Doubly compressed diffusion LMS over adaptive networks

Ibrahim El Khalil Harrane, Rémi Flamary, Cédric Richard

Doubly compressed diffusion LMS over adaptive networks

Ibrahim El Khalil Harrane, Rémi Flamary, Cédric Richard
Université Côte d'Azur, OCA, CNRS, France
ibrahim.harrane@oca.eu, remi.flamary@unice.fr, cedric.richard@unice.fr

***Abstract*—Diffusion LMS is an efficient strategy for solving distributed optimization problems with cooperating agents. Nodes are interested in estimating the same parameter vector and exchange information with their neighbors to improve their local estimates. Successful implementation of such applications relies on a substantial amount of communication resources. In this paper, we introduce diffusion LMS strategies that offer significantly reduced communication load without compromising performance. We perform analyses in the mean and mean-square sense of these algorithms. Simulations results are provided to confirm the theoretical findings.**

I. Introduction

Distributed adaptation over network has become an active research area with the increasing popularity of wireless sensor networks and, more recently, with the advent of the internet of things. An accessible overview of results in the field can be found in [1]–[5]. The interconnected agents continuously collect data, learn and adapt by carrying out data mining tasks on their own measurements as well as their neighbors. Albeit outperformed by centralized strategies, diffusion strategies are more robust and resilient to agents and links failures. Furthermore, their scalability and flexibility make them attractive for addressing inference problems in a collaborative manner. Among other possible strategies [6]–[8], diffusion LMS plays a central role with its enhanced efficiency and low complexity, and has been extensively studied in the literature in single task [1]–[3] and multitask frameworks [9]–[13]. It was also considered in other contexts such as nonlinear system identification [14] and dictionary learning [15].

Consider an interconnected network of N nodes. The aim of each node is to estimate an $L \times 1$ unknown vector $\boldsymbol{w}^o$ from collected measurements. Node k has access to local streaming measurements $\{d_k(i), \boldsymbol{u}_{k,i}\}$ where $d_k(i)$ is a scalar zero-mean reference signal, and $\boldsymbol{u}_{k,i}$ is an $L \times 1$ zero-mean regression vector with covariance matrix $\boldsymbol{R}_{u_k} = \mathbb{E}\{\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^\top\} > 0$. The data at agent k and time i are assumed to be related via the linear regression model:

$$d_k(i) = \boldsymbol{u}_{k,i}^\top \boldsymbol{w}^o + v_k(i) \tag{1}$$

where $\boldsymbol{w}^o$ is the unknown parameter vector to be estimated, and $v_k(i)$ is a zero-mean i.i.d. noise with variance $\sigma_{v,k}^2$. The noise $v_k(i)$ is assumed to be independent of any other signal. Let $J_k(\boldsymbol{w})$ be a differentiable convex cost function at agent k. In this paper, we shall consider the mean-square-error criterion:

$$J_k(\boldsymbol{w}) = E\{|d_k(i) - \boldsymbol{u}_{k,i}^\top \boldsymbol{w}|^2\} \tag{2}$$

Diffusion LMS strategies seek the minimizer of the following aggregate cost function:

$$J^{\text{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} J_k(\boldsymbol{w}) \tag{3}$$

in a cooperative manner. Let $\boldsymbol{w}_{k,i}$ denote the estimate of the minimizer of (3) at node k and time instant i. The general structure of diffusion LMS in its Adapt-then-Combine (ATC) form is given by:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) \tag{4}$$

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

where $\hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) = -\boldsymbol{u}_{\ell,i}[d_\ell(i) - \boldsymbol{u}_{\ell,i}^\top \boldsymbol{w}_{k,i-1}]$, $\mathcal{N}_k$ denotes the neighborhood of node k including k, and μ_k is a positive step-size. Nonnegative coefficients $\{a_{\ell k}\}$ and $\{c_{\ell k}\}$ define a left-stochastic matrix $\boldsymbol{A}$ and a right-stochastic matrix $\boldsymbol{C}$, respectively. Diffusion LMS provides a scalable estimation framework with high flexibility for ad-hoc deployment. Nevertheless, the requirement for all nodes to exchange their current estimates $\boldsymbol{w}_{k,i-1}$, $\hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1})$ and $\boldsymbol{\psi}_{k,i}$ with their neighbors at each iteration imposes a substantial burden on communication and energy ressources. Therefore, reducing the communication cost while maintaining the benefits of cooperation is of major importance for systems with limited energy budget such as wireless sensor networks.

