, node k can minimize the mean-square error $J_k(\boldsymbol{w})$:

$$J_k(\boldsymbol{w}) = \mathbb{E}\,|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}|^2, \tag{2}$$

using a stochastic gradient algorithm of the LMS type. In this case, the performance at node k depends on the variance $\sigma_{z,k}^2$ [3].

In a single-task environment, all nodes are interested in estimating the same parameter vector $\boldsymbol{w}^o$, i.e., $\boldsymbol{w}_k^o = \boldsymbol{w}^o\ \forall k$. In this case, it was shown that the use of a diffusion LMS strategy for minimizing, in a fully-distributed manner, the following aggregate cost [1–3]:

$$J^{\text{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} \mathbb{E}\,|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}|^2, \tag{3}$$

improves the estimation accuracy. In this work, we consider the adapt-then-combine (ATC) form of diffusion LMS [1, 2]:

$$\boldsymbol{\psi}_k(i+1) = \boldsymbol{w}_k(i) + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{x}_\ell(i)\left[d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k(i)\right] \tag{4}$$

$$\boldsymbol{w}_k(i+1) = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_\ell(i+1) \tag{5}$$

The work of C. Richard and A. Ferrari was partly supported by the ANR and the DGA, France, (ANR-13-ASTR-0030). The work of A. H. Sayed was supported in part by NSF grant CIF-1524250 and ECCS-1407712.

where μ_k is a small positive step-size parameter at node k and $\boldsymbol{w}_k(i)$ is the estimate of $\boldsymbol{w}^o$ at node k and iteration i. The non-negative coefficients $c_{\ell k}$ and $a_{\ell k}$, which are used to scale the data $\{\boldsymbol{x}_\ell(i), d_\ell(i)\}$ and the intermediate estimates $\boldsymbol{\psi}_\ell(i+1)$ transmitted from node ℓ to node k, are zero if node ℓ is not connected to node k, that is, $\ell \notin \mathcal{N}_k$. These coefficients are the (ℓ, k)-th entries of a right-stochastic matrix $\boldsymbol{C}$ and a left-stochastic matrix $\boldsymbol{A}$, respectively.

Each step of the ATC algorithm (4)-(5) involves the transmission of information from node $\ell \in \mathcal{N}_k$ to node k. In the presence of noisy communication links, the ATC diffusion LMS algorithm becomes:

$$\boldsymbol{\psi}_k(i+1) = \boldsymbol{w}_k(i) + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{x}_{\ell k}(i) \, [d_{\ell k}(i) - \boldsymbol{x}_{\ell k}^\top(i) \boldsymbol{w}_k(i)], \quad (6)$$

$$\boldsymbol{w}_k(i+1) = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell k}(i+1). \quad (7)$$

where $\boldsymbol{x}_{\ell k}(i)$, $d_{\ell k}(i)$, and $\boldsymbol{\psi}_{\ell k}(i+1)$ are the noisy data received by node k from its neighbor ℓ. For modeling noisy communication links, we adopt the model proposed in [3, 18]:

$$\begin{aligned} d_{\ell k}(i) &= d_\ell(i) + z_{d,\ell k}(i), & (8)\\ \boldsymbol{x}_{\ell k}(i) &= \boldsymbol{x}_\ell(i) + \boldsymbol{z}_{x,\ell k}(i), & (9)\\ \boldsymbol{\psi}_{\ell k}(i) &= \boldsymbol{\psi}_\ell(i) + \boldsymbol{z}_{\psi,\ell k}(i), & (10) \end{aligned}$$

where $z_{d,\ell k}(i)$ is a scalar noise signal, $\boldsymbol{z}_{x,\ell k}(i)$ and $\boldsymbol{z}_{\psi,\ell k}(i)$ are noise vectors of dimension $L \times 1$. Note that this model is more general than in [15, 16] where diffusion LMS is considered without exchange of gradient information, that is, $\boldsymbol{C} = \boldsymbol{I}_N$.

