IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 21, 2018, accepted July 26, 2018, date of publication August 8, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2864205

Robust Distributed Clustering Algorithm Over Multitask Networks

JUN-TAEK KONG[1], DO-CHANG AHN[1], SEONG-EUN KIM[2], (Member, IEEE), AND WOO-JIN SONG[1], (Member, IEEE)

[1]Department of Electrical Engineering, Pohang University of Science and Technology, Pohang 790-784, South Korea
[2]Department of Electronics and Control Engineering, Hanbat National University, Daejeon 34158, South Korea

Corresponding author: Woo-Jin Song (wjsong@postech.ac.kr)

This work was supported in part by the Ministry of Science, ICT and Future Planning (MSIP), South Korea, through the ICT Consilience Creative Program supervised by the Institute for Information and Communications Technology Promotion under Grant IITP-R0346-16-1007 and in part by the National Research Foundation of Korea funded by the South Korean Government (MSIP) under Grant 2017R1C1B5017254.

ABSTRACT We propose a new adaptive clustering algorithm that is robust to various multitask environments. Positional relationships among optimal vectors and a reference signal are determined by using the mean-square deviation relation derived from a one-step least-mean-square update. Clustering is performed by combining determinations on the positional relationships at several iterations. From this geometrical basis, unlike the conventional clustering algorithms using simple thresholding method, the proposed algorithm can perform clustering accurately in various multitask environments. Simulation results show that the proposed algorithm has more accurate estimation accuracy than the conventional algorithms and is insensitive to parameter selection.

INDEX TERMS Decentralized clustering, multitask learning, adaptive networks, distributed estimation, diffusion adaptation.

I. INTRODUCTION

Distributed estimation over adaptive networks has become an important research area due to its diverse applications [1]–[5]. Previous research has been well explained in tutorial work [6]–[10]. For inference over networks, three classes of approach have mostly been studied: incremental algorithms [11]–[13], consensus algorithms [1], [14], [15] and diffusion algorithms [6]–[10], [16], [17]. The diffusion algorithms are attractive because they do not need cyclic path to cooperate with adjacent nodes, and show wider stability ranges and enhanced performance than the consensus algorithms [18].

Earlier work on distributed learning algorithms focused on the single-task problem in which every node in a network must estimate a single optimal parameter vector. However, many applications happen to be multitask-oriented, in which different clusters of nodes are interested in estimating different optimal parameter vectors [19]–[24]. Applications include tracking of multiple targets [25]–[27], cooperative spectrum sensing under several local interferers [28] and classification problems involving multiple models [29]–[33]. Many studies assume that cluster information is known in advance, but in practical applications the nodes usually do not know beforehand which clusters they belong to and which other nodes have the same objective [25]–[27]. If the nodes simply cooperate with all adjacent nodes without clustering, the estimation performance can be seriously degraded [34], [35].

For the solution of such problem, several adaptive clustering algorithms have been proposed [34]–[38]. First, Zhao and Sayed [34] proposed a combination rule for the adapt-then-combine (ATC) diffusion least-mean-square (DLMS) algorithm [17], which is derived by minimizing the network mean-square deviation (MSD). In [35] and [36], new combination rules for the ATC and combine-then-adapt (CTA) DLMS algorithms were proposed, which were derived in a manner similar to [34] but used different approximation of the optimal vectors. The resulting combination rules have clustering effect by giving small weights to data from nodes that appear to belong to other clusters. However, [34] is sensitive to the setting of the initial condition. Chen *et al.* [35], [36] showed better estimation accuracy than [34], but from inaccurate approximation of optimal vector used in the derivation, they usually give too much weight to data from itself.

Therefore, the performance improvement by combining estimates may be limited. In [37] and [38], clustering is performed by calculating 2-norm distance of estimates from different nodes and comparing it with a threshold parameter. If the 2-norm distance is larger than the threshold, the two nodes are determined to be in different clusters; otherwise, the two nodes are determined to be in a same cluster. Because of this simple thresholding mechanism, clustering performance of the algorithms is very sensitive to the choice of this arbitrary threshold. Furthermore, if the threshold is not set properly for a given multitask environment, performance can be severely degraded.

In this article, we propose a novel clustering method that is robust to the various multitask environments, which include different topologies, signal-to-noise ratio (SNR) values, optimum values and user parameters. Each node generates a reference signal that continuously approaches its optimal vector. Using the MSD relation derived from an one-step LMS update, the positional relationships among optimal vectors and the reference signal can be determined. By combining determinations on the positional relationships at several iterations, each node finally determines whether a neighbor node belongs to the same cluster. Afterward, any distributed estimation algorithm can be used with the clustered neighborhood information. In this paper, we use the DLMS algorithm [17] that has a simple structure yet obtains good estimation performance. From its accurate clustering, the resulting algorithm shows improved estimation accuracy than the conventional algorithms. Furthermore, the proposed algorithm is robust to various multitask environments, which means that the parameters need not be finely tuned according to the given environment. In this paper, to show the robustness of the proposed algorithm, we vary step sizes, weight tap length and optimum vectors that have a critical impact on the estimation performance of the conventional algorithms.

This work is organized as follows. We formulate the problem and briefly introduce the DLMS algorithm [17] in Section 2. We derive the proposed clustering method considering practical implementation in Section 3. We give the simulation results in Section 4, and conclude the paper in Section 5.

Notation: We use boldface letters for random variables and normal letters for deterministic quantities.

