Optimal Combination Weight for Sparse Diffusion Least-Mean-Square based on Consensus Propagation

Ayano Nakai-Kasai* and Kazunori Hayashi†
* Kyoto University, Kyoto, Japan
E-mail: nakai.ayano@sys.i.kyoto-u.ac.jp Tel/Fax: +81-75-753-4822
† Kyoto University, Kyoto, Japan
E-mail: hayashi.kazunori.4w@kyoto-u.ac.jp Tel/Fax: +81-75-753-9690

***Abstract*—This paper considers distributed adaptive signal processing for tracking an unknown sparse parameter vector in large-scale networks. We propose a sparsity-promoting diffusion least-mean-square algorithm based on consensus propagation, which is an average consensus algorithm using message passing techniques. The main contributions of the paper are optimizing coefficients in the algorithm in terms of the steady-state error to achieve better convergence and robustness, and presenting the adaptive implementation.**

***Index Terms*—Diffusion LMS, distributed signal processing, average consensus, sparseness**

I. Introduction

Distributed signal processing or distributed optimization is gathering more interest in the context of wireless sensor networks and multi-agent networked systems [1]–[6]. Diffusion least-mean-square (D-LMS) [7]–[11] is one of the most popular algorithms to track and estimate an unknown parameter vector in a fully distributed manner in such networks. Moreover, for the applications where the unknown parameter is known to be sparse, an extension of D-LMS called sparse diffusion LMS (SD-LMS) has been proposed in [12], which considers sparse regularization as in compressed sensing [13]. SD-LMS is superior to the original D-LMS in the case that the unknown vector is sufficiently sparse.

In D-LMS and SD-LMS, each node in the network iterates the updates of estimate by using its own measurement like LMS algorithm [14] and also by averaging the neighbors' estimates. The former update is called LMS step and the latter is averaging step. The averaging step can be regarded as the update for average consensus, which is a well-known problem to obtain the average of all nodes' state values in a distributed manner. In D-LMS and SD-LMS, the average consensus protocol [15] is employed for the averaging step, however, this protocol is known to require a lot of iterations for the convergence especially in large networks.

We have proposed to use a faster average consensus algorithm instead of the average consensus protocol in [15] to achieve faster convergence of D-LMS and SD-LMS. In our previous work [11] and [16], the proposed methods have shown better convergence performance than D-LMS and SD-LMS, respectively, by employing consensus propagation (CP) [17], which is based on the idea of message passing algorithms [18]. In [11], we have further optimized coefficients involved in the combination weights of the averaging step by minimizing mean-square-deviation (MSD) of D-LMS at the steady-state. On the other hand, in [16], we have employed the coefficients that minimize the steady-state MSD of D-LMS instead of that of SD-LMS because minimizing the latter requires complicated calculations. Thus, the coefficients in [16] have not considered the effect of the sparse regularization.

In this paper, we tackle the optimization of the coefficients of our previous method [16] by minimizing MSD of SD-LMS and considering the effect of the sparse regularization to make the method faster and more robust to the change of the unknown sparse vector. The optimal coefficients are derived under some approximations while the effect of the sparse regularization term into consideration. This is the first work that shows the optimal combination weight for SD-LMS as long as we know while that for D-LMS has been considered in [11], [19]–[21]. Moreover, we derive an adaptive implementation to track the optimal coefficients as well as the unknown parameter. The proposed algorithm requires a slightly higher computational complexity but simulation results show that it achieves a better convergence performance and robustness to the change of the unknown vector than that of conventional methods especially in dense networks.

In the rest of the paper, we use the following notations. Let $\mathbb{R}$ and $\mathbb{C}$ be the set of real and complex numbers, respectively. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote the transpose and the Hermitian transpose, respectively. $\mathrm{E}[\cdot]$ and $\mathrm{Tr}(\cdot)$ stand for the expectation and the trace operators, respectively. $\boldsymbol{I}_M$ is the identity matrix of size $M \times M$. $\|\boldsymbol{w}\|_p$ for a vector $\boldsymbol{w} = [w_1, \ldots, w_M]^{\mathrm{T}} \in \mathbb{C}^M$ and $p \geq 1$ is ℓ_p-norm defined by $\left(\sum_{m=1}^{M} |w_m|^p\right)^{1/p}$. $\|\boldsymbol{w}\|_0$ for a vector $\boldsymbol{w}$ is the number of nonzero elements of $\boldsymbol{w}$ and is called ℓ_0-norm. $\mathrm{diag}\{\cdots\}$ means a diagonal or block diagonal matrix whose diagonal elements or matrices are given by the elements in the braces. $\mathrm{sign}(\boldsymbol{w})$ for $\boldsymbol{w} = [w_1, \ldots, w_M]^{\mathrm{T}} \in \mathbb{R}^M$ is a vector of size M whose m-th element $[\mathrm{sign}(\boldsymbol{w})]_m$ is defined as

$$[\mathrm{sign}(\boldsymbol{w})]_m = \begin{cases} w_m/|w_m|, & \text{if } w_m \neq 0 \\ 0, & \text{if } w_m = 0. \end{cases} \quad (m = 1, \ldots, M)$$

II. Preliminaries

A. System Model

We consider a connected network composed of N sensor nodes. Each node k ($k = 1, 2, \ldots, N$) obtains a noisy measurement $d_k^{(i)} \in \mathbb{C}$ at each time i ($i = 0, 1, 2, \ldots$) as a measurement of an unknown vector of interest $\boldsymbol{w}^{\mathrm{o}} \in \mathbb{C}^M$. This indicates the following linear measurement model as in [7]–[12]

$$d_k^{(i)} = \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}^{\mathrm{o}} + v_k^{(i)}, \tag{1}$$

where $\boldsymbol{u}_k^{(i)} \in \mathbb{C}^M$ is a random measurement vector and $v_k^{(i)} \in \mathbb{C}$ is a zero-mean additive complex white Gaussian noise with variance of σ_k^2. We assume that each node in the network can perform single-hop communication only with its neighboring nodes. The information which each node can obtain is only a part of all nodes' information, i.e., its own and neighbors' information. Moreover, the total number of nodes N is assumed to be unknown to each node. The set of neighbors of node k including k itself is denoted by $\mathcal{N}_k$.