In a multitask environment, the local costs $J_k(\boldsymbol{w})$ are not all minimized at the same location. It is shown in [10] that, in this case, when $\boldsymbol{C} = \boldsymbol{I}_N$, the ATC algorithm (4)-(5) leads to a Pareto optimum solution for (3). In [11], the authors study the behavior of the ATC algorithm (4)-(5) when it is run over a multitask environment, and analyze the critical role of the distance between tasks, $\boldsymbol{w}_k^o$. In this work, we extend [11] to the case of noisy communication links. Before proceeding, we introduce the following assumptions.

Assumption 1. The regressors $\boldsymbol{x}_k(i)$ arise from a zero-mean random process that is temporally white and spatially independent.

Assumption 2. The noises $z_{d,\ell k}(i)$, $\boldsymbol{z}_{x,\ell k}(i)$, and $\boldsymbol{z}_{\psi,\ell k}(i)$ are temporally white, spatially independent zero-mean random variables. We denote by $\sigma^2_{zd,\ell k}$, $\boldsymbol{R}_{zx,\ell k}$, and $\boldsymbol{R}_{z\psi,\ell k}$ their variance and covariance matrices, respectively.

Assumption 3. $\{z_{d,mn}(i_1)\}$, $\{\boldsymbol{z}_{x,pq}(i_2)\}$, $\{\boldsymbol{z}_{\psi,st}(i_3)\}$, $\{\boldsymbol{x}_k(i_4)\}$, and $\{z_\ell(i_5)\}$ are mutually independent for all $\{k, \ell, m, n, p, q, s, t\}$ and $\{i_1, i_2, i_3, i_4, i_5\}$.

Assumption 4. The step-sizes μ_k are sufficiently small so that terms depending on higher order powers of the step-sizes can be ignored.

3. PERFORMANCE ANALYSIS

Using model (1), the noisy data $\{d_{\ell k}(i), \boldsymbol{x}_{\ell k}(i)\}$ in (8)-(9) at node k can be related to the unknown vector $\boldsymbol{w}_\ell^o$ at node ℓ via the relation:

$$d_{\ell k}(i) = \boldsymbol{x}_{\ell k}^\top(i) \boldsymbol{w}_\ell^o + z_{\ell k}(i), \quad (11)$$

where we introduce the scalar zero-mean noise signal:

$$z_{\ell k}(i) = z_\ell(i) + z_{d,\ell k}(i) - \boldsymbol{z}_{x,\ell k}^\top(i) \, \boldsymbol{w}_\ell^o, \quad (12)$$

whose variance is:

$$\sigma^2_{z,\ell k} = \sigma^2_{z,\ell} + \sigma^2_{zd,\ell k} + (\boldsymbol{w}_\ell^o)^\top \boldsymbol{R}_{zx,\ell k} \boldsymbol{w}_\ell^o. \quad (13)$$

Let $\widetilde{\boldsymbol{w}}_k(i) \triangleq \boldsymbol{w}_k^o - \boldsymbol{w}_k(i)$ be the error vector at node k and time instant i. Using (11), the estimation error that appears in the adaptation step (6) can be written as:

$$d_{\ell k}(i) - \boldsymbol{x}_{\ell k}^\top(i) \boldsymbol{w}_k(i) = \boldsymbol{x}_{\ell k}^\top(i) \widetilde{\boldsymbol{w}}_k(i) + \boldsymbol{x}_{\ell k}^\top(i) \boldsymbol{u}_{\ell k}^o + z_{\ell k}(i), \quad (14)$$

where $\boldsymbol{u}_{\ell k}^o \triangleq \boldsymbol{w}_\ell^o - \boldsymbol{w}_k^o$. Let $\widetilde{\boldsymbol{w}}(i)$ and $\boldsymbol{w}^o$ denote the network block error vector and the network block optimum vector, namely,

$$\widetilde{\boldsymbol{w}}(i) \triangleq \text{col}\{\widetilde{\boldsymbol{w}}_k(i)\}_{k=1}^N, \qquad \boldsymbol{w}^o \triangleq \text{col}\{\boldsymbol{w}_k^o\}_{k=1}^N. \quad (15)$$