In recent years, several strategies have been proposed to address this issue. They can be divided into two categories. On the one hand, some authors propose to restrict the number of active links between neighbors at each time instant [5]. On the other hand, there are authors that recommend to reduce the communication load by projecting parameter vectors onto lower dimensional spaces [16], or transmitting only partial parameter vectors [17]–[19]. In [17]–[19], the combination step (5) is redefined as:

$$\boldsymbol{w}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{\ell k} \Big(\boldsymbol{H}_{\ell,i}\boldsymbol{\psi}_{\ell,i} + [\boldsymbol{I} - \boldsymbol{H}_{\ell,i}]\boldsymbol{\psi}_{k,i} \Big) \tag{6}$$

where $\boldsymbol{H}_{\ell,i}$ is a diagonal entry-selection matrix with M ones and $L-M$ zeros on its diagonal. This means that the nodes can use the entries of their own intermediate estimates in lieu of the ones from the neighbors that have not been communicated. Matrix $\boldsymbol{H}_{\ell,i}$ can be deterministic, or can randomly select M entries from all entries.

The literature has mainly focused on the reformulation (6) of the combination step (4) of the diffusion LMS. This means that, with the so-called *partial-diffusion LMS*, only the intermediate estimates $\boldsymbol{\psi}_{k,i}$ are partially shared by neighboring nodes. However, it is also of interest to consider the adaptation step as it accounts for a major part of information exchanges.

Our main contribution in this paper is to consider and compress every transmitted information over the network, namely, local estimates $\boldsymbol{w}_{i,k}$ and estimated gradients $\hat{\nabla}_w J_k(\boldsymbol{w}_{\ell,i-1})$. In a preliminary step, we study the case where only the local estimates in (4) are partially transmitted. Next, we study the case where both the estimates and the estimated gradients are partially transmitted. This method allows a control of network communication flows by setting the numbers M and M_∇ of selected entries in $\boldsymbol{w}_{i,k}$ and $\hat{\nabla}_w J_k(\boldsymbol{w}_{\ell,i-1})$, respectively. Finally, we carry out a theoretical analysis of the algorithms in the mean and mean-square sense, and we perform numerical experiments to confirm the theoretical findings.

Notation: Boldface small letters denote vectors. All vectors are column vectors. Boldface capital letters denote matrices. The (k,ℓ)-th entry of a matrix is denoted by $(\cdot)_{k\ell}$, and the (k,ℓ)-th block of a block matrix is denoted by $[\,\cdot\,]_{k\ell}$. Matrix trace is denoted by trace$\{\cdot\}$. The expectation operator is denoted by $\mathbb{E}\{\cdot\}$. The identity matrix of size N is denoted by $\boldsymbol{I}_N$, and the all-one vector of length N is denoted by $\boldsymbol{1}_N$. We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself, and $|\mathcal{N}_k|$ its cardinality. The operator col$\{\cdot\}$ stacks its vector arguments on the top of each other to generate a connected vector. The notation diag$\{a,b\}$ denotes a diagonal matrix with entries a and b. Likewise, the notation diag$\{\boldsymbol{A},\boldsymbol{B}\}$ denotes a block diagonal matrix with block entries $\boldsymbol{A}$ and $\boldsymbol{B}$. The other symbols will be defined in the context where they are used.

II. Compressed diffusion LMS

We shall now introduce and analyze the stochastic behavior of the compressed diffusion LMS (CD) as an intermediate step towards the doubly-compressed diffusion LMS (DCD). For the sake of simplicity, we shall consider that $\boldsymbol{C}$ is doubly stochastic. We shall set $\boldsymbol{A}$ to the identity matrix $\boldsymbol{I}_N$, with the purpose of limiting the network load. The compressed diffusion LMS algorithm is defined as:

$$\begin{aligned}\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\boldsymbol{u}_{\ell,i}\big[d_\ell(i) - \boldsymbol{u}_{\ell,i}^\top(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} \\ + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}_{\ell,i-1})\big]\end{aligned} \tag{7}$$

where $\boldsymbol{H}_{\ell,i} = \text{diag}\{\boldsymbol{h}_{\ell,i}\}$. We shall assume that $\boldsymbol{h}_{\ell,i}$ is a $L\times 1$ binary vector, generated by randomly setting M of its L entries to 1, and the other $L-M$ entries to 0. We shall also assume that all possible outcomes for $\boldsymbol{h}_{\ell,i}$ are equally likely, and i.i.d. over time and space. This leads to:

