DECENTRALIZED CLUSTERING OVER ADAPTIVE NETWORKS

Sahar Khawatmi[*], *Abdelhak M. Zoubir*[*], *Ali H. Sayed*[†]

[*] Technische Universität Darmstadt
Signal Processing Group
64283 Darmstadt, Germany
Email: {khawatmi,zoubir}@spg.tu-darmstadt.de

[†] University of California, Los Angeles
Department of Electrical Engineering
Los Angeles, CA 90095, USA
Email: sayed@ee.ucla.edu

ABSTRACT

Cooperation among agents across the network leads to better estimation accuracy. However, in many network applications the agents infer and track different models of interest in an environment where agents do not know beforehand which models are being observed by their neighbors. In this work, we propose an adaptive and distributed clustering technique that allows agents to learn and form clusters from streaming data in a robust manner. Once clusters are formed, cooperation among agents with similar objectives then enhances the performance of the inference task. The performance of the proposed clustering algorithm is discussed by commenting on the behavior of probabilities of erroneous decision. We validate the performance of the algorithm by numerical simulations, that show how the clustering process enhances the mean-square-error performance of the agents across the network.

Index Terms— Decentralized clustering, multitask networks, self-organization, diffusion adaptation, adaptive networks.

1. INTRODUCTION AND RELATED WORK

We consider a distributed mean-square-error estimation problem over an N-agent network. The connectivity of the agents is described by a graph. We assume that the data sensed by any particular agent can arise from one of C models. Each model is represented by an $M \times 1$ weight vector denoted by $w^o_{\mathcal{C},c}$, where $c = 1, \ldots, C$. There are many applications in practice where agents can be subjected to data from different sources as it happens, for example, in target tracking or swarming towards different food sources (cf. [1–10]).

In most prior works, it is generally assumed that each agent knows which neighbors are influenced by the same model. In this work, we assume that the agents do not know which model generated their received data; they also do not know which other agents in their neighborhood sense data arising from the same model. We are then interested in performing clustering. By clustering, we mean the determination of the sets of connected agents that are interested in the same model.

A useful strategy for clustering over adaptive networks was proposed in [11], relying on the use of adaptive combination weights. This strategy was further refined in [12] to reduce its sensitivity to initial conditions. Under these strategies, there still exists a possibility that links between some agents belonging to the same cluster may be overlooked. In order to avoid this difficulty and obtain a more robust method, the work in [13] proposed an alternative construction where the clustering and inference tasks are separated from each other. For sufficiently small step-sizes, this approach was shown to lead to probabilities of error that decay exponentially to zero. Motivated by [13], we propose a modified strategy where we merge the clustering and inference tasks, thus reducing the computation burden while enhancing the accuracy of the clustering step relative to [11, 12].

The work is organized as follows, the network model is described in Section 2. In Section 3 we motivate the diffusion LMS algorithm for our distributed estimation task. The clustering technique and how the agents compute its combination weights are illustrated in detail in Section 4. The probabilities of erroneous decision are analyzed in Section 5. Simulations results are presented in Section 6. Finally, we conclude the work in Section 7.

Notation. We use lowercase letters to denote vectors, uppercase letters for matrices, plain letters for deterministic variables, and boldface letters for random variables. $\mathbb{E}$ denotes the expectation operator and $\|\cdot\|$ the Euclidean norm. All vectors are column vectors, except for the regression vectors $\boldsymbol{u}_{k,i}$, which are row vectors. The symbols $\mathbb{1}$ and I denote the all-one vector and identity matrix of appropriate sizes, respectively. We write $(\cdot)^\intercal$, $(\cdot)^*$, and $\mathrm{Tr}(\cdot)$ to denote transposition, complex conjugate-transposition, and the matrix trace operation, respectively.

[†] The work of A. H. Sayed was supported in part by NSF grants CCF-1011918 and ECCS-1407712.