B. Diffusion LMS

D-LMS [7], [8] is one of the distributed adaptive filters to track and estimate unknown vector at all nodes in a fully distributed manner. All nodes in the network aim to obtain the estimate of $\boldsymbol{w}^{\mathrm{o}}$ by minimizing the following global cost function:

$$\mathcal{J}_{\mathrm{dif}}^{\mathrm{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} \mathrm{E}\big[|d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}|^2\big]. \tag{2}$$

Each node cannot directly solve this problem because it includes all nodes' information that is unavailable at the node. Therefore, the following approximated local cost function has been considered at each node k,

$$\mathcal{J}_{\mathrm{dif},k}^{\mathrm{loc}}(\boldsymbol{w}) = \mathrm{E}\big[|d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}|^2\big] + \sum_{l \in \mathcal{N}_k \setminus \{k\}} b_{lk} \|\boldsymbol{w} - \boldsymbol{\phi}_l^{(i)}\|^2, \tag{3}$$

where $\boldsymbol{\phi}_l^{(i)}$ is the current estimate of $\boldsymbol{w}^{\mathrm{o}}$ at node l at time i, and b_{lk} is the weight naturally determined later. The second term of (3) means the penalty for the difference between node k's and neighbors' estimates.

In order to minimize (3), the steepest descent method [22] and LMS algorithm [14] are employed, then node k's estimate $\boldsymbol{\phi}_k^{(i)}$ is updated as

$$\begin{aligned} \boldsymbol{\phi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} &+ \mu_k \boldsymbol{u}_k^{(i)} (d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{\phi}_k^{(i-1)}) \\ &+ \mu_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} b_{lk} (\boldsymbol{\phi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}), \end{aligned} \tag{4}$$

where $\mu_k > 0$ is a step-size parameter. The update by (4) can be divided into 2 steps. The one is to update the current estimate $\boldsymbol{\psi}_k^{(i)}$ by LMS-like rule using its instantaneous measurement. Another is to update the estimate $\boldsymbol{\phi}_k^{(i)}$ by using neighbors' current estimates $\boldsymbol{\psi}_l^{(i)}$. Namely,

$$\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)} (d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{\phi}_k^{(i-1)}), \tag{5}$$

Algorithm 1 Diffusion LMS algorithm

1: Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
2: **for** each time $i \geq 0$ and each node k **do**
3: $\quad \boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)} (d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{\phi}_k^{(i-1)})$
4: $\quad \boldsymbol{\phi}_k^{(i)} = \sum_{l \in \mathcal{N}_k} a_{lk} \boldsymbol{\psi}_l^{(i)}$
5: **end for**

$$\boldsymbol{\phi}_k^{(i)} = \boldsymbol{\psi}_k^{(i)} + \mu_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} b_{lk} (\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\psi}_k^{(i)}), \tag{6}$$

where $\boldsymbol{\phi}_l^{(i)}$ and $\boldsymbol{\phi}_k^{(i-1)}$ in the third term of (4) are replaced with $\boldsymbol{\psi}_l^{(i)}$ and $\boldsymbol{\psi}_k^{(i)}$, respectively. Here, let $\boldsymbol{A} = \{a_{lk}\} \in \mathbb{R}^{N \times N}$ be

$$a_{lk} = \begin{cases} \mu_k b_{lk}, & \text{if } l \in \mathcal{N}_k \setminus \{k\}, \\ 1 - \mu_k \sum_{l \in \mathcal{N}_k \setminus k} b_{lk}, & \text{if } l = k, \\ 0, & \text{if } l \notin \mathcal{N}_k, \end{cases} \tag{7}$$

then (6) can be rewritten as

$$\boldsymbol{\phi}_k^{(i)} = \sum_{l \in \mathcal{N}_k} a_{lk} \boldsymbol{\psi}_l^{(i)}. \tag{8}$$

Summarizing (5) (LMS step) and (8) (averaging step), the updating rules of D-LMS are shown in *Algorithm 1*. Note that we employ in this paper the adaptive-then-combine type of D-LMS proposed in [8].

C. Sparse Diffusion LMS

SD-LMS [12] is an extension of D-LMS for the case when the unknown vector $\boldsymbol{w}^{\mathrm{o}}$ is known to be sparse but the indices of nonzero elements are unknown. All nodes in the network aim to obtain the estimate of $\boldsymbol{w}^{\mathrm{o}}$ by minimizing the following global cost function:

$$\mathcal{J}_{\mathrm{spa}}^{\mathrm{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} \mathrm{E}\big[|d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}|^2\big] + \lambda f(\boldsymbol{w}), \tag{9}$$

where $\lambda > 0$ is a regularization parameter and $f(\boldsymbol{w})$ is a convex sparse regularization function. The specific form of $f(\boldsymbol{w})$ will be discussed later. In the same way as in D-LMS, the alternative problem considered to perform at each node k is minimizing the following approximated local cost function

$$\mathcal{J}_{\mathrm{spa},k}^{\mathrm{loc}}(\boldsymbol{w}) = \mathrm{E}\big[|d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}|^2\big] + \lambda f(\boldsymbol{w}) + \sum_{l \in \mathcal{N}_k \setminus \{k\}} b'_{lk} \|\boldsymbol{w} - \boldsymbol{\phi}_l^{(i)}\|^2, \tag{10}$$

where b'_{lk} is the weight naturally determined later. The LMS-type update for this cost function also can be divided into 2 steps as

$$\begin{aligned} \boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} &+ \mu_k \boldsymbol{u}_k^{(i)} (d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{\phi}_k^{(i-1)}) \\ &- \mu_k \lambda \partial f(\boldsymbol{\phi}_k^{(i-1)}), \end{aligned} \tag{11}$$

$$\boldsymbol{\phi}_k^{(i)} = \boldsymbol{\psi}_k^{(i)} + \mu_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} b'_{lk} (\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\psi}_k^{(i)}), \tag{12}$$

Algorithm 2 Sparse diffusion LMS algorithm

1: Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
2: **for** each time $i \geq 0$ and each node k **do**
3: $\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)}) - \mu_k \lambda \partial f(\boldsymbol{\phi}_k^{(i-1)})$
4: $\boldsymbol{\phi}_k^{(i)} = \sum_{l \in \mathcal{N}_k} a_{lk} \boldsymbol{\psi}_l^{(i)}$
5: **end for**

where $\partial f(\cdot)$ is the sub-gradient of $f(\cdot)$. Letting $b'_{lk} = b_{lk}$ and introducing $\boldsymbol{A} = \{a_{lk}\}$ in (7) lead to

$$\boldsymbol{\phi}_k^{(i)} = \sum_{l \in \mathcal{N}_k} a_{lk} \boldsymbol{\psi}_l^{(i)}. \tag{13}$$

Summarizing (11) (LMS step) and (13) (averaging step), the updating rules of SD-LMS are shown in *Algorithm 2*. It should be noted that SD-LMS is the extension of Algorithm 1 by adding the regularization term to the LMS step.