II. BACKGROUND

A. PROBLEM FORMULATION

Consider a network of N nodes that are distributed over some geographic region (Fig. 1). The set of neighbors of node k, including node k itself, is called the neighborhood of node k and is denoted by $\mathcal{N}_k$. At each time instant i, each node k collects a scalar measurement $d_k(i)$ and an $1 \times M$ regression vector $u_{k,i}$ of some random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. At each i and k, the data are assumed to be related to an $M \times 1$ unknown vector w_k^o by a linear regression model as

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

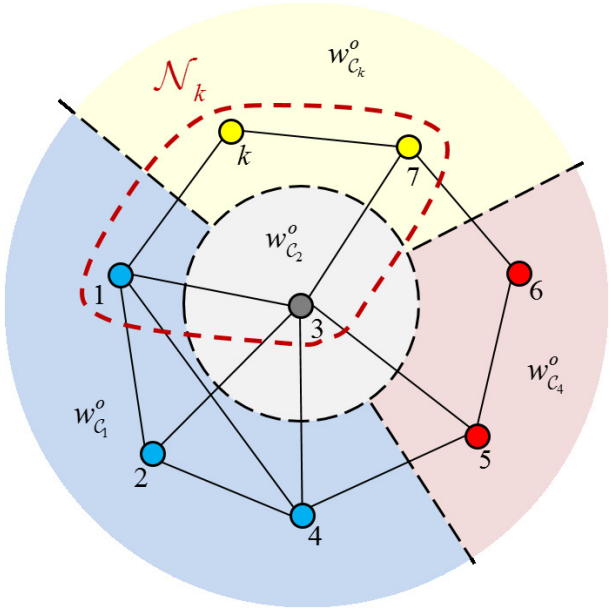

FIGURE 1. Network topology of $N = 8$ nodes in four different clusters.

where $\boldsymbol{v}_k(i)$ is zero-mean measurement noise with variance $\sigma^2_{v,k}$, and is assumed to be temporally white and spatially independent. $\boldsymbol{v}_k(i)$ and $\boldsymbol{u}_{l,h}$ are assumed to be independent of each other for all $\{k, l, i, h\}$. The objective of each node k is to estimate w_k^o in a distributed and adaptive manner by sharing information within $\mathcal{N}_k$.

B. DLMS ALGORITHM

The DLMS algorithm [17] consists of two steps: adaptation and combination. During the adaptation step, each node k updates its estimator by using the observed data $\{d_l(i), u_{l,i}\}_{l\in\mathcal{N}_k}$ that are available at node k. In the combination step, each node k calculates the weighted average of the estimators from its neighborhood. We focus on the ATC DLMS algorithm which is given as

$$\psi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{lk} u^*_{l,i}(d_l(i) - u_{l,i} w_{k,i-1}) \tag{2}$$

$$w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{lk} \psi_{l,i}, \tag{3}$$

where μ_k is a positive step size, and $\{c_{lk}, a_{lk}\}$ are non-negative weighting coefficients that satisfy

$$c_{lk} = a_{lk} = 0 \quad \text{if } l \notin \mathcal{N}_k, \ \sum_{k=1}^{N} c_{lk} = \sum_{l=1}^{N} a_{lk} = 1. \tag{4}$$

The DLMS algorithm works well in a single-task network ($w_k^o = w^o$ for $k = 1, \ldots, N$). However, in multitask scenario in which w_k^o can differ from node to node, simply using all the data from neighborhood may result in performance degradation compared to the single (no-cooperation) LMS [34], [35]. When nodes do not know which nodes in the neighborhood belong to the same cluster, a clustering process is necessary to attain the benefits of cooperation.

III. PROPOSED ALGORITHM

We consider the situation in which the cluster information is not known in advance. If the nodes simply cooperate with all adjacent nodes as usual in a single-task network, the performance further degrades with increase in the difference between optimal vectors of different clusters. Therefore, in general, to benefit from cooperation in multitask network, clustering is necessary. For clustering, we use a reference signal that is generated separately from the main estimation procedure (in this paper, DLMS); therefore, we prevent propagation of clustering error that may occur temporarily during the transient phase. Unlike the conventional clustering algorithms that use simple thresholding method, we use the positional relationship among the optimal vectors and the reference signal; this relationship can be determined by an one-step LMS update equation. From this geometrical basis, the proposed algorithm is expected to work well in various multitask environments without fine tuning of the parameters.

A. DERIVATION OF MSD RELATION

Each node k keeps running single LMS to use the result as a reference signal:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\phi}_{k,i-1}), \tag{5}$$

where μ_k is a positive step size. The objective of each node k is to distinguish whether a neighbor node l is in the same cluster as k. First, consider an LMS update of the reference signal from l at $i-1$ using the data of k:

$$\boldsymbol{\phi}_{kl,i} = \boldsymbol{\phi}_{l,i-1} + \mu_l \boldsymbol{u}_{k,i}^{*}\boldsymbol{\epsilon}_{kl}(i), \tag{6}$$

where μ_l is a positive step size and $\boldsymbol{\epsilon}_{kl}(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\phi}_{l,i-1}$. We call (6) 'one-step LMS update' because $\boldsymbol{\phi}_{l,i-1}$ is not updated recursively, but only once at each iteration i. This equation will be used to determine the positional relationship among the unknown parameters w_k^o, w_l^o and the reference signal $\boldsymbol{\phi}_{l,i-1}$. We introduce the weight error vectors from w_k^o as

$$\tilde{\boldsymbol{\phi}}_{k,l,i-1} \triangleq w_k^o - \boldsymbol{\phi}_{l,i-1}, \quad \tilde{\boldsymbol{\phi}}_{k,kl,i} \triangleq w_k^o - \boldsymbol{\phi}_{kl,i}. \tag{7}$$

Then, the LMS update (6) can be rewritten as

$$\tilde{\boldsymbol{\phi}}_{k,kl,i} = \tilde{\boldsymbol{\phi}}_{k,l,i-1} - \mu_l \boldsymbol{u}_{k,i}^{*}\boldsymbol{\epsilon}_{kl}(i). \tag{8}$$

Squaring both sides of (8) and taking expectations yields the relation between MSDs from w_k^o as

$$\mathrm{E}\left\|\tilde{\boldsymbol{\phi}}_{k,kl,i}\right\|^2 = \mathrm{E}\left\|\tilde{\boldsymbol{\phi}}_{k,l,i-1}\right\|^2 - \Delta_{kl}(i), \tag{9}$$

where

$$\Delta_{kl}(i) \triangleq 2\mu_l \mathrm{Re}\left\{\mathrm{E}\left[\boldsymbol{\epsilon}_{kl}^{*}(i)\boldsymbol{u}_{k,i}\tilde{\boldsymbol{\phi}}_{k,l,i-1}\right]\right\} - \mu_l^2 \mathrm{E}\left\|\boldsymbol{u}_{k,i}^{*}\boldsymbol{\epsilon}_{kl}(i)\right\|^2, \tag{10}$$

and $\Delta_{kl}(i) > 0$ means that $\boldsymbol{\phi}_{l,i-1}$ becomes closer to w_k^o by the one-step LMS update (6). To simplify (10), we use the following relation:

$$\boldsymbol{\epsilon}_{kl}(i) = \boldsymbol{u}_{k,i}\tilde{\boldsymbol{\phi}}_{k,l,i-1} + \boldsymbol{v}_k(i). \tag{11}$$