Using relation (14), the network error vector recursion for the diffusion strategy (6)-(7) can be written as:

$$\boxed{\widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{B}}(i) \, \widetilde{\boldsymbol{w}}(i) - \boldsymbol{g}(i) - \boldsymbol{r}(i) - \boldsymbol{z}_\psi(i+1),} \quad (16)$$

where

$$\begin{aligned} \boldsymbol{\mathcal{B}}(i) &= \boldsymbol{\mathcal{A}}^\top (\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{R}}(i)) & (17)\\ \boldsymbol{\mathcal{M}} &= \text{diag}\{\mu_k \boldsymbol{I}_L\}_{k=1}^N & (18)\\ \boldsymbol{\mathcal{R}}(i) &= \text{diag}\Big\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{x}_{\ell k}(i) \boldsymbol{x}_{\ell k}^\top(i)\Big\}_{k=1}^N & (19)\\ \boldsymbol{g}(i) &= \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}} \boldsymbol{s}(i) & (20)\\ \boldsymbol{r}(i) &= \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}} \boldsymbol{h}(i) - (\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{A}}^\top) \boldsymbol{w}^o & (21)\\ \boldsymbol{h}(i) &= \text{col}\Big\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{x}_{\ell k}(i) \boldsymbol{x}_{\ell k}^\top(i) \boldsymbol{u}_{\ell k}^o\Big\}_{k=1}^N & (22)\\ \boldsymbol{s}(i) &= \text{col}\Big\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{x}_{\ell k}(i) z_{\ell k}(i)\Big\}_{k=1}^N & (23)\\ \boldsymbol{z}_\psi(i+1) &= \text{col}\Big\{\sum_{\ell \in \mathcal{N}_k^-} a_{\ell k} \boldsymbol{z}_{\psi,\ell k}(i+1)\Big\}_{k=1}^N & (24) \end{aligned}$$

Based on recursion (16), we examine the performance of the ATC algorithm (6)-(7) in the mean and mean-square-error sense. Due to space limitations, we only list the main results of the analysis and omit the proofs.

3.1. Mean behavior analysis

Taking the expectation of both sides of (16), we get:

$$\mathbb{E}\, \widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{B}} \, \mathbb{E}\, \widetilde{\boldsymbol{w}}(i) - \boldsymbol{g} - \boldsymbol{r}, \quad (25)$$

where

$$\begin{aligned} \boldsymbol{\mathcal{B}} &= \boldsymbol{\mathcal{A}}^\top (\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{R}}), & (26)\\ \boldsymbol{g} &= \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}} \boldsymbol{s}, & (27)\\ \boldsymbol{r} &= \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}} \boldsymbol{h} - (\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{A}}^\top) \boldsymbol{w}^o, & (28)\\ \boldsymbol{\mathcal{R}} &= \text{diag}\Big\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} (\boldsymbol{R}_{x,\ell} + \boldsymbol{R}_{zx,\ell k})\Big\}_{k=1}^N, & (29)\\ \boldsymbol{h} &= \text{col}\Big\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} (\boldsymbol{R}_{x,\ell} + \boldsymbol{R}_{zx,\ell k}) \boldsymbol{u}_{\ell k}^o\Big\}_{k=1}^N & (30)\\ \boldsymbol{s} &= -\boldsymbol{R}_{zx} \boldsymbol{w}^o, & (31) \end{aligned}$$

and, $\boldsymbol{R}_{zx}$ is the $N \times N$ block matrix whose (ℓ, k)-th block is $c_{k\ell} \boldsymbol{R}_{zx,k\ell}$. It can be verified that for any initial conditions, the diffusion LMS algorithm (6)-(7) converges in the mean if the step-sizes μ_k satisfy:

$$0 < \mu_k < \frac{2}{\lambda_{\max}\{\sum_{\ell \in \mathcal{N}_k} c_{\ell k} (\boldsymbol{R}_{x,\ell} + \boldsymbol{R}_{zx,\ell k})\}} \quad (32)$$

for $k = 1, \ldots, N$, where $\lambda_{\max}\{\cdot\}$ is the maximum eigenvalue of its matrix argument. The asymptotic mean bias is given by:

$$\boldsymbol{b} = \lim_{i \to \infty} \mathbb{E}\, \widetilde{\boldsymbol{w}}(i) = -(\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{B}})^{-1} (\boldsymbol{g} + \boldsymbol{r}). \quad (33)$$

Note that $\boldsymbol{g}$ in (27) is zero if the regressors are not corrupted by noise during their transmission. The vector $\boldsymbol{r}$ in (28) is zero if there is no cooperation between neighbors with different objectives. Finally, we observe from (32) that the noise corrupting the communication of regressors affects the stability condition.