$$\mathbb{E}\{\boldsymbol{H}_{\ell,i}\} = \tfrac{M}{L}\boldsymbol{I}_L \tag{8}$$

and, given any $L\times L$ matrix $\boldsymbol{\Sigma}$,

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{H}_{\ell,i}\boldsymbol{\Sigma}\boldsymbol{H}_{k,i}\} = \\ &\begin{cases}\frac{M}{L}\left(\left(1-\frac{M-1}{L-1}\right)\boldsymbol{I}_L\odot\boldsymbol{\Sigma} + \frac{M-1}{L-1}\boldsymbol{\Sigma}\right) & \text{if } \ell = k\\ \left(\frac{M}{L}\right)^2\boldsymbol{\Sigma} & \text{otherwise}\end{cases}\end{aligned} \tag{9}$$

where $\odot$ denotes the Hadamard entrywise product. Due to the constraint $\boldsymbol{1}_L^\top\boldsymbol{h}_{\ell,i} = M$, node ℓ receives exactly M entries from its neighbors at each time instant i. The missing $(L-M)$ entries are filled in with estimates that are available at node ℓ.

Assumption 1 The regression vectors $\boldsymbol{u}_{k,i}$ arise from a zero-mean random process that is temporally white and spatially independent. A direct consequence of this assumption is that $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{w}_{\ell,j}$ for all ℓ and $j<i$.

Assumption 2 The matrices $\boldsymbol{H}_{i,k}$ arise from a random process that is temporally white, spatially independent, and independent of any other process.

We introduce the $L\times 1$ error vectors:

$$\widetilde{\boldsymbol{w}}_{k,i} = \boldsymbol{w}^o - \boldsymbol{w}_{k,i} \tag{10}$$

and we collect them from across all nodes into the vectors:

$$\widetilde{\boldsymbol{w}}_i = \text{col}\{\widetilde{\boldsymbol{w}}_{1,i}, \widetilde{\boldsymbol{w}}_{2,i}, \ldots, \widetilde{\boldsymbol{w}}_{N,i}\} \tag{11}$$

Let $\boldsymbol{R}_{u_\ell,i} = \boldsymbol{u}_{\ell,i}\boldsymbol{u}_{\ell,i}^\top$. We also introduce:

$$\boldsymbol{\mathcal{M}} = \text{diag}\{\mu_1\boldsymbol{I}_L, \mu_2\boldsymbol{I}_L, \ldots, \mu_N\boldsymbol{I}_L\} \tag{12}$$

$$\boldsymbol{\mathcal{R}}_i = \text{diag}\left\{\sum_{\ell\in\mathcal{N}_1} c_{\ell,1}\boldsymbol{R}_{u_\ell,i}, \ldots, \sum_{\ell\in\mathcal{N}_N} c_{\ell,N}\boldsymbol{R}_{u_\ell,i}\right\} \tag{13}$$

$$\boldsymbol{\mathcal{R}}_{u,i} = \text{diag}\{\boldsymbol{R}_{u_1,i}, \boldsymbol{R}_{u_2,i}, \ldots, \boldsymbol{R}_{u_N,i}\} \tag{14}$$

$$\boldsymbol{\mathcal{H}}_i = \text{diag}\{\boldsymbol{H}_{1,i}, \boldsymbol{H}_{2,i}, \ldots, \boldsymbol{H}_{N,i}\} \tag{15}$$

$$\boldsymbol{\mathcal{C}} = \boldsymbol{C}\otimes\boldsymbol{I}_L \tag{16}$$

where $\otimes$ denotes the Kronecker product. Finally, we introduce the $N\times N$ block matrix $\boldsymbol{\mathcal{R}}_{m,i}$ with each block defined as:

$$[\boldsymbol{\mathcal{R}}_{I-H,i}]_{k\ell} = c_{\ell k}\boldsymbol{R}_{u_\ell,i}(\boldsymbol{I}_L - \boldsymbol{H}_{k,i}) \tag{17}$$

Using recursion (7) and definitions (10), (11), we write:

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} - \boldsymbol{\mathcal{G}}\boldsymbol{s}_i \tag{18}$$

where

$$\boldsymbol{\mathcal{B}}_i = \boldsymbol{I}_{NL} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{H}}_i - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{I-H,i} \tag{19}$$

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \tag{20}$$

$$\boldsymbol{s}_i = \text{col}\{\boldsymbol{u}_{1,i}v_1(i), \boldsymbol{u}_{2,i}v_2(i), \ldots, \boldsymbol{u}_{N,i}v_N(i)\} \tag{21}$$