Then the expectation terms of (10) can be rewritten as

$$\mathrm{E}\left[\boldsymbol{\epsilon}_{kl}^{*}(i)\boldsymbol{u}_{k,i}\tilde{\boldsymbol{\phi}}_{k,l,i-1}\right] = \mathrm{E}\,|\boldsymbol{\epsilon}_{kl}(i)|^2 - \sigma_{v,k}^2, \tag{12}$$

$$\mathrm{E}\left\|\boldsymbol{u}_{k,i}^{*}\boldsymbol{\epsilon}_{kl}(i)\right\|^2 = \mathrm{E}\left\|\boldsymbol{p}_{kl,i}\right\|^2 + \mathrm{E}\left\|\boldsymbol{u}_{k,i}\right\|^2 \sigma_{v,k}^2, \tag{13}$$

where we define $\boldsymbol{p}_{kl,i} \triangleq \boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}\tilde{\boldsymbol{\phi}}_{k,l,i-1}$. Therefore, (10) is rewritten as

$$\Delta_{kl}(i) = 2\mu_l\left\{\mathrm{E}\,|\boldsymbol{\epsilon}_{kl}(i)|^2 - \sigma_{v,k}^2\right\} - \mu_l^2\left\{\mathrm{E}\left\|\boldsymbol{p}_{kl,i}\right\|^2 + \mathrm{E}\left\|\boldsymbol{u}_{k,i}\right\|^2\sigma_{v,k}^2\right\}. \tag{14}$$

B. PROPOSED ADAPTIVE CLUSTERING ALGORITHM

Considering the positional relationship among w_k^o, w_l^o and the intermediate estimate $\phi_{l,i-1}$, each node k can use $\Delta_{kl}(i)$ to determine whether a neighbor node l is in the same cluster. There are two cases to be considered.

1) $w_k^o \neq w_l^o$ (Fig. 2)

Consider when nodes k and l are in different clusters. Because $\phi_{l,i}$ constantly approaches w_l^o as i increases, it is far from w_k^o with a very high probability for every iteration. Therefore, the LMS update (6) makes the estimate $\phi_{l,i-1}$ become closer to w_k^o, i.e., $\Delta_{kl}(i) > 0$. The exception is when w_k^o is located very close to the trajectory of the update (5) at node l at some iterations, which will also be considered later by stacking method and cross checking between two nodes.

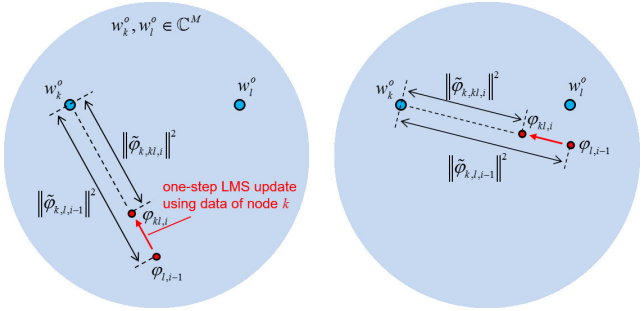

FIGURE 2. Positional relationship among estimates and unknown vectors $w_k^o \neq w_l^o$ at early stage of the update (6) (left) and after sufficient convergence to w_l^o (right).

2) $w_k^o = w_l^o$ (Fig. 3)

Consider when nodes k and l are in the same cluster. At the early stage of the update (6) at node l (Fig. 3, left), $\phi_{l,i-1}$ is far from w_k^o so that $\Delta_{kl}(i) > 0$. As $\phi_{l,i-1}$ approaches to w_k^o (Fig. 3, right), the contribution of measurements decreases [39], which means that the probability that the update (5) will reduce MSD decreases. Therefore, either $\Delta_{kl}(i) > 0$ or $\Delta_{kl}(i) < 0$. Motivated by this insight, we proceed to derive a new clustering algorithm.

From these observations, the sign of $\Delta_{kl}(i)$ will be used for clustering. For practical implementation, we first estimate the

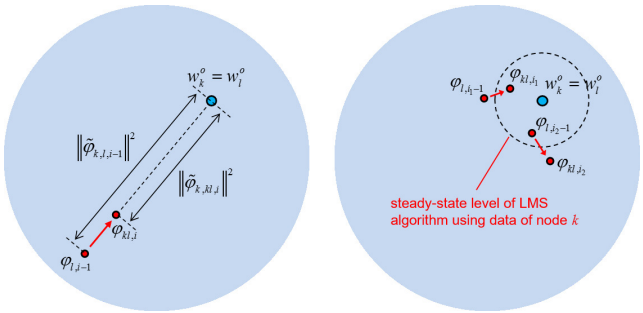

FIGURE 3. Positional relationship among estimates and unknown vectors $w_k^o = w_l^o$ at early stage of the update (6) (left) and after sufficient convergence to w_l^o (right).

noise variance $\sigma_{v,k}^2$ as in [40]:

$$\hat{\sigma}_{v,k}^2 = \mathrm{E}\,\|\boldsymbol{\epsilon}_{kk}(i)\|^2 - \frac{\mathrm{E}\,\|\boldsymbol{p}_{kk,i}\|^2}{\mathrm{E}\,\|\boldsymbol{u}_{k,i}\|^2}. \tag{15}$$