In this paper, according to [12], we employ the following regularization function

$$f(\boldsymbol{w}) = \sum_{m=1}^{M} \frac{|w_m|}{\epsilon + |w_m|}, \tag{14}$$

where w_m is the m-th element of $\boldsymbol{w}$ and $\epsilon > 0$ is a parameter. While ℓ_1-norm is widely used for the sparse regularization function [23], the function (14) is known to be a better approximation of $\|\boldsymbol{w}\|_0$ than $\|\boldsymbol{w}\|_1$ as long as ϵ is sufficiently small. One of the sub-gradients of (14) is given by

$$\partial f(\boldsymbol{w}) = \mathrm{diag}\left\{\frac{1}{\epsilon + |w_1|}, \ldots, \frac{1}{\epsilon + |w_m|}\right\} \mathrm{sign}(\boldsymbol{w}). \tag{15}$$

The convergence of SD-LMS in mean and mean-square senses are guaranteed in [12] under reasonable assumptions.

D. Consensus Propagation

CP [17] is the algorithm that achieves average consensus by using the idea of belief propagation [18]. Consider a bidirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ is that of edges, and $|\mathcal{V}| = N$. Assume that each node $k \in \mathcal{V}$ has an initial state value x_k. Each node aims at obtaining the average of all nodes' initial values, $\bar{x} = \frac{1}{N}\sum_{u=1}^{N} x_u$, that is, achieving average consensus, by using message propagation algorithm. Two types of updating rules of CP are as follows:

$$K_{(u\to k)}^{[j]} = \frac{1 + \sum_{m \in \mathcal{N}_u \setminus \{k,u\}} K_{(m\to u)}^{[j-1]}}{1 + \frac{1}{\beta_k}\left(1 + \sum_{m \in \mathcal{N}_u \setminus \{k,u\}} K_{(m\to u)}^{[j-1]}\right)}, \tag{16}$$

$$\theta_{(u\to k)}^{[j]} = \frac{x_u + \sum_{m \in \mathcal{N}_u \setminus \{k,u\}} K_{(m\to u)}^{[j-1]} \theta_{(m\to u)}^{[j-1]}}{1 + \sum_{m \in \mathcal{N}_u \setminus \{k,u\}} K_{(m\to u)}^{[j-1]}}, \tag{17}$$

where $(u \to k) \in \mathcal{E}$, $K_{(u\to k)}^{[j]}$ and $\theta_{(u\to k)}^{[j]}$ are the messages from node u to k at time j, $K_{(u\to k)}^{[0]} = \theta_{(u\to k)}^{[0]} = 0$, and $\beta_k > 0$ is a parameter. After iterating (16) and (17) T times between all nodes, the estimate of $\bar{x}$ at node k is given as

$$x_k^{[T]} = \frac{x_k + \sum_{u \in \mathcal{N}_k \setminus \{k\}} K_{(u\to k)}^{[T]} \theta_{(u\to k)}^{[T]}}{1 + \sum_{u \in \mathcal{N}_k \setminus \{k\}} K_{(u\to k)}^{[T]}}. \tag{18}$$

CP can achieve exact average consensus when the network has a tree structure and β_ks $(k = 1, \ldots, N)$ are set to be ∞ only with the same number of iterations T as the diameter of the tree. However, for the case of the network with some cycles, the behavior of CP such as the consensus value and the required number of iterations are unknown. Moreover, the coefficient β_k plays an important role in guaranteeing convergence but its optimal value for general networks is also unknown.

E. Sparse Diffusion LMS using Consensus Propagation

In this section, we introduce the improved SD-LMS that we have proposed in [16], which applies CP instead of the conventional average consensus protocol. As mentioned in Sect. II-D, CP achieves exact average consensus with the same number of iterations as the diameter of the network, when the network has a tree structure. On the other hand, when the network has some cycles, the required number of iterations and convergence value remain unknown. We have thus considered applying a special case of CP, where only the first iteration ($T = 1$) is employed, to SD-LMS as in our previous work [11].

We describe the averaging step of SD-LMS (13) by using the messages of CP, $K_{(u\to k)}^{[1]}$ and $\theta_{(u\to k)}^{[1]}$. The former is naturally calculated as

$$K_{(u\to k)}^{[1]} = \frac{\beta_k}{1 + \beta_k}. \tag{19}$$

Substituting the current estimate $\boldsymbol{\psi}_k^{(i)}$ in (11) for the initial value x_k of CP in (17) $(k = 1, 2, \ldots, N)$, then

$$\theta_{(u\to k)}^{[1]} = x_u = \boldsymbol{\psi}_u^{(i)}. \tag{20}$$

Moreover, the estimate $\boldsymbol{\phi}_k^{(i)}$ is obtained by the derivation of $x_k^{[1]}$ in (18) instead of (13) and can be calculated as

$$\begin{aligned}
\boldsymbol{\phi}_k^{(i)} &= \frac{x_k + \sum_{u \in \mathcal{N}_k \setminus \{k\}} K_{(u\to k)}^{[1]} \theta_{(u\to k)}^{[1]}}{1 + \sum_{u \in \mathcal{N}_k \setminus \{k\}} K_{(u\to k)}^{[1]}} \\
&= \frac{1 + \beta_k}{1 + |\mathcal{N}_k|\beta_k} \boldsymbol{\psi}_k^{(i)} + \frac{\beta_k}{1 + |\mathcal{N}_k|\beta_k} \sum_{u \in \mathcal{N}_k \setminus \{k\}} \boldsymbol{\psi}_u^{(i)} \\
&= \sum_{l \in \mathcal{N}_k} a_{lk}^{\mathrm{cp}} \boldsymbol{\psi}_l^{(i)},
\end{aligned}$$

where

$$a_{lk}^{\mathrm{cp}} = \begin{cases} \frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}, & \text{if } l \in \mathcal{N}_k \setminus \{k\}, \\ \frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}, & \text{if } l = k, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

This can be considered as a combination weight for SD-LMS, which satisfies the conditions in (7). Therefore, this method arrives at an SD-LMS that uses the coefficient β_k for the combination weights.

The optimal choice of β_k in terms of CP is also unknown as mentioned in Sect. II-D. In [16], we have focused on the average of all nodes' MSD at the steady-state, named steady-state network MSD defined as

$$\mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}} = \lim_{i\to\infty} \frac{1}{N}\sum_{k=1}^{N} \mathrm{E}[\|\boldsymbol{\phi}_k^{(i)} - \boldsymbol{w}^{\mathrm{o}}\|^2],$$

and tried to determine β_k by minimizing $\mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}}$. The steady-state network MSD for any combination weight has been considered in [12] under the following assumptions.

Assumptions : The noise process $\{v_k^{(i)}\}$ is temporally white and spatially independent. The measurement vector process $\{\boldsymbol{u}_k^{(i)}\}$ is temporally white and spatially independent. $v_k^{(i)}$ is independent of $\boldsymbol{u}_l^{(j)}$ for all $l \neq k$ and $i \neq j$. The step-sizes $\{\mu_k\}$ are sufficiently small.