A. Mean convergence

Taking expectations of both sides of recursion (18), using Assumptions 1 and 2, and $\mathbb{E}\{\boldsymbol{s}_i\} = 0$, we find:

$$\begin{aligned}&\mathbb{E}\{\widetilde{\boldsymbol{w}}_i\}\\ &= [\boldsymbol{I}_{NL} - \mathbb{E}\{\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{H}}_i\} - \mathbb{E}\{\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{I-H,i}\}]\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\}\\ &\quad - \mathbb{E}\{\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{s}_i\}\\ &= \left[\boldsymbol{I}_{NL} - \tfrac{M}{L}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \left(1-\tfrac{M}{L}\right)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u\right]\mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\}\end{aligned} \tag{22}$$

where

$$\boldsymbol{\mathcal{R}} = \mathbb{E}\{\boldsymbol{\mathcal{R}}_i\} = \text{diag}\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_N\} \tag{23}$$

$$\boldsymbol{\mathcal{R}}_u = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{u,i}\} = \text{diag}\{\boldsymbol{R}_{u_1}, \boldsymbol{R}_{u_2}, \ldots, \boldsymbol{R}_{u_N}\} \tag{24}$$

with

$$\boldsymbol{R}_k = \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \boldsymbol{R}_{u_\ell} \tag{25}$$

From (22), we observe that the algorithm (7) asymptotically converges in the mean toward $\boldsymbol{w}^o$ if, and only if,

$$\rho\Big(\boldsymbol{I}_{NL} - \tfrac{M}{L}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \big(1 - \tfrac{M}{L}\big)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u\Big) < 1 \tag{26}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. We know that $\rho(\boldsymbol{X}) < \|\boldsymbol{X}\|$ for any induced norm. Then:

$$\begin{aligned}
&\rho\Big(\boldsymbol{I}_{NL} - \tfrac{M}{L}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \big(1 - \tfrac{M}{L}\big)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u\Big) \\
&\le \|\boldsymbol{I}_{NL} - \tfrac{M}{L}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \big(1 - \tfrac{M}{L}\big)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u\|_{b,\infty} \\
&\le \max_{\ell,k} \|\boldsymbol{I}_L - \tfrac{M}{L}\mu_k \boldsymbol{R}_k - \big(1 - \tfrac{M}{L}\big)\mu_k c_{\ell k}\boldsymbol{R}_{u_\ell}\|
\end{aligned} \tag{27}$$

where $\|\cdot\|_{b,\infty}$ denotes the block maximum norm. This leads us to the following condition:

$$\mu_k < \frac{2}{\max\limits_{\ell \in \mathcal{N}_k} \lambda_{\max}\left(\tfrac{M}{L}\boldsymbol{R}_k + \big(1 - \tfrac{M}{L}\big)c_{\ell k}\boldsymbol{R}_{u_\ell}\right)} \tag{28}$$

where $\lambda_{\max}(\cdot)$ stands for the maximum eigenvalue of its matrix argument. Furthermore, by Weyl's theorem we have:

$$\mu_k < \frac{2}{\tfrac{M}{L}\lambda_{\max}(\boldsymbol{R}_k) + \big(1 - \tfrac{M}{L}\big)\max\limits_{\ell \in \mathcal{N}_k}[c_{\ell k}\lambda_{\max}(\boldsymbol{R}_{u_\ell})]} \tag{29}$$

since $\boldsymbol{R}_k$ and $\boldsymbol{R}_{u_\ell}$ are Hermitian matrices.

B. Mean-square stability

We shall now analyze the mean-square deviation $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\Sigma}}\}$ where $\boldsymbol{\Sigma}$ denotes a nonnegative $N \times N$ block diagonal matrix. The choice of matrix $\boldsymbol{\Sigma}$ determines the type of information extracted from the network and nodes. Using the independence assumption and (18), we find:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\Sigma}} = \mathbb{E}\{\widetilde{\boldsymbol{w}}^\top_{i-1}\boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\} + \mathbb{E}\{\boldsymbol{s}_i^\top \boldsymbol{\mathcal{G}}^\top \boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}\boldsymbol{s}_i\} \tag{30}$$

On the one hand, note that the second term on the RHS of (30) can be written as:

$$\mathbb{E}\{\boldsymbol{s}_i^\top \boldsymbol{\mathcal{G}}^\top \boldsymbol{\Sigma}\boldsymbol{\mathcal{G}}\boldsymbol{s}_i\} = \text{trace}\{\boldsymbol{\mathcal{G}}\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{G}}^\top \boldsymbol{\Sigma}\} \tag{31}$$

where

$$\boldsymbol{\mathcal{S}} = \text{diag}(\sigma^2_{v,1}\boldsymbol{R}_{u,1}, \ldots, \sigma^2_{v,N}\boldsymbol{R}_{u,N}) \tag{32}$$