Also, the statistical values in (14) are not available in practice. We estimate $\mathrm{E}\|\boldsymbol{u}_{k,i}\|^2$ and $\mathrm{E}\,|\boldsymbol{\epsilon}_{kl}(i)|^2$ by time-averaging methods [41] as

$$\|\hat{u}_{k,i}\|^2 = \alpha\,\|\hat{u}_{k,i-1}\|^2 + (1-\alpha)\,\|u_{k,i}\|^2, \tag{16}$$

$$\hat{\sigma}_{\epsilon_{kl}(i)}^2 = \alpha\hat{\sigma}_{\epsilon_{kl}(i-1)}^2 + (1-\alpha)\epsilon_{kl}^2(i), \tag{17}$$

where $0 \le \alpha \le 1$ is a forgetting factor. Using the relation $\mathrm{E}\left[\boldsymbol{p}_{kl,i}\right] = \boldsymbol{E}\left[\boldsymbol{u}_{k,i}^*\boldsymbol{\epsilon}_{kl}(i)\right]$, we estimate $\mathrm{E}\left[\boldsymbol{p}_{kl,i}\right]$ by time-averaging as

$$\hat{p}_{kl,i} = \alpha\hat{p}_{kl,i-1} + (1-\alpha)u_{k,i}^*\epsilon_{kl}(i) \tag{18}$$

and use $\|\hat{p}_{kl,i}\|^2$ instead of $\mathrm{E}\,\|\boldsymbol{p}_{kl,i}\|^2$ [41]. Using (14)–(18), inequality $\Delta_{kl}(i) > 0$ can be rewritten as

$$\hat{\sigma}_{\epsilon_{kl}(i)}^2 > \frac{\mu_l}{2}\,\|\hat{p}_{kl,i}\|^2 + \left(1 + \frac{\mu_l}{2}\,\|\hat{u}_{k,i}\|^2\right)\left(\hat{\sigma}_{\epsilon_{kk}(i)}^2 - \frac{\|\hat{p}_{kk,i}\|^2}{\|\hat{u}_{k,i}\|^2}\right). \tag{19}$$

If inequality (19) is satisfied, we assign $b_{kl}(i) = 1$; otherwise, we assign $b_{kl}(i) = 0$, and we stack L past values in an $L \times 1$ vector $B_{kl,i} \triangleq [b_{kl}(i), b_{kl}(i-1), \ldots, b_{kl}(i-L+1)]^T$. In general situations, $\Delta_{kl}(i)$ is always positive when $w_k^o = w_l^o$, whereas $\Delta_{kl}(i)$ can be either positive or negative when $w_k^o \neq w_l^o$. Therefore, the two cases can be distinguished by observing values stacked in $B_{kl,i}$; overwhelming dominance of 1 means that nodes are in different clusters. We propose to choose the connection between nodes k and $l \neq k$ at iteration i by using the probability that the inequality (19) is satisfied as follows:

$$t_{kl}(i) = \begin{cases} 0 & \text{if } \Sigma_{m=1}^{L} B_{kl,i}/L > p, \\ 1 & \text{otherwise,} \end{cases} \tag{20}$$

where $\{t_{kl}(i)\}$ are entries of T_i, $N \times N$ adjacency matrix at iteration i and $0 < p < 1$ is an user parameter. p should be chosen to be close to 1, because in general $\Delta_{kl}(i) > 0$ when nodes k and l are in different clusters. In addition, we disconnect the link $k \to l$ if the link in the opposite direction $l \to k$ is disconnected:

$$t_{kl}(i) = 0 \quad \text{if } t_{lk}(i) = 0 \quad \text{for } l \in \mathcal{N}_k^- \tag{21}$$

where $\mathcal{N}_k^-$ denotes the set $\mathcal{N}_k$ except for node k. By the cross checking in (21), clustering can be done accurately for the rare exceptions when $w_k^o \neq w_l^o$ and w_k^o is located very close to the trajectory of update (5) at node l. Of course, each node always uses its data without the above procedure, i.e., $t_{kk} = 1$ for all k. We denote clustered neighborhood of node k at iteration i as $\mathcal{N}_{k,i}$. Now, each node k can use the data from $\mathcal{N}_{k,i}$ to perform any distributed estimation algorithm. Table 1 shows the pseudo code of the proposed algorithm when using the ATC DLMS for the estimation process.

TABLE 1. Pseudo code of the proposed algorithm.

Initialization : For all $k = 1, \ldots, N$ and $l \in \mathcal{N}_k$,
set $\hat{p}_{kl,0} = \overrightarrow{0}_M$, $\|\hat{u}_{k,0}\|^2 = \hat{\sigma}_{\epsilon_{kl}(0)}^2 = w_{k,0} = w_{k,0}^s = 0$,
$b_{kl}(0) = \ldots = b_{kl}(-L+1) = 1$.

At each time instant $n \ge 1$ and for each node k,

(1) Distributed clustering

Set $t_{kk}(i) = 1, t_{kl}(i) = 0$ for all $l \ge \mathcal{N}_k^-$.

$\|\hat{u}_{k,i}\|^2 = \alpha\,\|\hat{u}_{k,i-1}\|^2 + (1-\alpha)\,\|u_{k,i}\|^2$

for $l \in \mathcal{N}_k$

$\hat{p}_{kl,i} = \alpha\hat{p}_{kl,i-1} + (1-\alpha)u_{k,i}^*\epsilon_{kl}(i)$

$\hat{\sigma}_{\epsilon_{kl}(i)}^2 = \alpha\hat{\sigma}_{\epsilon_{kl}(i-1)}^2 + (1-\alpha)\epsilon_{kl}^2(i)$

end

for $l \in \mathcal{N}_k^-$

$$b_{kl}(i) = \begin{cases} 1, & \text{if } \hat{\sigma}_{\epsilon_{kl}(i)}^2 > \frac{\mu_l}{2}\,\|\hat{p}_{kl,i}\|^2 \\ & \quad + \left(1 + \frac{\mu_l}{2}\,\|\hat{u}_{k,i}\|^2\right)\left(\hat{\sigma}_{\epsilon_{kk}(i)}^2 - \frac{\|\hat{p}_{kk,i}\|^2}{\|\hat{u}_{k,i}\|^2}\right) \\ 0, & \text{otherwise} \end{cases}$$

$B_{kl,i} = [b_{kl}(i), b_{kl}(i-1), \ldots, b_{kl}(i-L+1)]$

$$t_{kl}(i) = \begin{cases} 0, & \text{if } \Sigma_{m=1}^{L} B_{kl,i}(m) > Lp \\ 1, & \text{otherwise} \end{cases}$$

Transmit $t_{kl}(i)$ to $l \in \mathcal{N}_k^-$.