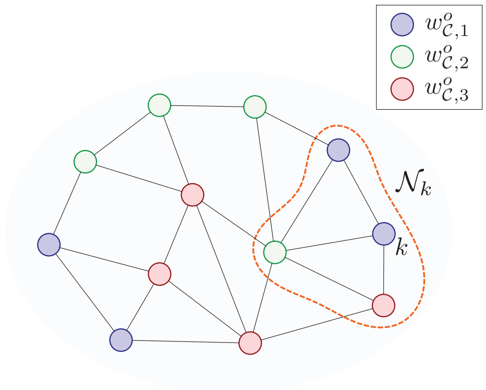

Fig. 1: Illustration of a network topology with three clusters represented by the three colors. Note that agent k has neighbors that sense data originating from different clusters than its own.

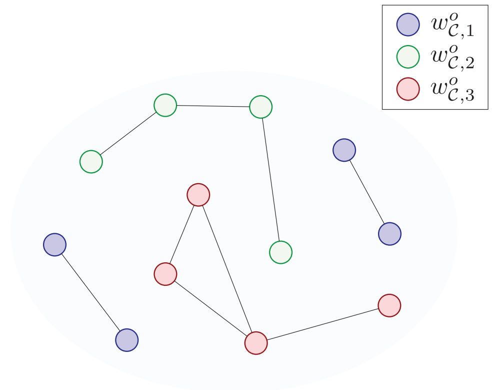

Fig. 2: Illustration of the clustered topology that will result once agents identify which neighbors belong to the same cluster and cut links to the remaining neighbors.

2. NETWORK MODEL

Figure 1 illustrates the network structure for a case involving $C = 3$ three unknown models. The unknown models are denoted by $\{w^o_{\mathcal{C},1}, w^o_{\mathcal{C},2}, w^o_{\mathcal{C},3}\}$. In the figure, agents with the same color belong to the same cluster and are therefore interested in estimating the same parameter vector. We denote the set of neighbors of an agent k by $\mathcal{N}_k$. We may represent the network topology by means of the $N \times N$ adjacency matrix E whose entries $e_{\ell k}$ are defined as follows:

$$e_{\ell k} = \begin{cases} 1, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

We assume that each agent k belongs to its neighborhood set, $k \in \mathcal{N}_k$. Agents know their neighborhoods but they do not know which subset of their neighbors is subjected to data from the same model. In order to devise a procedure that allows them to arrive at this information, we introduce an $N \times N$ clustering matrix E_i, at time i, in a manner similar to the adjacency matrix, except that the value at location (ℓ, k) will be set to one if agent k believes at time i that its neighbor ℓ belongs to the same cluster. In this way, the data exchange among the agents will be applied based on the set $\mathcal{N}_{k,i}$, which is deduced from matrix E_i instead of the original adjacency matrix, E. Each entry $e_{\ell k}(i)$ from the matrix E_i is computed according to the clustering scheme described in the sequel.

3. DATA MODEL AND DIFFUSION STRATEGY

At every time instant i, every agent k has access to a scalar measurement $\boldsymbol{d}_k(i)$ and a $1 \times M$ regression vector $\boldsymbol{u}_{k,i}$. The measurements across all agents are assumed to be generated via the linear regression model:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o_k + \boldsymbol{v}_k(i) \tag{2}$$

where w^o_k denotes the unknown model for agent k. All random processes are assumed to be stationary. Moreover, $\boldsymbol{v}_k(i)$ is a zero-mean white measurement noise that is independent over space and has variance $\sigma^2_{v,k}$. It is assumed that the regression data $\boldsymbol{u}_{k,i}$ is a zero-mean Gaussian process, independent over time and space, and independent of $\boldsymbol{v}_\ell(j)$ for all k, ℓ, i, j. We denote the covariance matrix of $\boldsymbol{u}_{k,i}$ by $R_{u,k} \triangleq \mathbb{E}\boldsymbol{u}^*_{k,i}\boldsymbol{u}_{k,i}$. In (2), the unknown models $\{w^o_k\}$, $k = 1, \ldots, N$, arise from the C models, i.e., $w^o_k = w^o_{\mathcal{C},c}$ for some c . We stack the w^o_k into a column vector:

$$w^o \triangleq \text{col}\{w^o_1, w^o_2, \ldots, w^o_N\}. \tag{3}$$

We seek the optimal estimator that minimizes the following global cost function over the vectors $\{w_k\}$:

$$J(w_1, w_2, \ldots, w_N) \triangleq \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w_k|^2. \tag{4}$$