On the other hand, the first term on the RHS of (30) can be expressed as:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}^\top_{i-1}\boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\} = \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'} \tag{33}$$

where the weighting matrix $\boldsymbol{\Sigma}'$ is defined as

$$\begin{aligned}
\boldsymbol{\Sigma}' &= \mathbb{E}\{\boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma}\,\boldsymbol{\mathcal{B}}_i\} \\
&= \boldsymbol{\Sigma} - \tfrac{M}{L}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{M}}\,\boldsymbol{\mathcal{R}} - \big(1 - \tfrac{M}{L}\big)\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u \\
&\quad - \tfrac{M}{L}\boldsymbol{\mathcal{R}}^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma} - \big(1 - \tfrac{M}{L}\big)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma} \\
&\quad + \boldsymbol{P}_1 + \boldsymbol{P}_2 + \boldsymbol{P}_2^\top + \boldsymbol{P}_3
\end{aligned} \tag{34}$$

with

$$\boldsymbol{P}_1 = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_i^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{H}}_i\} \tag{35}$$

$$\boldsymbol{P}_2 = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_i^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{I-H,i}\} \tag{36}$$

$$\boldsymbol{P}_3 = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{I-H,i}^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{I-H,i}\} \tag{37}$$

To evaluate $\boldsymbol{P}_1$, we write:

$$\boldsymbol{P}_1 = \mathbb{E}_{\mathcal{H}}\Big\{\boldsymbol{\mathcal{H}}_i\,\mathbb{E}_{\mathcal{R}}\{\boldsymbol{\mathcal{R}}_i^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_i\}\,\boldsymbol{\mathcal{H}}_i\Big\} \tag{38}$$

Consider first $\mathbb{E}_{\mathcal{H}}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{H}}_i\}$ where $\boldsymbol{\Pi}$ denotes any $NL \times NL$ deterministic matrix. It can be shown that:

$$\begin{aligned}
&\mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{H}}_i\} \\
&\quad = \alpha_1\,(\boldsymbol{I}_N \otimes \boldsymbol{1}_{LL}) \odot \boldsymbol{\Pi} + \alpha_2\,\boldsymbol{I}_{NL} \odot \boldsymbol{\Pi} + \alpha_3\,\boldsymbol{\Pi}
\end{aligned} \tag{39}$$

where $\boldsymbol{1}_{LL}$ denotes an all-one $L \times L$ matrix, and

$$\alpha_1 = \tfrac{M}{L}\left(\tfrac{M-1}{L-1} - \tfrac{M}{L}\right)\ \ \alpha_2 = \tfrac{M}{L}\left(1 - \tfrac{M-1}{L-1}\right)\ \ \alpha_3 = \big(\tfrac{M}{L}\big)^2$$

Next, consider the inner expectation in the RHS of (38). The evaluation of this expectation depends on higher-order moments of the regression data. While we can continue with the analysis by calculating these terms, it is sufficient for the exposition to focus on the case of sufficiently small step-sizes where a reasonable approximation is [1]:

$$\mathbb{E}\{\boldsymbol{\mathcal{R}}_i^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_i\} = \boldsymbol{\mathcal{R}}^\top \boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} \tag{40}$$

Substituting (39) and (40) into (38) leads to $\boldsymbol{P}_1$. Consider $\boldsymbol{P}_2$. Using the same higher-order approximation as above, we have:

$$\begin{aligned}
[\boldsymbol{P}_2]_{k\ell} &= \tfrac{M}{L}\boldsymbol{R}_k[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}c_{\ell k}\boldsymbol{R}_{u_\ell} \\
&\quad - \mathbb{E}\,\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{k,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}c_{\ell k}\boldsymbol{R}_{u_\ell,i}\boldsymbol{H}_{k,i}\}
\end{aligned} \tag{41}$$

The second term in RHS of (41) is of the form:

$$[\varphi(\boldsymbol{\Pi})]_{k\ell} = \mathbb{E}\{\boldsymbol{H}_{k,i}[\boldsymbol{\Pi}]_{k\ell}\boldsymbol{H}_{k,i}\} \tag{42}$$

with $[\boldsymbol{\Pi}]_{k\ell} = \boldsymbol{R}_{k,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}c_{\ell k}\boldsymbol{R}_{u_\ell,i}$. It can be shown that:

$$\varphi(\boldsymbol{\Pi}) = \alpha_2\,(\boldsymbol{1}_{NN} \otimes \boldsymbol{I}_L) \odot \boldsymbol{\Pi} + (\alpha_1 + \alpha_3)\,\boldsymbol{\Pi} \tag{43}$$