$t_{kl}(i) = 0$ if $t_{lk}(i) = 0$

end

(2) Distributed estimation

Select $\{c_{lk}(i)\}$ that satisfy

$c_{kk}(i) > 0, c_{lk}(i) = 0$ for $l \notin \mathcal{N}_{k,i}$, $\sum_{l=1}^{N} c_{kl}(i) = 1$.

$\phi_{k,i} = \phi_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\phi_{k,i-1})$

$\psi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_{k,i}} c_{lk}(i)u_{k,i}^*(d_k(i) - u_{k,i}w_{k,i-1})$

Transmit $\phi_{k,i}$ to $l \in \mathcal{N}_k^-$ and $\psi_{k,i}$ to $l \in \mathcal{N}_{k,i}^-$.

Select $\{a_{lk}(i)\}$ that satisfy

$a_{kk}(i) > 0, a_{lk}(i) = 0$ for $l \notin \mathcal{N}_{k,i}$, $\sum_{l=1}^{N} a_{lk}(i) = 1$.

$w_{k,i} = \sum_{l\in\mathcal{N}_{k,i}} a_{lk}(i)\psi_{l,i}$

The conventional algorithms [36], [37] perform clustering by comparing the 2-norm distance of the estimates with an arbitrary threshold. This threshold must be less than $\delta \triangleq \min\{\|w_k^o - w_l^o\|^2 \mid k = 1, \ldots, N,\ l \in \mathcal{N}_k,\ w_k^o \neq w_l^o\}$, which is the minimum value of 2-norm distances of optimal vectors between every two adjacent nodes in different clusters. This value is unknown in general applications. Even if

the threshold is chosen to be $< \delta$, these methods still have the disadvantage that the performance is sensitive to the setting of the threshold value. In contrast, the proposed algorithm is insensitive to the choice of the parameters L and p, and can perform clustering accurately in various multitask environments.

TABLE 2. Computational complexity (×, +) and communication cost (C) of the conventional clustering algorithms and the proposed algorithm for node k at each time i. Communication cost denotes the number of scalar values which are transmitted from node k.

[35],[36]	×	$(3\lvert\mathcal{N}_k\rvert+2)M+\lvert\mathcal{N}_k\rvert+1$
	+	$(4\lvert\mathcal{N}_k\rvert+3)M-\lvert\mathcal{N}_k\rvert-2$
	C	$(\lvert\mathcal{N}_k\rvert-1)M$
[37]	×	$(\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+3)M+2$
	+	$(2\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+3)M-\lvert\mathcal{N}_k\rvert-1$
	C	$(\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert-2)M$
[38]	×	$(\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+1)M+2\lvert\mathcal{N}_k\rvert-1$
	+	$(2\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+1)M+2\lvert\mathcal{N}_k\rvert-\lvert\mathcal{N}_{k,i}\rvert-3$
	C	$(\lvert\mathcal{N}_k\rvert-1)M$
Proposed	×	$(2\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+5)M+8\lvert\mathcal{N}_k\rvert+5$
	+	$(3\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert+6)M+(\lvert\mathcal{N}_k\rvert-1)L+2\lvert\mathcal{N}_k\rvert-\lvert\mathcal{N}_{k,i}\rvert$
	C	$(\lvert\mathcal{N}_k\rvert+\lvert\mathcal{N}_{k,i}\rvert-2)M+\lvert\mathcal{N}_k\rvert-1$

Table 2 shows computational complexity and communication cost of the conventional clustering algorithms and the proposed algorithm. For fair comparison, we use $c_{lk}=0$ for $l\neq k$ (no information exchange) and any deterministic rule for the combination weights $a_{lk}(i)$. Also for [35] and [36], we assume that the algorithms don't use normalization of $q_k(n)$, which is optional. The results are given as linear combinations of variables $M, L, |\mathcal{N}_k|$ and $|\mathcal{N}_{k,i}|$, where $|\mathcal{N}_k|$ denotes the degree of node k. By comparing the coefficients of the resulting formulas, we note that the proposed algorithm needs more computations and communications than the conventional methods. Nonetheless, with these additional requirements, the proposed algorithm shows improved estimation accuracy and robustness to various multitask environments. We will examine these features in the following simulation section.

IV. SIMULATION RESULTS

To illustrate the performance of the proposed clustering algorithm, we present simulations for a network topology with $N=80$ nodes (Fig. 4, left), and specified statistical profiles of regressor power and noise power (Fig. 4, right). We consider channel identification problem of an FIR model with channel length of $M=2$ taps. The regressors are zero-mean Gaussian, spatially independent and temporally white. The step sizes were set to $\mu_k=0.03$ and all simulation results were obtained by taking the ensemble average of the network MSD:

$$\text{MSD}^{\text{network}}=\frac{1}{N}\sum_{k=1}^{N}\text{E}\|w_k^o-\boldsymbol{w}_{k,i}\|^2 \tag{22}$$

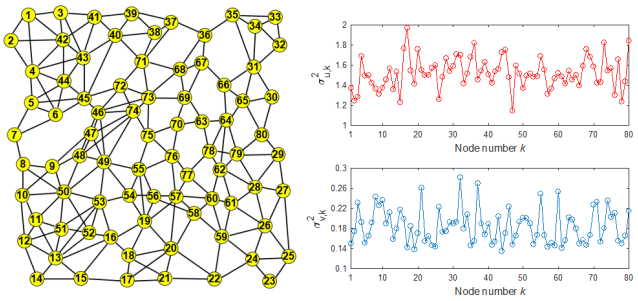

FIGURE 4. Network topology for $N=80$ nodes (left), regressor powers $\sigma^2_{u,k}$ (top right) and noise variances $\sigma^2_{v,k}$ (bottom right) for each node.

over 200 independent experiments. We use the uniform rule $a_{lk}(i)=\frac{1}{|\mathcal{N}_{k,i}|}$ [42] for the combination step. For some conventional methods, each node uses only its data during the adaptation step, so that we assume that no information is exchanged during the adaptation step ($c_{lk}=0$ if $l\neq k$) for fair comparison. We use $\alpha=0.95$, $p=0.9$ and $L=30$ for the proposed algorithm. For the conventional algorithms, we use the parameter settings used in the original papers: $\xi=0.01$ for [35] and [36], $\nu=0.98, \alpha=0.015, \gamma=0.5$ for [38] and $\theta=0.015$ for [37] (because [37] do not show parameter settings, the value of θ is chosen to be the same as $\alpha=0.015$ in [38]).