3.2. Mean-square-error behavior analysis

We now study the behavior of the variance $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\Sigma}}$, where $\boldsymbol{\Sigma}$ is a positive semi-definite matrix that we are free to choose. Let $\boldsymbol{\sigma}$ denote the vectorized version of $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\sigma} \triangleq \text{vec}(\boldsymbol{\Sigma})$. We obtain from (16) the following equation:

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}} \approx \mathbb{E}\,\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}} + [\text{vec}\{\boldsymbol{T}^\top\} - 2(\boldsymbol{\mathcal{B}}\,\mathbb{E}\,\widetilde{\boldsymbol{w}}(i)) \otimes (\boldsymbol{g}+\boldsymbol{r})]^\top \boldsymbol{\sigma}, \quad (34)$$

where we use the notation $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}}$ and $\|\boldsymbol{x}\|^2_{\boldsymbol{\sigma}}$ interchangeably to denote the same quantity $\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$. The terms in (34) are given by:

$$\boldsymbol{\mathcal{F}} = \mathbb{E}\,\boldsymbol{\mathcal{B}}^\top(i) \otimes \boldsymbol{\mathcal{B}}^\top(i) \approx \boldsymbol{\mathcal{B}}^\top \otimes \boldsymbol{\mathcal{B}}^\top, \quad (35)$$

$$\boldsymbol{\mathcal{C}} = \boldsymbol{C} \otimes \boldsymbol{I}_L \quad (36)$$

$$\boldsymbol{T} = \boldsymbol{G} + \boldsymbol{R}_{z\psi} + \boldsymbol{R}_r + 2\boldsymbol{G}_r, \quad (37)$$

$$\boldsymbol{G} = \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{S}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{A}}, \quad (38)$$

$$\boldsymbol{S} = \text{diag}\{\sigma^2_{z,k}\,\boldsymbol{R}_{x,k}\}_{k=1}^N + \boldsymbol{s}\boldsymbol{s}^\top + \text{diag}\{\boldsymbol{D}_k\}_{k=1}^N, \quad (39)$$

$$\boldsymbol{R}_r = \boldsymbol{r}\boldsymbol{r}^\top + \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\,\text{diag}\{\boldsymbol{H}_k\}_{k=1}^N\,\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{A}}, \quad (40)$$

$$\boldsymbol{G}_r = \boldsymbol{g}\boldsymbol{r}^\top - \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\,\text{diag}\{\boldsymbol{J}_k\}_{k=1}^N\,\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{A}}, \quad (41)$$

$$\boldsymbol{R}_{z\psi} = \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_k^-} a^2_{\ell k}\,\boldsymbol{R}_{z\psi,\ell k}\Big\}_{k=1}^N, \quad (42)$$

and

$$\boldsymbol{H}_k = \sum_{\ell\in\mathcal{N}_k} c^2_{\ell k}\Big(\boldsymbol{R}_{x,\ell}\boldsymbol{u}^o_{\ell k}(\boldsymbol{u}^o_{\ell k})^\top\boldsymbol{R}_{zx,\ell k} + (\boldsymbol{u}^o_{\ell k})^\top\boldsymbol{R}_{zx,\ell k}\boldsymbol{u}^o_{\ell k}\boldsymbol{R}_{x,\ell} + (\boldsymbol{u}^o_{\ell k})^\top\boldsymbol{R}_{x,\ell}\boldsymbol{u}^o_{\ell k}\boldsymbol{R}_{zx,\ell k} + \boldsymbol{R}_{zx,\ell k}\boldsymbol{u}^o_{\ell k}(\boldsymbol{u}^o_{\ell k})^\top\boldsymbol{R}_{x,\ell}\Big),$$