This leads to:

$$\boldsymbol{P}_2 = \tfrac{M}{L}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u - \varphi(\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u) \tag{44}$$

Finally, using the same higher-order approximation as above with $\boldsymbol{P}_3$ leads to:

$$[\boldsymbol{P}_3]_{k\ell} = \sum_{m=1}^{N} \mathbb{E}\{[\boldsymbol{\mathcal{R}}_{I-H,i}]^\top_{km}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{mm}[\boldsymbol{\mathcal{R}}_{I-H,i}]_{m\ell}\} \tag{45}$$

and

$$\begin{aligned}
\boldsymbol{P}_3 &= \big(1 - 2\tfrac{M}{L}\big)\,\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u \\
&\quad + \varphi(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top \boldsymbol{\mathcal{R}}_u)
\end{aligned} \tag{46}$$

Following the same reasoning as in [1] allows to express matrix $\boldsymbol{\Sigma}'$ in a vector form as

$$\boldsymbol{\sigma}' = \boldsymbol{\mathcal{F}}\boldsymbol{\sigma} \tag{47}$$

where $\boldsymbol{\sigma} = \text{vec}(\boldsymbol{\Sigma})$, $\boldsymbol{\sigma}' = \text{vec}(\boldsymbol{\Sigma}')$ and:

$$\begin{aligned}\boldsymbol{\mathcal{F}} &= \boldsymbol{I}_{(NM)^2} - \tfrac{M}{L}(\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}} \otimes \boldsymbol{I}_{NL}) \\ &- (1-\tfrac{M}{L})(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}} \otimes \boldsymbol{I}_{NL}) - \tfrac{M}{L}(\boldsymbol{I}_{NL} \otimes \boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}) \\ &- (1-\tfrac{M}{L})(\boldsymbol{I}_{NL} \otimes \boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}) + \boldsymbol{Z}_1 + \boldsymbol{Z}_2 + \boldsymbol{Z}_{2^\top} + \boldsymbol{Z}_4\end{aligned} \tag{48}$$

where matrices $\boldsymbol{Z}_j$ are obtained by applying the $\text{vec}(\cdot)$ operator on $\boldsymbol{P}_j$ and using the following property:

$$\text{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^\top \otimes \boldsymbol{A})\text{vec}(\boldsymbol{B}) \tag{49}$$

The analytical expressions of $\boldsymbol{Z}_i$ are not provided in this paper due to the lake of space. They will be made available in an extended version of this paper. Substituting (31) and (33) in (30), and applying the vec operation to both sides, we get:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\sigma}} = \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}} + [\text{vec}(\boldsymbol{\mathcal{Y}}^\top)]^\top\boldsymbol{\sigma} \tag{50}$$

where

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{G}}\,\boldsymbol{\mathcal{S}}\,\boldsymbol{\mathcal{G}}^\top \tag{51}$$

III. Doubly-compressed diffusion LMS

The compressed diffusion LMS studied before only partially transmit local estimates while the estimated gradient vectors are fully transmitted through the network. We shall now introduce the doubly-compressed diffusion LMS algorithm, which offers an additional compression layer by transmitting partial estimated gradient vectors. Node k receives M_∇ entries of $\hat{\nabla}_w J_k(\boldsymbol{w}_{\ell,i-1})$ from its neighbors at each time instant i. The missing $(L - M_\nabla)$ entries are filled in with the gradient estimates that are available at node k. The algorithm can be formulated as:

$$\begin{aligned}\boldsymbol{w}_{k,i} &= \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\boldsymbol{Q}_{\ell,i}\boldsymbol{u}_{\ell,i}\big[d_\ell(i) \\ &- \boldsymbol{u}^\top_{\ell,i}(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\,\boldsymbol{w}_{\ell,i-1})\big] \\ &+ \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\boldsymbol{u}_{k,i}\big[d_k(i) \\ &- \boldsymbol{u}^\top_{k,i}(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\,\boldsymbol{w}_{\ell,i-1})\big]\end{aligned} \tag{52}$$

where $\boldsymbol{Q}_{\ell,i}$ is a stochastic selection matrix that has the same properties as $\boldsymbol{H}_{k,i}$ defined in the previous section.

Assumption 3 The matrices $\boldsymbol{Q}_{i,\ell}$ arise from a random process that is temporally white, spatially independent, and independent of any other process.