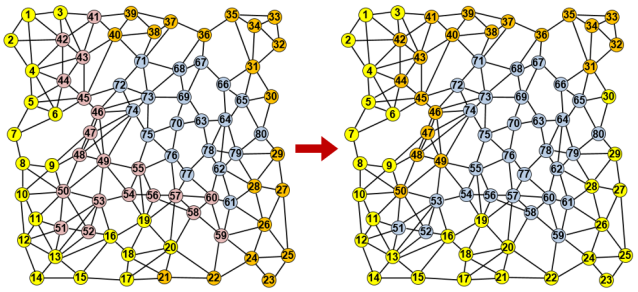

FIGURE 5. Change of cluster structure of network where nodes in the same cluster are painted with the same color.

We first simulated an environment where the cluster structure changed abruptly in the middle of iteration (Fig. 5). The unknown vectors were

$$w_k^o=\begin{cases}[1,1]^T, & k=1,\ldots,20\\ [1.5,1]^T, & k=21,\ldots,40\\ [0.6,0.6]^T, & k=41,\ldots,60\\ [2,0.5]^T, & k=61,\ldots,80\end{cases} \tag{23}$$

for $0<i\leq 500$, and

$$w_k^o=\begin{cases}[1.5,1.5]^T, & k=1,\ldots,30\\ [2,1]^T, & k=31,\ldots,50\\ [0.7,2.3]^T, & k=51,\ldots,80\end{cases} \tag{24}$$

for $500<i\leq 1000$. The transient network MSD curves were obtained for the single (no-cooperation) LMS, DLMS with known cluster structure of the network, which is denoted by T^o, and the various clustering algorithms (Fig. 6). All of

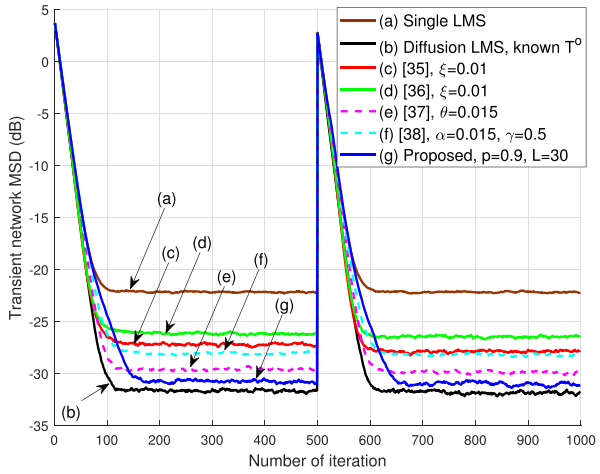

FIGURE 6. Transient network MSD for the conventional algorithms and the proposed algorithm in an environment where the cluster structure was abrubtly changed at $i = 501$.

the clustering algorithms had MSD between that of the single LMS and the DLMS with known T^o, and kept track of the unknown vectors well in an environment where unknown vectors and cluster structure both changed abruptly.

As a result of inaccurate approximation of the unknown w_k^o, the algorithms in [37] and [38] had higher steady-state MSD than the other clustering algorithms (Fig. 6, c and d). The proposed algorithm (Fig. 6, g) had a slightly slower convergence speed than the other algorithms because it needs some iterations to perform clustering correctly when two neighbor nodes have the same unknown vector (as explained in Section 3 with Fig. 3). However, the proposed algorithm had the lowest steady-state MSD among the clustering algorithms. This result indicates that the proposed algorithm has the best clustering accuracy.

The unknown vectors were set as (23) for the rest of the simulations. The transient network MSD curves were obtained for $\mu = 0.03$ (Fig. 7, top) and $\mu = 0.06$ (Fig. 7, bottom). As μ increased, the steady-state MSDs of the conventional clustering algorithms were degraded, especially that of [37]. As μ increases, the variation of the estimate value increases, so that often, two nodes in the same cluster can be identified as different cluster nodes. However, the proposed algorithm always had similar steady-state MSD to that of the DLMS with known T^o.

The transient network MSD curves were obtained for $M = 4$ (Fig. 7, top) and $M = 8$ (Fig. 7, bottom) with fixed step size $\mu = 0.03$. We increased M by simply adding zeros for the added taps; for example, the unknown vectors when $M = 4$ were set as

$$w_k^o = \begin{cases} [1, 1, 0, 0]^T, & k = 1, \ldots, 20 \\ [1.5, 1, 0, 0]^T, & k = 21, \ldots, 40 \\ [0.6, 0.6, 0, 0]^T, & k = 41, \ldots, 60 \\ [2, 0.5, 0, 0]^T, & k = 61, \ldots, 80. \end{cases} \tag{25}$$

As M increased, the performances were severely degraded for the conventional algorithms in [37] and [38] (Fig. 8, e, f),

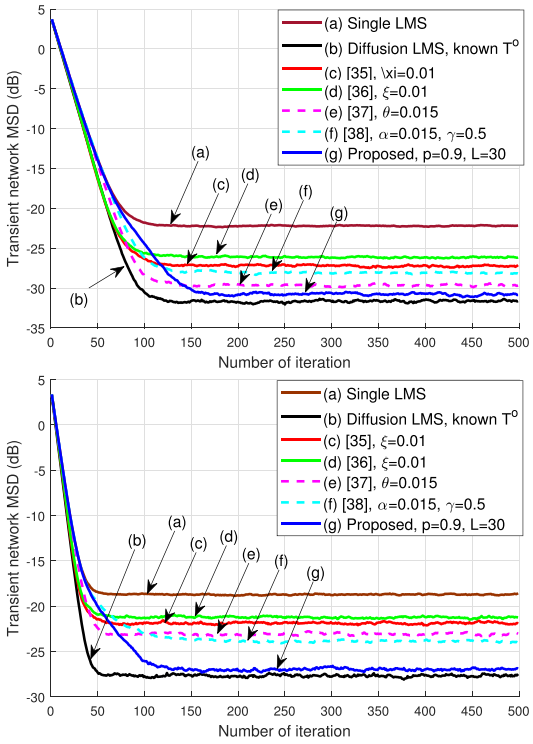

FIGURE 7. Transient network MSD for the conventional algorithms and the proposed algorithm for $\mu = 0.03$ (top) and $\mu = 0.06$ (bottom).