$$\boldsymbol{D}_k = \sum_{\ell\in\mathcal{N}_k} c^2_{\ell k}\Big((\sigma^2_{zd,\ell k} + \|\boldsymbol{w}^o_\ell\|^2_{\boldsymbol{R}_{zx,\ell k}})\boldsymbol{R}_{x,\ell} + (\sigma^2_{zd,\ell k} + \sigma^2_{z,\ell})\boldsymbol{R}_{zx,\ell k}\Big),$$

$$\boldsymbol{J}_k = \sum_{\ell\in\mathcal{N}_k} c^2_{\ell k}\Big(\boldsymbol{R}_{x,\ell}\boldsymbol{u}^o_{\ell k}(\boldsymbol{w}^o_\ell)^\top\boldsymbol{R}_{zx,\ell k} + (\boldsymbol{w}^o_\ell)^\top\boldsymbol{R}_{zx,\ell k}\boldsymbol{u}^o_{\ell k}\boldsymbol{R}_{x,\ell}\Big).$$

The approximations in (34)-(35) follow from Assumption 4. Furthermore, the evaluation of some terms of $\boldsymbol{T}$ in (37) requires the calculation of 4-th order moments that are approximated by products of 2-nd order moments. Under these approximations, and for sufficiently small step-sizes, it can be verified that for any initial conditions, the diffusion LMS algorithm (6)-(7) is mean-square stable if the error recursion (16) is mean stable and the matrix $\boldsymbol{\mathcal{F}}$ is stable.

From equation (34), we obtain a recursion that enables us to evaluate the variance over time [5]:

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}} = \mathbb{E}\,\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\sigma}} - \mathbb{E}\,\|\widetilde{\boldsymbol{w}}(0)\|^2_{(\boldsymbol{I}-\boldsymbol{\mathcal{F}})\boldsymbol{\mathcal{F}}^i\boldsymbol{\sigma}} + [\text{vec}\{\boldsymbol{T}^\top\}]^\top\boldsymbol{\mathcal{F}}^i\boldsymbol{\sigma} - 2[(\boldsymbol{\mathcal{B}}\mathbb{E}\,\widetilde{\boldsymbol{w}}(i) \otimes (\boldsymbol{g}+\boldsymbol{r}))^\top + \boldsymbol{\Gamma}(i)]\boldsymbol{\sigma}, \quad (43)$$

where

$$\boldsymbol{\Gamma}(i) = \boldsymbol{\Gamma}(i-1)\boldsymbol{\mathcal{F}} + [(\boldsymbol{\mathcal{B}}\,\mathbb{E}\,\widetilde{\boldsymbol{w}}(i-1)) \otimes (\boldsymbol{g}+\boldsymbol{r})]^\top(\boldsymbol{\mathcal{F}} - \boldsymbol{I}_{(LN)^2}),$$

with $\boldsymbol{\Gamma}(0) = \boldsymbol{0}_{1\times(LN)^2}$. Once convergence is achieved, we obtain in steady state:

$$\lim_{i\to\infty}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{(\boldsymbol{I}-\boldsymbol{\mathcal{F}})\boldsymbol{\sigma}'}\} = [\text{vec}\{\boldsymbol{T}^\top\} - 2(\boldsymbol{\mathcal{B}}\boldsymbol{b}) \otimes (\boldsymbol{g}+\boldsymbol{r})]^\top\boldsymbol{\sigma}'.$$

For the simulations, we shall use $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{I}_{LN}$, i.e., $\boldsymbol{\sigma} = \frac{1}{N}\text{vec}\{\boldsymbol{I}_{LN}\}$ and $\boldsymbol{\sigma}' = \frac{1}{N}(\boldsymbol{I}-\boldsymbol{\mathcal{F}})^{-1}\text{vec}\{\boldsymbol{I}_{LN}\}$. This will allow us to evaluate the network mean-square-deviation (MSD).