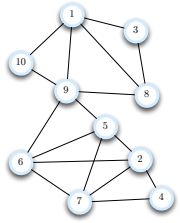
Proceeding as in the previous section, we find that:

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} - \boldsymbol{\mathcal{G}}_i\boldsymbol{s}_i \tag{53}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{B}}_i &= \boldsymbol{I}_{NL} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q,i}\boldsymbol{\mathcal{H}}_i - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i\boldsymbol{\mathcal{R}}_{u,i}\boldsymbol{\mathcal{H}}_i \\ &- \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{(I-Q)(I-H),i}\end{aligned} \tag{54}$$

$$\boldsymbol{\mathcal{G}}_i = \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}^\top + \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i \tag{55}$$



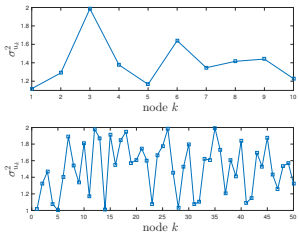


Fig. 1: (left) Network topology. (right) Variance $\sigma^2_{u_k}$ of regressors in Experiment 1 (top) and in Experiment 2 (bottom).

with

$$\boldsymbol{\mathcal{R}}_{Q,i} = \text{diag}\left\{\sum_{\ell\in N_1} c_{\ell 1}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}, \ldots, \sum_{\ell\in N_N} c_{\ell N}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\right\}$$

$$\boldsymbol{\mathcal{Q}}_i = \text{diag}\{\boldsymbol{Q}_{1,i}, \boldsymbol{Q}_{2,i}, \ldots, \boldsymbol{Q}_{N,i}\}$$

$$\boldsymbol{\mathcal{Q}}'_i = \text{diag}\left\{\sum_{\ell\in N_1} c_{\ell 1}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i}), \ldots, \sum_{\ell\in N_N} c_{\ell N}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\right\}$$

$$[\boldsymbol{\mathcal{R}}_{Q(I-H),i}]_{k\ell} = c_{\ell k}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}(\boldsymbol{I}_L - \boldsymbol{H}_{k,i})$$

$$[\boldsymbol{\mathcal{R}}_{(I-Q)(I-H),i}]_{k\ell} = c_{\ell k}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\boldsymbol{R}_{u_k,i}(\boldsymbol{I}_L - \boldsymbol{H}_{k,i})$$

The performance analysis of this algorithm is similar to the analysis of compressed diffusion LMS, although more tedious. The following result can be obtained following the same steps.

A. Convergence in mean

Taking expectations of both sides of (53), we obtain:

$$\begin{aligned}\mathbb{E}\{\widetilde{\boldsymbol{w}}_i\} = \Big(&\boldsymbol{I}_{NL} - \tfrac{MM_\nabla}{L^2}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \tfrac{M}{L}\left(1-\tfrac{M_\nabla}{L}\right)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u \\ &- \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u \\ &-\left(1-\tfrac{M_\nabla}{L}\right)\left(1-\tfrac{M}{L}\right)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}^\top\Big)\mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\}\end{aligned} \tag{56}$$

From (56), the algorithm is stable as long as the following condition is met:

$$\mu_k < \frac{2}{\lambda_{\max,k}} \tag{57}$$

where

$$\begin{aligned}\lambda_k = &\tfrac{MM_\nabla}{L^2}\lambda_{\max}(\boldsymbol{R}_k) + \tfrac{M}{L}\left(1-\tfrac{M_\nabla}{L}\right)\lambda_{\max}(\boldsymbol{R}_{u_k}) \\ &+ \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\max_{\ell\in\mathcal{N}_k}\left[c_{\ell k}\,\lambda_{\max}(\boldsymbol{R}_{u_\ell})\right] \\ &+ \left(1-\tfrac{M_\nabla}{L}\right)\left(1-\tfrac{M}{L}\right)\max_{\ell\in\mathcal{N}_k}\left[c_{\ell k}\lambda_{\max}(\boldsymbol{R}_{u_k})\right]\end{aligned} \tag{58}$$

Due to lake of space and of the complexity of the analysis in the mean-square sense, this analysis is not reported here. It will be made available in an extended version of this article.