and when $M = 8$, they even had the same MSD levels as the single LMS (Fig. 8, a); this result means that each node k determined all of its neighbor nodes as different cluster nodes. Because those algorithms simply compare the 2-norm distance of estimates between adjacent nodes to arbitrary threshold θ and α, the clustering performance is highly dependent on the parameter setting and network environment. As M increased, the 2-norm distance increased both when the nodes belonged to the same cluster and when they belonged to different clusters; this result means that θ and α must be increased somewhat to maintain the clustering performance. On the other hand, the proposed algorithm (Fig. 8, g) had the similar steady-state MSD with the DLMS with known T^o (Fig. 8, b) for various tap lengths. To summarize, the proposed algorithm is robust to the various step sizes μ_k and tap lengths M, whereas for [37] and [38] to perform clustering accurately their parameters should be finely tuned to suit the environment.

The transient network MSD curves of the proposed algorithm were obtained for various L (Fig. 9, top) and various p (Fig. 9, bottom). L is length of the stacking vector $B_{kl,i}$, so that the clustering accuracy can be improved by using large value. p is an user parameter that should be chosen to be close to 1; high p can cause miss detection (fail to detect same cluster node) and low p can cause false alarm. Nevertheless, the proposed algorithm had similar MSD curves for various choices of L and p. The results show that the proposed algorithm is insensitive to the choice of the parameters.

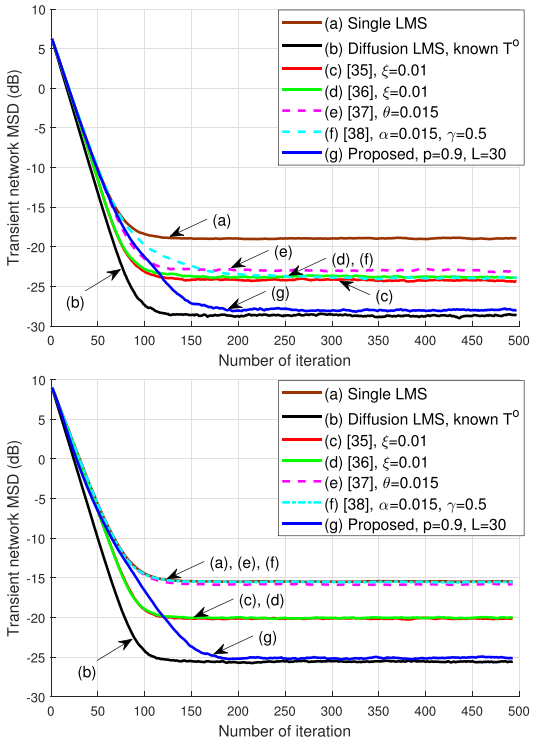

FIGURE 8. Transient network MSD for the conventional algorithms and the proposed algorithm for $M = 4$ (top) and $M = 8$ (bottom).

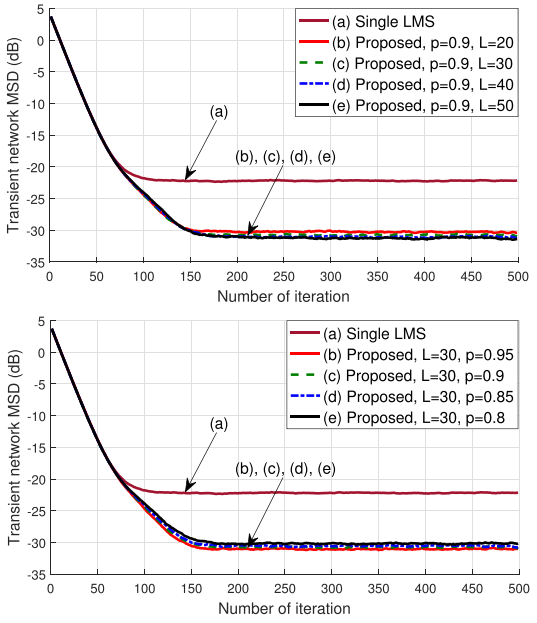

FIGURE 9. Transient network MSD for the proposed algorithm with various L (top) and various p (bottom).

We simulated in a network composed of two clusters (Fig. 10) with unknown vectors

$$w_k^o = \begin{cases} [1,1]^T, & k = 1,\ldots,40 \\ [1,1]^T + c_{\text{diff}}\,[0.1,-0.1]^T, & k = 41,\ldots,80 \end{cases} \quad (26)$$

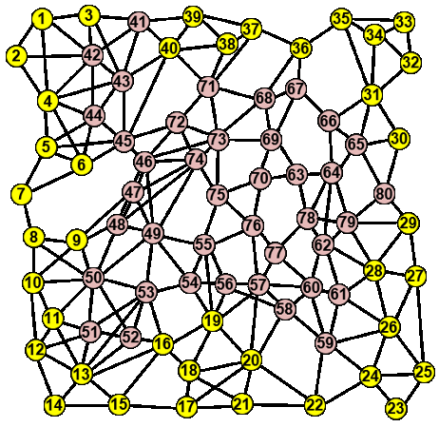

FIGURE 10. Network topology for $N = 80$ nodes in two clusters.