By using the assumptions, the steady-state network MSD of SD-LMS is given by

$$\mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}} = \mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}} + \frac{\lambda}{N} g_{\lambda,\epsilon}(\boldsymbol{A}), \tag{22}$$

where $\mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}}$ is the steady-state network MSD of D-LMS [8] and $g_{\lambda,\epsilon}(\boldsymbol{A})$ is a function which depends on the parameters λ and ϵ, and the combination matrix $\boldsymbol{A}$. The second term of (22) indicates the influence of the sparse regularization term. The specific formula of $g_{\lambda,\epsilon}(\boldsymbol{A})$ is considerably complex and will be discussed later. In [16], for simplicity, we have approximated $g_{\lambda,\epsilon}(\boldsymbol{A})$ as $g_{\lambda,\epsilon}(\boldsymbol{I_N})$ then minimizing $\mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}}$ in terms of β_k can be replaced with minimizing $\mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}}$. Thus, we can use previous results in [11] and derive the optimal value of β_k as

$$\beta_k^{\mathrm{opt}} = \begin{cases} \frac{(|\mathcal{N}_k|-1)\gamma_k^2}{\tilde{\Gamma}_k} & \text{if } \tilde{\Gamma}_k > 0 \\ +\infty & \text{otherwise,} \end{cases} \tag{23}$$

where $\gamma_k^2 = \mu_k^2\sigma_k^2\mathrm{Tr}(\boldsymbol{R}_{u_k})$ and $\tilde{\Gamma}_k = \sum_{l\in\mathcal{N}_k}\gamma_l^2 - |\mathcal{N}_k|\gamma_k^2 \neq 0$.

Furthermore, we have also derived an adaptive solution to estimate γ_k^2 and then update β_k to avoid the direct calculation of γ_k^2 that depends on locally unavailable network statistics such as noise variance σ_k^2 and the correlation matrix $\boldsymbol{R}_{u_k}$. The estimate $\beta_k^{(i)}$ of β_k^{opt} and the corresponding combination weight $a_{lk}^{\mathrm{cp},(i)}$ at time i are described as

$$\beta_k^{(i)} = \begin{cases} \frac{(|\mathcal{N}_k|-1)\tilde{\gamma}_{kk}^{2,(i)}}{\tilde{\Gamma}_k^{(i)}} & \text{if } \tilde{\Gamma}_k^{(i)} > 0 \\ +\infty & \text{otherwise,} \end{cases} \tag{24}$$

$$a_{lk}^{\mathrm{cp},(i)} = \begin{cases} \frac{\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}} & \text{if } l \in \mathcal{N}_k \text{ and } l \neq k \\ \frac{1+\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}} & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases} \tag{25}$$

where $\tilde{\Gamma}_k^{(i)} = \sum_{l\in\mathcal{N}_k}\tilde{\gamma}_{lk}^{2,(i)} - |\mathcal{N}_k|\tilde{\gamma}_{kk}^{2,(i)}$ and $\tilde{\gamma}_{lk}^{2,(i)}$ is the estimate of γ_l^2 at node k and time i and it is updated by $\tilde{\gamma}_{lk}^{2,(i)} = (1-\tilde{\nu}_k)\tilde{\gamma}_{lk}^{2,(i-1)} + \tilde{\nu}_k\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\|^2$, where $\tilde{\nu}_k$ $(0 < \tilde{\nu}_k < 1)$ is the forgetting factor. This combination weight has been named as *adaptive CP rule*. SD-LMS using the adaptive CP rule is summarized in *Algorithm 3*.

Algorithm 3 Sparse diffusion LMS with adaptive CP rule

Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$, $\gamma_{lk}^{2,(-1)}$
for each time $i \geq 0$ and each node k **do**
  $\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k\boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)}) - \mu_k\lambda\partial f(\boldsymbol{\phi}_k^{(i-1)})$
  $\gamma_{lk}^{2,(i)} = (1-\nu_k)\gamma_{lk}^{2,(i-1)} + \nu_k\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\|^2$
  if $\sum_{l\in\mathcal{N}_k}\gamma_{lk}^{2,(i)} - \gamma_{kk}^{2,(i)}|\mathcal{N}_k| > 0$ **then**
    $\beta_k^{(i)} = \frac{(|\mathcal{N}_k|-1)\gamma_{kk}^{2,(i)}}{\sum_{l\in\mathcal{N}_k}\gamma_{lk}^{2,(i)} - \gamma_{kk}^{2,(i)}|\mathcal{N}_k|}$
  else
    $\beta_k^{(i)} = +\infty$ (large positive constant)
  end if
  $a_{lk}^{(i)} = \frac{\beta_k^{(i)}}{1+\beta_k^{(i)}|\mathcal{N}_k|}$ $(l \in \mathcal{N}_k \setminus k)$, $\frac{1+\beta_k^{(i)}}{1+\beta_k^{(i)}|\mathcal{N}_k|}$ $(l = k)$
  $\boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}^{(i)}\boldsymbol{\psi}_l^{(i)}$
end for

III. Proposed Optimal Combination Weight

A. Optimization of Coefficients

The coefficient proposed in [16] has some room for improvement because it has been derived by ignoring the influence of the sparse regularization term. In this paper, we optimize the coefficient β_k by considering the contribution of the second term of (22). The specific formula of $g_{\lambda,\epsilon}(\boldsymbol{A})$ has been shown in [12] as

$$g_{\lambda,\epsilon}(\mathbf{A}) = \lambda g_{1,\boldsymbol{\Sigma},\infty} - g_{2,\boldsymbol{\Sigma},\infty}, \tag{26}$$

where

$$g_{1,\Sigma,\infty} = \lim_{i\to\infty} \mathrm{E}\left[\partial f(\boldsymbol{\phi}^{i-1})^{\mathrm{T}}\mathcal{M}\mathcal{A}\boldsymbol{\Sigma}\mathcal{A}^{\mathrm{T}}\mathcal{M}\partial f(\boldsymbol{\phi}^{i-1})\right], \tag{27}$$

$$g_{2,\boldsymbol{\Sigma},\infty} = \lim_{i\to\infty} -2\mathrm{E}\Big[(\partial f(\boldsymbol{\phi}^{i-1}))^{\mathrm{T}}\mathcal{M}\mathcal{A}\boldsymbol{\Sigma}\mathcal{A}^{\mathrm{T}}(\boldsymbol{I}_{NM} - \mathcal{M}\mathcal{D})\tilde{\boldsymbol{\phi}}^{i-1}\Big], \tag{28}$$

$$\boldsymbol{\phi}^{i-1} = \begin{bmatrix} \boldsymbol{\phi}_1^{(i-1)} \\ \vdots \\ \boldsymbol{\phi}_N^{(i-1)} \end{bmatrix}, \quad \tilde{\boldsymbol{\phi}}^{i-1} = \begin{bmatrix} \boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_1^{(i-1)} \\ \vdots \\ \boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_N^{(i-1)} \end{bmatrix},$$

$$\mathcal{M} = \mathrm{diag}\{\mu_1\boldsymbol{I}_M, \ldots, \mu_N\boldsymbol{I}_M\},$$

$$\mathcal{A} = \boldsymbol{A}\otimes\boldsymbol{I}_M,$$

$$\mathcal{D} = \mathrm{diag}\left\{\sum_{l=1}^{N}\boldsymbol{R}_{u_l}, \ldots, \sum_{l=1}^{N}\boldsymbol{R}_{u_l}\right\},$$

and $\boldsymbol{\Sigma}$ is any Hermitian nonnegative-definite matrix.