4. OPTIMIZING THE COMBINATION WEIGHTS

In order to attenuate the negative effects of running (6)-(7) in a multi-task environment in the presence of noisy links, we suggest to follow the strategy used in [11, 12, 18]. It consists of adjusting the combination weights $\{a_{\ell k}\}$ by minimizing the instantaneous MSD at each node k given by:

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_k(i+1)\|^2 = \mathbb{E}\,\Big\|\boldsymbol{w}^o_k - \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell k}(i+1)\Big\|^2 \quad (44)$$

Relaxing problem (44) as suggested in [12, 18] leads to

$$\begin{aligned} &\min_{\{a_{\ell k}\}} \quad \sum_{\ell=1}^N a^2_{\ell k}\|\widehat{\boldsymbol{w}}^o_k - \boldsymbol{\psi}_{\ell k}(i+1)\|^2, \\ &\text{s. t.} \quad \sum_{\ell=1}^N a_{\ell k} = 1, \;\; a_{\ell k} \geq 0, \;\; \text{and} \;\; a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k, \end{aligned} \quad (45)$$

where $\widehat{\boldsymbol{w}}^o_k$ is some approximation for $\boldsymbol{w}^o_k$. One useful approximation is the local one-step approximation used in [11]:

$$\widehat{\boldsymbol{w}}^o_k(i+1) = \boldsymbol{\psi}_k(i+1) + \mu_k\frac{\boldsymbol{q}_k(i)}{\|\boldsymbol{q}_k(i)\|+\epsilon}, \quad (46)$$

where ϵ is a small positive value to avoid the vanishing of the denominator and

$$\boldsymbol{q}_k(i) = \boldsymbol{x}_k(i)[d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{\psi}_k(i+1)]. \quad (47)$$

Introducing the notation $\gamma^2_{\ell k}(i+1) \triangleq \|\widehat{\boldsymbol{w}}^o_k(i+1) - \boldsymbol{\psi}_{\ell k}(i+1)\|^2$, the solution of problem (45) is given by:

$$a_{\ell k}(i+1) = \frac{\gamma^{-2}_{\ell k}(i+1)}{\sum_{n\in\mathcal{N}_k}\gamma^{-2}_{nk}(i+1)}, \qquad \ell \in \mathcal{N}_k. \quad (48)$$

It is noticed in [14] that the clustering strategy (48) may suffer from a larger probability of false alarm, that is, $a_{\ell k}$ may tend to zero even in situations where nodes k and ℓ share the same task. To overcome this problem, we propose to smooth $\gamma^2_{\ell k}(i+1)$ as follows:

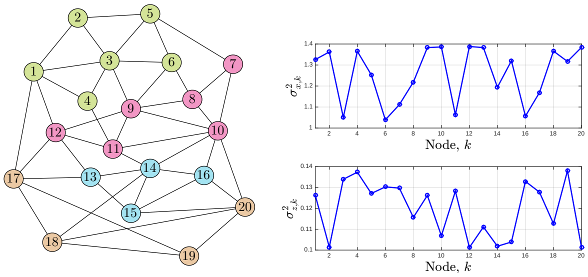
$$\boxed{\gamma^2_{\ell k}(i+1) = (1-\nu_k)\gamma^2_{\ell k}(i) + \nu_k\|\widehat{\boldsymbol{w}}^o_k(i+1) - \boldsymbol{\psi}_{\ell k}(i+1)\|^2} \quad (49)$$

where $\nu_k \in [0, 1]$ is a forgetting factor.

The protocol for adjusting the combination weights in [12, 18] differs from (48) and (49) by using the estimate $\boldsymbol{w}_k(i)$ as an approximation for $\boldsymbol{w}^o_k$ in (46). Moreover, the clustering strategy proposed in [11] does not include the smoothing step (49). As shown by simulations, this step reduces the probability of erroneous clustering especially in the presence of noisy links.

5. SIMULATION RESULTS

We considered the connected network in Fig. 1(a), consisting of 20 agents grouped into 4 clusters: $\mathcal{C}_1 = \{1,\ldots,6\}$, $\mathcal{C}_2 = \{7,\ldots,12\}$, $\mathcal{C}_3 = \{13,\ldots,16\}$, and $\mathcal{C}_4 = \{17,\ldots,20\}$. Regressors were 2×1 zero-mean Gaussian random vectors with covariance matrices $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_2$. Noises $z_k(i)$ were zero-mean i.i.d. Gaussian with variance $\sigma^2_{z,k}$. Variances $\sigma^2_{x,k}$ and $\sigma^2_{z,k}$ are shown in Fig. 1(b).



(a) Network topology. (b) Regression and noise variances.

Fig. 1. Experimental setup.