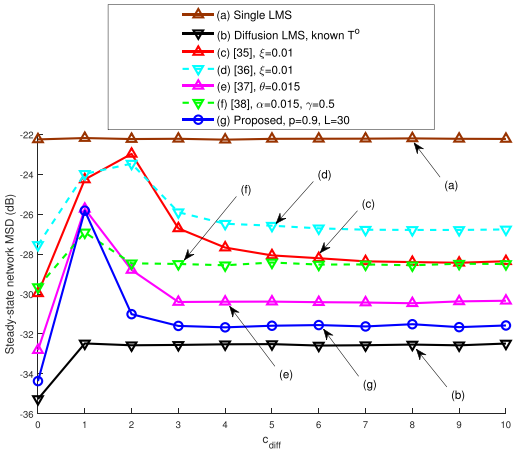

FIGURE 11. Steady-state network MSD for the conventional algorithms and the proposed algorithm with various c_{diff}.

where c_{diff} is a non-negative integer. For various c_{diff} values, the steady-state MSD was obtained by averaging MSDs at $401 \le i \le 500$ for 200 independent experiments (Fig. 11). For small $c_{\text{diff}} > 0$, because two different unknown vectors have similar values, and the decision of whether they are in the same cluster is a difficult task, the steady-state MSDs of the clustering algorithms were most degraded for $c_{\text{diff}} = 1$. Other than that, the proposed algorithm always had the lowest steady-state MSD among the clustering algorithms. This result confirms that the proposed algorithm works well in various multitask environments.

V. CONCLUSION

We proposed a new adaptive clustering algorithm that determines positional relationships among the optimal vectors and a reference signal by using the MSD relation derived from an LMS update equation. The proposed algorithm had slightly slower convergence speed, but achieved lower steady-state MSD than the conventional algorithms. The proposed algorithm performed clustering well in a variety of environments that had different M, μ and optimal vectors. The proposed

algorithm was insensitive to the choice of parameters L and p. The greatest advantage of the proposed algorithm is its ability to work well in various environments without careful tuning of the parameters.

REFERENCES

[1] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
[2] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.
[3] F. S. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2038–2051, May 2011.
[4] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, Aug. 2011.
[5] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3183–3197, Jun. 2013.
[6] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.
[7] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3, R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Elsevier, 2014, pp. 323–454.
[8] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.
[9] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
[10] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, nos. 4–5, pp. 311–801, 2014.
[11] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, 1997.
[12] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.
[13] 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.
[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
[15] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.
[16] 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.
[17] 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.
[18] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.
[19] 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.
[20] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks with common latent representations," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 563–579, Apr. 2017.
[21] 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.
[22] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2196–2210, May 2011.
[23] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 5425–5429.
[24] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 7223–7227.
[25] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 36–46, May 2007.
[26] X. Zhang, "Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2429–2444, Oct. 2011.
[27] M. Z. Lin, M. N. Murthi, and K. Premaratne, "Mobile adaptive networks for pursuing multiple targets," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, South Brisbane, QLD, Australia, Apr. 2015, pp. 3217–3221.
[28] 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, Jul. 2015.
[29] I. Francis and S. Chatterjee, "Classification and estimation of several multiple regressions," *Ann. Statist.*, vol. 2, no. 3, pp. 558–561, 1974.
[30] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, Apr. 1996.
[31] V. Cherkassky and Y. Ma, "Multiple model regression estimation," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 785–798, Jul. 2005.
[32] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Oxford, U.K.: Academic, 2009.
[33] L. Jacob, J.-P. Vert, and F. Bach, "Clustered multi-task learning: A convex formulation," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2008, pp. 745–752.
[34] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. Int. Workshop Cognit. Inf. Process. (CIP)*, Baiona, Spain, May 2012, pp. 1–6.
[35] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.
[36] J. Chen, C. Richard, and A. H. Sayed, "Adaptive clustering for multitask diffusion networks," in *Proc. IEEE 23rd Eur. Signal Conf. (EUSIPCO)*, Aug./Sep. 2015, pp. 200–204.
[37] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.
[38] S. Khawatmi, A. H. Sayed, and A. M. Zoubir, "Decentralized clustering and linking by networked agents," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3526–3537, Jul. 2017.
[39] J. W. Lee, S. E. Kim, and W. J. Song, "Data-selective diffusion LMS for reducing communication overhead," *Signal Process.*, vol. 113, pp. 211–217, Aug. 2015.
[40] M. A. Iqbal and S. L. Grant, "Novel variable step size NLMS algorithms for echo cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, NV, USA, Mar./Apr. 2008, pp. 241–244.
[41] H.-C. Shin, A. H. Sayed, and W.-J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132–135, Feb. 2004.
[42] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decis. Control Eur. Control Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.

JUN-TAEK KONG was born in Pyeongtaek, South Korea, in 1989. He received the B.S. degree in electronic and electrical engineering from the Pohang University of Science Technology (POSTECH), South Korea, in 2012, where he is currently pursuing the Ph.D. degree.

Since 2012, he has been a Research Assistant at the Department of Electronic and Electrical Engineering, POSTECH. His research interests are signal processing, adaptive filtering, and adaptive networks.

DO-CHANG AHN was born in Busan, South Korea, in 1986. He received the B.S. degree in electronic and electrical engineering from the Pohang University of Science Technology (POSTECH), South Korea, in 2010, where he is currently pursuing the Ph.D. degree.

Since 2010, he has been a Research Assistant at the Department of Electronic and Electrical Engineering, POSTECH. His research interests are signal processing, adaptive filtering, and adaptive networks.

SEONG-EUN KIM received the B.Sc. and Ph.D. degrees in electronics and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2004 and 2010, respectively. From 2011 to 2014, he has been a Research Staff Member at the Samsung Advanced Institute of Technology, Yongin, South Korea. From 2014 to 2017, he was a Post-Doctoral Associate with the MIT/Harvard Neuroscience Statistics Research Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently an Assistant Professor at the Department of Electronics and Control Engineering, Hanbat National University, Daejeon, South Korea. His research interests include statistical and adaptive signal processing, biomedical signal processing, and systems neuroscience.

WOO-JIN SONG was born in Seoul, South Korea, in 1956. He received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, in 1979 and 1981, respectively, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1986.

From 1981 to 1982, he was with the Electronics and Telecommunication Research Institute, Daejeon, South Korea. In 1986, he was a Senior Engineer with Polaroid Corporation, where he was involved in digital image processing. In 1989, he was promoted to Principal Engineer at Polaroid. In 1989, he joined the faculty at the Pohang University of Science and Technology, Pohang, South Korea, where he is currently a Professor of electronic and electrical engineering. His current research interests are in the area of digital signal processing, in particular, radar signal processing, signal processing for digital television and multimedia products, and adaptive signal processing.