The first term of (22), $\mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}}$, is difficult to calculate directly but its upper bound has been already derived in [8] as

$$\mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}} \leq c\sum_{k=1}^{N}\sum_{l\in\mathcal{N}_k}\gamma_l^2 a_{lk}^2, \tag{29}$$

where c is some constant. By substituting (21), (29) can be rewritten as

$$\mathrm{MSD}_{\mathrm{dif}}^{\mathrm{net}} \leq \sum_{k=1}^{N}\left(c\gamma_k^2\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2 + \sum_{l\in\mathcal{N}_k\setminus\{k\}} c\gamma_l^2\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2\right). \tag{30}$$

The second term of (22) is also difficult to directly obtain because it includes the limitation, the expectation, and the unknown vector. Thus, we consider to simplify (27) and (28). First, we approximate $\boldsymbol{\Sigma}$ with the identity matrix. Second, since the step-sizes are sufficiently small, we ignore the quadratic term of (28) with respect to $\mathcal{M}$ and approximate it as

$$g_{2,\boldsymbol{I},\infty} \simeq \lim_{i\to\infty} -2\mathrm{E}\left[(\partial f(\boldsymbol{\phi}^{i-1}))^{\mathrm{T}}\mathcal{M}\mathcal{A}\mathcal{A}^{\mathrm{T}}\tilde{\boldsymbol{\phi}}^{i-1}\right]. \tag{31}$$

We further approximate by removing the limitation and the expectation in (27) and (31), and use the instantaneous values. Specifically, the approximated values of $g_{1,\boldsymbol{I},\infty}$ and $g_{2,\boldsymbol{I},\infty}$ are given by

$$g_{1,\boldsymbol{I},i} \simeq \partial f(\boldsymbol{\phi}^{i-1})^{\mathrm{T}}\mathcal{M}\mathcal{A}\mathcal{A}^{\mathrm{T}}\mathcal{M}\partial f(\boldsymbol{\phi}^{i-1}), \tag{32}$$

$$g_{2,\boldsymbol{I},i} \simeq -2\left[(\partial f(\boldsymbol{\phi}^{i-1}))^{\mathrm{T}}\mathcal{M}\mathcal{A}\mathcal{A}^{\mathrm{T}}\tilde{\boldsymbol{\phi}}^{i-1}\right], \tag{33}$$

respectively. By using these terms, we have the approximated $g_{\lambda,\boldsymbol{\epsilon}}(\boldsymbol{A})$ as

$$\begin{aligned} g_{\lambda,\boldsymbol{\epsilon}}(\boldsymbol{A}) &\approx \lambda g_{1,\boldsymbol{I},i} - g_{2,\boldsymbol{I},i} \\ &= \sum_{k=1}^{N}\sum_{j\in\mathcal{N}_k}\sum_{l\in\mathcal{N}_k}\Big[\lambda\mu_j\mu_l a_{jk}a_{lk}\partial f(\boldsymbol{\phi}_j^{(i-1)})^{\mathrm{T}}\partial f(\boldsymbol{\phi}_l^{(i-1)}) \\ &\quad + 2\mu_j a_{jk}a_{lk}\partial f^{\mathrm{T}}(\boldsymbol{\phi}_j^{(i-1)})\left(\boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_l^{(i-1)}\right)\Big]. \end{aligned} \tag{34}$$

It seems hard to optimize N^2 combination weights $\{a_{lk}\}$ directly from (34) but it can be captured by the optimization of N coefficients $\{\beta_k\}$ in the case of our proposed method. Substituting (21) into (34) leads to

$$\begin{aligned} g_{\lambda,\boldsymbol{\epsilon}}(\boldsymbol{A}) \approx \sum_{k=1}^{N}\Big[&(\tilde{\eta}_{kl}+\tilde{\eta}_{jk})\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right) \\ &+ \tilde{\eta}_{kk}\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2 + \tilde{\eta}_{jl}\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2\Big], \end{aligned} \tag{35}$$

where

$$\tilde{\eta}_{kl} = \sum_{l\in\mathcal{N}_k\setminus\{k\}}\mu_k\partial f(\boldsymbol{\phi}_k^{(i-1)})^{\mathrm{T}}\left\{\lambda\mu_l\partial f(\boldsymbol{\phi}_l^{(i-1)}) + 2\left(\boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_l^{(i-1)}\right)\right\},$$

$$\tilde{\eta}_{jk} = \sum_{j\in\mathcal{N}_k\setminus\{k\}}\mu_j\partial f(\boldsymbol{\phi}_j^{(i-1)})^{\mathrm{T}}\left\{\lambda\mu_k\partial f(\boldsymbol{\phi}_k^{(i-1)}) + 2\left(\boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_k^{(i-1)}\right)\right\},$$

$$\tilde{\eta}_{kk} = \mu_k\partial f(\boldsymbol{\phi}_k^{(i-1)})^{\mathrm{T}}\left\{\lambda\mu_k\partial f(\boldsymbol{\phi}_k^{(i-1)}) + 2\left(\boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_k^{(i-1)}\right)\right\},$$

$$\tilde{\eta}_{jl} = \sum_{l\in\mathcal{N}_k\setminus\{k\}}\sum_{j\in\mathcal{N}_k\setminus\{k\}}\mu_j\partial f(\boldsymbol{\phi}_j^{(i-1)})^{\mathrm{T}} \cdot\left\{\lambda\mu_l\partial f(\boldsymbol{\phi}_l^{(i-1)}) + 2\left(\boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_l^{(i-1)}\right)\right\}.$$

Although it still includes the unknown vector, the replacement is considered in the next section.