IV. Simulation Results

We shall now evaluate the accuracy of the theoretical models with simulations. We performed two experiments to determine the performance and efficiency of both algorithms. First, we considered a small network in order to validate the theoretical models. Then, we considered a larger network and high dimensional measurements in order to test the algorithms. For both experiments, parameter vectors $\boldsymbol{w}^o$ were generated from

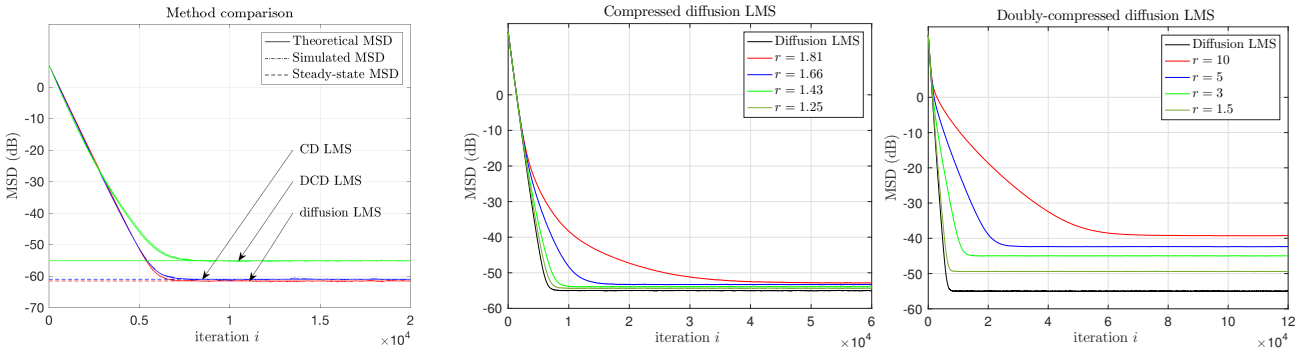


Fig. 2: Theoretical model and simulated curves (left), evolution of the MSD as a function of the compression ratio for compressed diffusion LMS (center), and doubly-compressed diffusion LMS (right).

a zero-mean Gaussian distribution. The input data $\boldsymbol{u}_{k,i}$ were drawn from zero-mean Gaussian distributions with covariance $\boldsymbol{R}_{u,k} = \sigma_{u,k}^2 \boldsymbol{I}_L$ reported in Fig. 1. The weighting matrices $\boldsymbol{C}$ were generated using the Metropolis rule [1]. Noises $v_k(i)$ were zero-mean, i.i.d. and Gaussian distributed with variance $\sigma_{v,k}^2 = 10^{-3}$. The step-sizes were set to $\mu_k = 10^{-3}$. The simulation results were averaged over 100 Monte Carlo runs.

1) Experiment 1: Due to the high dimensionality of the theoretical models, we considered the network of $N = 10$ nodes shown in Fig. 1 (left). We set $L = 5$, $M = 3$ and $M_\nabla = 1$. This resulted in compression ratios of $\frac{10}{8}$ and $\frac{5}{2}$ for the compressed and the doubly-compressed diffusion LMS, respectively. It can be observed in Fig. 2 (left) that the theoretical models fit accurately the simulated results for both algorithms. Unsurprisingly, diffusion LMS outperformed its compressed counterparts at the expense of a higher communication load.

2) Experiment 2: Since compression is relevant for relatively large data flows, we considered a network with $N = 50$ agents. Data dimension was set to $L = 50$. Figure 2 illustrates the performance of the algorithms for different compression ratios. Note that maximum compression ratio that can be reached with the compressed diffusion LMS is $\frac{100}{55}$ since estimated gradient vectors are fully transmitted. Doubly-compressed diffusion LMS is more flexible since the compression ratio can be controlled via M and M_∇.

V. Conclusion

In order to preserve energy resources, we investigated compression techniques for diffusion LMS that consist of exchanging partial local estimates. We carried out an analysis of the stochastic behavior of the proposed algorithms in the mean and mean-square sense. Finally we provided simulation results to illustrate the accuracy of the theoretical models and the performance of the compression layer.

References

[1] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2014. Also available as arXiv:1205.4220 [cs.MA], May 2012., pp. 322–454.

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

[3] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

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

[5] R. Arablouei, S. Werner, K. Doğançay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Processing*, vol. 117, pp. 355–361, 2015.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[8] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[9] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion lms for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE MLSP'14*, Reims, France, 2014, pp. 1–6.

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

[11] ——, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[12] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 11, pp. 2835–2850, 2016.

[13] ——, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6329–6344, 2016.

[14] W. Gao, J. Chen, C. Richard, and J. Huang, "Diffusion adaptation over networks with kernel least-mean-square," in *Proc. IEEE CAMSAP'15*, Cancún, Mexico, 2015.

[15] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE CAMSAP'13*, Saint Martin, French West Indies, 2013.

[16] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5308–5323, 2014.

[17] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.

[18] R. Arablouei, K. Dogancay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3510–3522, 2014.

[19] V. Vadidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Partial-diffusion least mean-square estimation over networks under noisy information exchange," *arXiv preprint arXiv:1511.09044*, 2015.