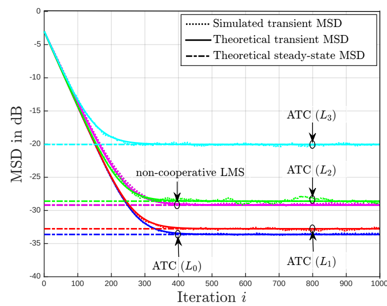


Fig. 2. Network MSD behavior for different levels of noise.

Noises over links $z_{d,\ell k}(i)$, $z_{x,\ell k}(i)$, and $z_{\psi,\ell k}(i)$ were also zero-mean i.i.d. Gaussian random variables of variance σ_z^2 for all $\ell \in \mathcal{N}_k \setminus \{k\}$. The objectives were uniformly distributed on a circle of radius r centered at $\boldsymbol{w}_c = [0.5, -0.5]^\top$. More details on the experimental setup appear in [11]. We used a constant step-size $\mu_k = 0.01$ for all k. The results were averaged over 100 runs. Let $\boldsymbol{A}_0$ and $\boldsymbol{C}_0$ be uniform combination matrices, namely, $a_{\ell k} = |\mathcal{N}_k|^{-1}$ for $\ell \in \mathcal{N}_k$ and $c_{\ell k} = |\mathcal{N}_\ell|^{-1}$ for $k \in \mathcal{N}_\ell$, respectively.

First, we considered the case where tasks are close to each other by setting $r = 0.02$. We ran the ATC algorithm (6)-(7) with $\boldsymbol{C}_0$ and $\boldsymbol{A}_0$ for 4 levels of noise over links: $L_0 : \sigma_z^2 = 0$, $L_1 : \sigma_z^2 = 10^{-4}$, $L_2 : \sigma_z^2 = 10^{-3}$, and $L_3 : \sigma_z^2 = 10^{-2}$. The non-cooperative LMS was also considered by setting $\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}_N$. The network MSD learning curves are reported in Fig. 2. It can be observed that the theoretical findings match well the simulated curves. Furthermore, for a certain degree of similarity between tasks, diffusion LMS with perfect information exchange can still deliver superior performance compared to non-cooperative strategies despite the bias introduced by the multitask scenario. The performance decreases when the level of noise over links increases.

In the following, we use $\boldsymbol{A}(0) = \boldsymbol{A}_0$ and $\boldsymbol{C}(0) = \boldsymbol{C}_0$. The coefficients $c_{\ell k}(i)$ were set such that $\boldsymbol{C}(i+1) = \boldsymbol{A}^\top(i)$. Three different protocols for adjusting the combination coefficients $a_{\ell k}$ were considered: the rule (46)-(49) with $\nu_k = 0.05$ and $\epsilon = 0.01$, the rule in [11] with $\epsilon = 0.01$, and the rule in [12, 18] with $\nu_k = 0.05$. We ran algorithm (6)-(7) for $r = 0.02$ and $\sigma_z^2 = 10^{-2}$ with the adaptive combination rules mentioned earlier. Figure 3 illustrates the network MSD behavior for these algorithms. It appears that all these rules allow us to reduce the negative effects of noise over communication links. Our rule (46)-(49) achieves the best performance.

To test the clustering ability of the ATC algorithm (6)-(7) with adaptive combiners in the presence of noisy links, we increased the distance between tasks by setting $r = 1$. In Fig. 4, we compare the network MSD of the algorithm under perfect (left plot) and imperfect (right plot) information exchange, by setting $\sigma_z^2 = 0$ and $\sigma_z^2 = 10^{-4}$, respectively. In each case, we considered fixed combiners $\{a_{\ell k}, c_{\ell k}\}$ and adaptive combiners using the 3 different protocols mentioned earlier. As shown by the experiments, the use of adaptive combiners is necessary when the tasks are not close enough. Furthermore, our rule (46)-(49) provides the best performance especially in the presence of noisy information exchange. To better analyze this behavior, we report in Fig. 5 the probabilities of erroneous clustering decisions of types I and II. Consider the link $\mathcal{L}_{\ell,k}$ connecting k to its neighbor ℓ. The probability of type I for node k is the probability that $\mathcal{L}_{\ell,k}$ is erroneously dropped while $\boldsymbol{w}_k^o = \boldsymbol{w}_\ell^o$. The probability of type II is the probability that $\mathcal{L}_{\ell,k}$ is erroneously connected while $\boldsymbol{w}_k^o \neq \boldsymbol{w}_\ell^o$. We considered that the link is dropped off if $a_{\ell k}(i) < 0.05$. The experiments show that the rule in [12, 18] suffers in the presence of imperfect information exchange. The rule in [11] tends to drop off links between agents of the same clusters, notably in the presence of noisy links. Our rule (46)-(49) is able to perform a perfect clustering in the presence and absence of noisy links since both types of probabilities are decaying to zero.