By incorporating (30) and (35), we can obtain the approximated upper bound of $\mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}}$ as

$$\begin{aligned} \mathrm{MSD}_{\mathrm{spa}}^{\mathrm{net}} &\leq c\sum_{k=1}^{N}\sum_{l\in\mathcal{N}_k}\gamma_l^2 a_{lk}^2 + \frac{\lambda}{N}(\lambda g_{1,\boldsymbol{I},i} - g_{2,\boldsymbol{I},i}) \qquad (36) \\ &= \sum_{k=1}^{N}\Big[\frac{\lambda}{N}(\tilde{\eta}_{kl}+\tilde{\eta}_{jk})\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right) \\ &\quad + \left(c\gamma_k^2 + \frac{\lambda}{N}\tilde{\eta}_{kk}\right)\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2 \\ &\quad + \left(\sum_{l\in\mathcal{N}_k\setminus\{k\}} c\gamma_l^2 + \frac{\lambda}{N}\tilde{\eta}_{jl}\right)\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2\Big] \\ &= \sum_{k=1}^{N}\Big[\eta_{klj}\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right) \\ &\quad + \eta_{kk}\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2 + \eta_{jl}\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2\Big] \\ &= \sum_{k=1}^{N}F_k(\beta_k), \qquad (37) \end{aligned}$$

where $\eta_{kk} = c\gamma_k^2 + \frac{\lambda}{N}\tilde{\eta}_{kk}$, $\eta_{klj} = \frac{\lambda}{N}(\tilde{\eta}_{kl}+\tilde{\eta}_{jk})$, and $\eta_{jl} = \sum_{l\in\mathcal{N}_k\setminus\{k\}} c\gamma_l^2 + \frac{\lambda}{N}\tilde{\eta}_{jl}$. In this paper, we optimize $\{\beta_k\}_{k=1}^N$ by minimizing the approximated upper bound (37), i.e.,

$$\min_{\{\beta_k\}_{k=1}^N}\sum_{k=1}^{N}F_k(\beta_k). \tag{38}$$

We can divide the problem into the following N problems,

$$\beta_k^{\mathrm{opt}} = \arg\min_{\beta_k} F_k(\beta_k) \quad k = 1,\ldots,N. \tag{39}$$

The differential of F_k with respect to β_k can be calculated as

$$\begin{aligned} \frac{\partial F_k}{\partial\beta_k} = \Big[&\left\{2\eta_{jl} - 2(|\mathcal{N}_k|-1)\eta_{kk} - (|\mathcal{N}_k|-2)\eta_{klj}\right\}\beta_k \\ &- \{2(|\mathcal{N}_k|-1)\eta_{kk} - \eta_{klj}\}\Big]\cdot\frac{1}{(1+|\mathcal{N}_k|\beta_k)^3}. \end{aligned} \tag{40}$$

Since the denominator is positive, $\frac{\partial F_k}{\partial\beta_k} = 0$ when

$$\beta_k = \frac{2(|\mathcal{N}_k|-1)\eta_{kk} - \eta_{klj}}{2\eta_{jl} - 2(|\mathcal{N}_k|-1)\eta_{kk} - (|\mathcal{N}_k|-2)\eta_{klj}}. \tag{41}$$

We put $\Gamma_k = 2\eta_{jl} - 2(|\mathcal{N}_k|-1)\eta_{kk} - (|\mathcal{N}_k|-2)\eta_{klj}$ and $\Lambda_k = 2(|\mathcal{N}_k|-1)\eta_{kk} - \eta_{klj}$. Considering $\beta_k > 0$, we can derive the following optimal parameter:

$$\begin{cases} \beta_k^{\mathrm{opt}} = \frac{\Lambda_k}{\Gamma_k}, & \text{if } \Gamma_k > 0 \text{ and } \Lambda_k > 0, \\ \beta_k^{\mathrm{opt}} \to +0, & \text{if } \Gamma_k > 0 \text{ and } \Lambda_k \leq 0, \\ \beta_k^{\mathrm{opt}} \to +\infty, & \text{if } \Gamma_k \leq 0. \end{cases} \tag{42}$$

Note that sufficiently small β_k indicates that the node do not use the neighbors' estimates at the averaging step but only use its own information. When β_k is very large, the resulting combination weight (21) coincides with conventional uniform rule [24].

B. Adaptive Implementation

The optimal parameter (42) includes unavailable information such as the unknown vector $\boldsymbol{w}^{\mathrm{o}}$, noise variance $\{\sigma_k^2\}$, correlation matrices $\{\boldsymbol{R}_{u_k}\}$, and the number of all nodes N. Therefore, in this section, we consider adaptive estimations and replacements of these factors, and finally derive an adaptive algorithm.

First, we consider adaptive estimations of $\gamma_l^2 = \mu_l^2\sigma_l^2\mathrm{Tr}(\boldsymbol{R}_{u_l})$ in a similar way as that of D-LMS [9], [20], [21]. The estimate $\boldsymbol{\phi}_k^{(i)}$ approaches to the unknown vector $\boldsymbol{w}^{\mathrm{o}}$ as the algorithm iterates (11) and (13), and reaches steady-state under the assumption that the step-sizes are sufficiently small. By using (11) and (1), we can rewrite

$$\boldsymbol{\psi}_l^{(i)} \approx \boldsymbol{w}^{\mathrm{o}} + \mu_l \boldsymbol{u}_l^{(i)} v_l^{(i)} - \mu_l \lambda \partial f(\boldsymbol{w}^{\mathrm{o}}).$$

Taking the expectation leads to

$$\mathrm{E}\left[\left\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{w}^{\mathrm{o}} + \mu_l \lambda \partial f(\boldsymbol{w}^{\mathrm{o}})\right\|^2\right] \approx \mu_l^2\sigma_l^2\mathrm{Tr}(\boldsymbol{R}_{u_l}).$$

We substitute the instantaneous values into the expectation and use it to estimate γ_l^2. Let $\gamma_{lk}^{2,(i)}$ be the estimate of γ_l^2 at node k and time i. We adaptively obtain the estimate by employing the following update:

$$\gamma_{lk}^{2,(i)} = (1-\nu_k)\gamma_{lk}^{2,(i-1)} + \nu_k \left\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)} + \mu_l \lambda \partial f(\boldsymbol{\phi}_k^{(i-1)})\right\|^2, \tag{43}$$

where ν_k $(0 < \nu_k < 1)$ is the forgetting factor.

Second, we replace the unknown vector $\boldsymbol{w}^{\mathrm{o}}$ and the number of all nodes N with the instantaneous estimate $\boldsymbol{\phi}_k^{(i-1)}$ and the number of neighbors $|\mathcal{N}_k|$, respectively. The coefficients η_{kk}, η_{klj}, and η_{jl} are redefined as

$$\eta_{kk}^{(i)} = c\gamma_{kk}^{2,(i)} + \frac{\lambda}{|\mathcal{N}_k|}\mu_k \partial f(\boldsymbol{\phi}_k^{(i-1)})^{\mathrm{T}} \cdot \left\{\lambda\mu_k \partial f(\boldsymbol{\phi}_k^{(i-1)}) + 2\left(\boldsymbol{\psi}_k^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\right)\right\}, \tag{44}$$

$$\eta_{klj}^{(i)} = \frac{\lambda}{|\mathcal{N}_k|}\Big[\boldsymbol{\zeta}_k^{(i)\mathrm{T}}\left\{\lambda\mu_k \partial f(\boldsymbol{\phi}_k^{(i-1)}) + 2\left(\boldsymbol{\psi}_k^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\right)\right\} + \mu_k \partial f(\boldsymbol{\phi}_k^{(i-1)})^{\mathrm{T}}\boldsymbol{\iota}_k^{(i)}\Big], \tag{45}$$

$$\eta_{jl}^{(i)} = c\sum_{l\in\mathcal{N}_k\setminus\{k\}} \gamma_{lk}^{2,(i)} + \frac{\lambda}{|\mathcal{N}_k|}\boldsymbol{\zeta}_k^{(i)\mathrm{T}}\boldsymbol{\iota}_k^{(i)}, \tag{46}$$

where $\boldsymbol{\zeta}_k^{(i)} = \sum_{j\in\mathcal{N}_k\setminus\{k\}} \mu_j \partial f(\boldsymbol{\phi}_j^{(i-1)})$ and $\boldsymbol{\iota}_k^{(i)} = \lambda\boldsymbol{\zeta}_k^{(i)} + 2(|\mathcal{N}_k| - 1)\boldsymbol{\psi}_k^{(i)} - 2\sum_{j\in\mathcal{N}_k\setminus\{k\}} \boldsymbol{\phi}_j^{(i-1)}$. We further redefine Γ_k and Λ_k by using (44)–(46) as

$$\Gamma_k^{(i)} = 2\eta_{jl}^{(i)} - 2(|\mathcal{N}_k| - 1)\eta_{kk}^{(i)} - (|\mathcal{N}_k| - 2)\eta_{klj}^{(i)}, \tag{47}$$

$$\Lambda_k^{(i)} = 2(|\mathcal{N}_k| - 1)\eta_{kk}^{(i)} - \eta_{klj}^{(i)}, \tag{48}$$

respectively.

Algorithm 4 Sparse diffusion LMS using proposed adaptive CPO rule

Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$, $\{\gamma_{lk}^{2,(-1)}\}$ $\forall k, l$
for each time $i \geq 0$ and each node k **do**
  $\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)}) - \mu_k\lambda\partial f(\boldsymbol{\phi}_k^{(i-1)})$
  Calculate $\gamma_{lk}^{2,(i)}$ $(l \in \mathcal{N}_k)$ as in (43)
  Calculate $\eta_{kk}^{(i)}, \eta_{klj}^{(i)}, \eta_{jl}^{(i)}$ as in (44)–(46)
  Calculate $\Gamma_k^{(i)}$ and $\Lambda_k^{(i)}$ as in (47) and (48)
  if $\Gamma_k^{(i)} > 0$ **then**
    if $\Lambda_k^{(i)} > 0$ **then**
      $\beta_k^{(i)} = \frac{\Lambda_k^{(i)}}{\Gamma_k^{(i)}}$
    else
      $\beta_k^{(i)} = +0$ (small positive constant)
    end if
  else
    $\beta_k^{(i)} = +\infty$ (large positive constant)
  end if
  $a_{lk}^{(i)} = \frac{\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}}$ $(l \in \mathcal{N}_k \setminus \{k\})$, $\frac{1+\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}}$ $(l = k)$
  $\boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}^{(i)}\boldsymbol{\psi}_l^{(i)}$
end for

Summarizing these estimations and replacements, we can derive an adaptive form of the parameter β_k as below:

$$\begin{cases} \beta_k^{\mathrm{o},(i)} = \frac{\Lambda_k^{(i)}}{\Gamma_k^{(i)}}, & \text{if } \Gamma_k^{(i)} > 0 \text{ and } \Lambda_k^{(i)} > 0, \\ \beta_k^{\mathrm{o},(i)} \to +0, & \text{if } \Gamma_k^{(i)} > 0 \text{ and } \Lambda_k^{(i)} \leq 0, \\ \beta_k^{\mathrm{o},(i)} \to +\infty, & \text{if } \Gamma_k^{(i)} \leq 0. \end{cases} \tag{49}$$

The subsequent combination weight $a_{lk}^{\mathrm{cpo},(i)}$ is described as

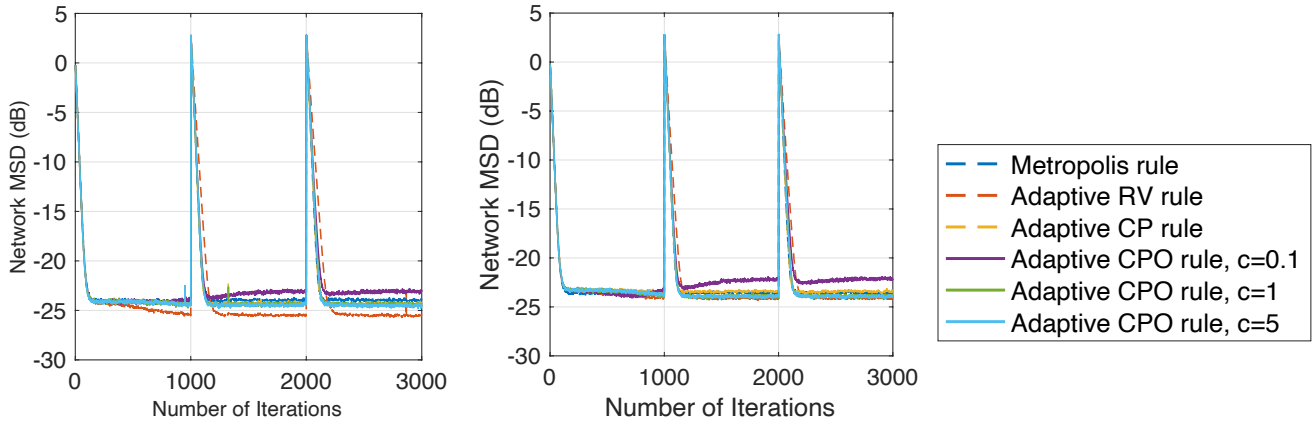
$$a_{lk}^{\mathrm{cpo},(i)} = \begin{cases} \frac{\beta_k^{\mathrm{o},(i)}}{1+|\mathcal{N}_k|\beta_k^{\mathrm{o},(i)}}, & \text{if } l \in \mathcal{N}_k \setminus \{k\}, \\ \frac{1+\beta_k^{\mathrm{o},(i)}}{1+|\mathcal{N}_k|\beta_k^{\mathrm{o},(i)}}, & \text{if } l = k, \\ 0, & \text{otherwise.} \end{cases} \tag{50}$$

SD-LMS using the proposed adaptive optimization is summarized in *Algorithm 4*. We name this weight *adaptive CP with Optimization (CPO) rule*.

It should be noted that Algorithm 4 requires a slightly higher computational complexity than Algorithm 2 and 3, but the required amount of communication is the same as Algorithm 3.