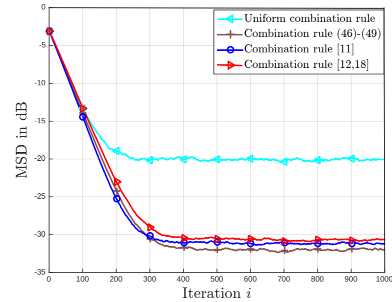


Fig. 3. Network MSD for different combination rules (close tasks).

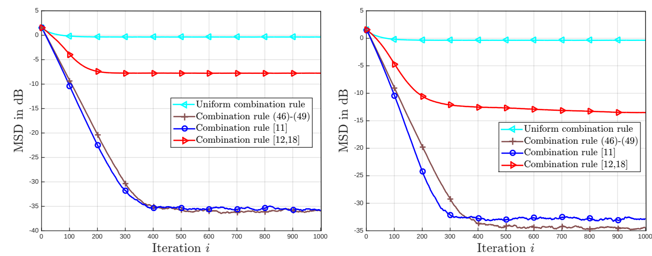


Fig. 4. Network MSD for different combination rules (distant tasks) with perfect (left) and imperfect (right) information exchange.

6. CONCLUSION

This work analyzed the performance of the diffusion LMS when it is run in a multitask environment in the presence of noisy links. An online strategy for adapting the combination coefficients was proposed to reduce the impact of these nuisance factors.

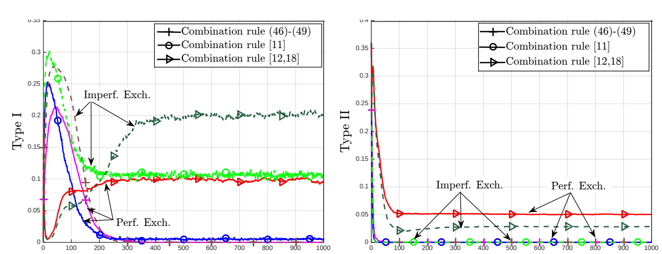


Fig. 5. Erroneous clustering decisions of type I (left) and II (right) with perfect (solid) and imperfect (dashed) information exchange.

7. REFERENCES

[1] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, July 2008.

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

[3] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, Elsevier, Ed., vol. 3, pp. 322–454. Elsevier, 2014.

[4] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

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

[6] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *Submitted for publication.* Also available as arXiv:1509.01360, Sep. 2015.

[7] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE Int. Workshop on Machine Learn. for Signal Process. (MLSP)*, Reims, France, Sep. 2014, pp. 1–6.

[8] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks-Part I: Sequential node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.

[9] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3448–3460, July 2015.

[10] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, April 2013.

[11] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. on Signal Process.*, vol. 63, no. 11, pp. 2733–2748, June 2015.

[12] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. Int. Workshop Cognitive Inf. Process. (CIP)*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[13] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, July 2015.

[14] S. Khawatmi, A. Zoubir, and A. H. Sayed, "Decentralized clustering over adaptive networks," in *Proc. EUSIPCO*, Nice, France, Sep. 2015, pp. 1–5.

[15] R. Abdolee and B. Champagne, "Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels," in *Proc. IEEE Int. Conf. Dist. Comput. Sensor Systems (DCOSS)*, Barcelona, Spain, June 2011, pp. 1–6.

[16] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[17] S.-Y. Tu and A. H. Sayed, "Adaptive networks with noisy links," in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, Houston, TX, Dec. 2011, pp. 1–5.

[18] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, July 2012.