IV. Simulation Results

We have evaluated the learning performance of the proposed method via computer simulations. All the simulation results are obtained by MATLAB. In order to compare the performance in networks of different density, we have generated 4 Erdős-Rényi random networks with $N = 20$, where the mean degrees are $D = 12, 14, 16$, and 18, respectively. We have used node-independent step-size parameters $\mu_k = \mu$, forgetting factors $\nu_k = \nu$, and initial values $\gamma_{lk}^{2,(-1)} = \gamma^{2,(-1)}$,



(a) Network MSD learning curves with $N = 20$, $D = 12$.

(b) Network MSD learning curves with $N = 20$, $D = 14$.

(c) Network MSD learning curves with $N = 20$, $D = 16$.

(d) Network MSD learning curves with $N = 20$, $D = 18$.

Fig. 1: Network MSD learning curves.

for all k, l. The step-size parameter, the forgetting factor, and the parameter in the regularization function are fixed to be $\mu = 0.05$, $\nu = 0.005$, and $\epsilon = 0.001$, respectively. The initial values $\gamma^{2,(-1)}$ are set to be 1. We have fixed the regularization parameter as $\lambda = 0.0005$ for $D = 12$ and 14, and $\lambda = 0.0004$ for $D = 16$ and 18, with which have shown the best steady-state performance among our trials. The parameter c that controls the balance of MSD (36) has been chosen as $c = 0.1$, 1, or 5. The unknown vector is with size $M = 100$ and the number of nonzero elements is 1, where the index switches every 1000 iterations at uniformly random. The measurement vectors $\{\boldsymbol{u}_k^{(i)}\}$ are zero-mean real Gaussian random vectors and have time-correlated shift structures [25]. The specific structure is given by $\boldsymbol{u}_k^{(i)} = [u_k(i)\ u_k(i-1)\ \cdots\ u_k(i-M+1)]^{\mathrm{T}}$, where $u_k(\cdot)$ is i.i.d. zero-mean real Gaussian random variable with variance $\sigma_{u_k}^2$, where $\sigma_{u_k}^2 \in (0, 1]$ is drawn from uniform distribution and fixed throughout the simulations. All simulation results are obtained by averaging 100 independent trials. The measurement noise power σ_k^2 is independently generated by uniform distribution over $[0.1, 0.2]$ in each trial. We compare learning curves of SD-LMS in terms of instant network MSD $\frac{1}{N}\sum_{k=1}^{N}\|\boldsymbol{\phi}_k^{(i)} - \boldsymbol{w}^{\mathrm{o}}\|^2$ using the proposed adaptive CPO rule (Algorithm 4) with that using the conventional adaptive CP rule (Algorithm3), static Metropolis rule a_{lk}^{met} [1], and adaptive relative-variance (RV) rule $a_{lk}^{\mathrm{rv},(i)}$ [21]:

$$a_{lk}^{\mathrm{met}} = \begin{cases} \frac{1}{\max(|\mathcal{N}_k|, |\mathcal{N}_l|)} & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 1 - \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{lk} & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases}$$

$$a_{lk}^{\mathrm{rv},(i)} = \begin{cases} \frac{[\tilde{\gamma}_{lk}^{2,(i)}]^{-1}}{\sum_{m \in \mathcal{N}_k} [\tilde{\gamma}_{mk}^{2,(i)}]^{-1}} & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise.} \end{cases}$$

Figs. 1(a)–(d) show the learning curves in the case of $D = 12, 14, 16$, and 18, respectively. In Fig. 1(a), the proposed adaptive CPO rule and the conventional adaptive CP rule achieve faster convergence than adaptive RV rule and lower MSD than Metropolis rule but higher than adaptive RV rule in $i = 1001$–3000 when $c = 1$ or 5. However, as the density of the network increases, the algorithm with the proposed adaptive CPO rule achieves faster convergence and lower MSD

under the suitable choice of c than that with all the other rules. Namely, the proposed method shows the best tracking performance to the change of the unknown vector.

V. Conclusion

We have optimized the coefficients involved in our previous method [16] in terms of the steady-state network MSD of SD-LMS to achieve better convergence performance and robustness. Moreover, we have shown an adaptive implementation for tracking the optimal coefficients as well as the unknown vector. The algorithm with the proposed adaptive CPO rule has shown better tracking performance for the change of the unknown vector especially in dense networks under the suitable choice of the parameter c, at the cost of a slightly higher computational complexity.

Acknowledgment

This paper was supported in part by the Grants-in-Aid for Scientific Research no. 18H03765 and 18K04148 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.,* vol. 53, no. 1, pp. 65–78, Sept. 2004.

[2] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.,* vol. 56, no. 5, pp. 1865–1877, May 2008.

[3] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.,* vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[4] S. Wang and A. Dekorsy, "Distributed consensus-based extended Kalman filtering: A Bayesian perspective," *Proc. 2019 27th European Signal Processing Conference (EUSIPCO),* A Coruna, Spain, pp. 1–5, Sept. 2019.

[5] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Process.,* vol. 107, pp. 82–95, Feb. 2015.

[6] F. Albu, "The constrained stability least mean square algorithm for active noise control," in *Proc. 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom),* Batumi, Georgia, Jun. 2018.

[7] 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, Jul. 2008.

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

[9] A. H. Sayed, "Diffusion adaptation over networks," *Academic Press Library in Signal Processing,* vol. 3, R. Chellapa and S. Theodoridis, *Eds.,* pp. 323–454, 2014.

[10] J.-T. Kong, J.-W. Lee, S.-E. Kim, and W.-J. Song, "Diffusion LMS algorithms with multi combination for distributed estimation: Formulation and performance analysis," *Digit. Signal Process.,* vol. 71, pp. 117–130, Dec. 2017.

[11] A. Nakai-Kasai and K. Hayashi, "Diffusion LMS based on message passing algorithm," *IEEE Access,* vol. 7, no. 1, pp. 47022–47033, Apr. 2019.

[12] P. Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.,* vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[13] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory,* vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[14] D. P. Mandic, S. Kanna, and A. G. Constantinides, "On the intrinsic relationship between the least mean square and Kalman filters [Lecture Notes]," *IEEE Signal Process. Mag.,* vol. 32, no. 6, pp. 117–122, Nov. 2015.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE,* vol. 95, no. 1, pp. 215–233, Jan. 2007.

[16] A. Nakai-Kasai and K. Hayashi, "An acceleration method of sparse diffusion LMS based on message propagation," *IEICE Trans. Communications,* accepted.

[17] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory,* vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann Publishers Inc., California, 1988.

[19] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.,* vol. 58, no. 9, pp. 4795–4810, Sept. 2010.

[20] S-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," *Proc. 2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP),* San Juan, USA, pp. 317–320, Dec. 2011.

[21] 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, Jul. 2012.

[22] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications, 2nd ed.,* John Wiley & Sons, New York City, 2013.

[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc., ser. B,* vol. 58, pp. 267–288, Nov. 1996.

[24] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC),* pp. 2996–3000, Seville, Spain, Dec. 2005.

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