Ideally, cooperation among agents should be restricted to neighbors that belong to the same cluster, i.e., agent k will share data with ℓ only if $w^o_k = w^o_\ell$. However, agents do not know beforehand which neighbors belong to the same cluster. Accordingly, cooperation will be limited to those neighbors that are believed to belong to the same cluster, as identified by $\mathcal{N}_{k,i}$. Motivated by the derivations in [2, 14], each agent k therefore runs the following LMS diffusion strategy:

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

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}(i) \boldsymbol{\psi}_{\ell,i} \tag{6}$$

where the nonnegative combination coefficients in (6) are seen to be time-dependent and should satisfy:

$$a_{\ell k}(i) = 0 \;\text{ for } \ell \notin \mathcal{N}_{k,i}, \quad \sum_{\ell=1}^{N} a_{\ell k}(i) = 1. \tag{7}$$

Observe that the combination coefficient $a_{\ell k}(i)$ is non-zero only if ℓ is believed to belong to the same neighborhood and cluster as agent k. The cluster information is retrieved from matrix E_i, which is updated continuously. The coefficients $a_{\ell k}(i)$ are selected as explained next.

4. SELECTION OF COMBINATION WEIGHTS

We denote the neighborhood set of agent k excluding k itself by $\mathcal{N}_k^-$. We also introduce an $N \times N$ trust matrix F_i; the entry $f_{\ell k}(i)$ of this matrix reflects the amount of trust that agent k has in neighbor $\ell \in \mathcal{N}_k^-$ belonging to its cluster. The entries of F_i are constructed as follows. Agent k first computes the Boolean variable:

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \|\psi_{\ell,i} - w_{k,i-1}\|^2 \leq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $0 < \alpha$ is some threshold value. Subsequently, the trust level $f_{\ell k}(i)$ is smoothed as follows:

$$f_{\ell k}(i) = \nu \times f_{\ell k}(i-1) + (1-\nu) \times b_{\ell k}(i) \tag{9}$$

where the forgetting factor, $0 < \nu < 1$, determines the speed with which trust in neighbor ℓ accumulates. Once $f_{\ell k}(i)$ exceeds 0.5, agent k declares that neighbor ℓ belongs to its cluster and sets the corresponding entry of E_i to one. In other words, at each time i, the entries of E_i are set as follows:

$$e_{\ell k}(i) = \lfloor f_{\ell k}(i) \rceil \tag{10}$$

where the notation $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. The smoothing step (9) provides a useful variation to the protocol proposed in [13]. By using smoothed values for the trust variables, we are able to couple the clustering and inference procedures into a single iterative algorithm rather than run them separately. This is because smoothing helps reduce the influence of erroneous clustering decisions on the accuracy of the inference task. The following listing summarizes the proposed diffusion strategy with clustering.

5. PERFORMANCE ANALYSIS

5.1. Clustering Errors

Motivated by the analyses in [1, 13], we comment briefly on the expected performance of the proposed clustering procedure by examining the behavior of the probabilities of erroneous decisions of types I and II for each agent k, namely, the probabilities that a link between k and one of its neighbors will be either erroneously disconnected (when it should be connected) or erroneously connected (when it should be disconnected):

$$\text{Type-I: } w_k^o = w_\ell^o \text{ and } a_{\ell k}(i) = 0 \tag{11}$$

$$\text{Type-II: } w_k^o \neq w_\ell^o \text{ and } a_{\ell k}(i) \neq 0 \tag{12}$$

Algorithm 1 ATC Diffusion LMS with Clustering.

Initialize $E_{-1} = E$, $w_{k,-1} = 0$, and $F_{-1} = I$. For each $k = 1, \ldots, N$:

for $i \geq 0$ **do**

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

  for $\ell \in \mathcal{N}_k^-$ **do** $\triangleright\ \ell \neq k$

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \|\psi_{\ell,i} - w_{k,i-1}\|^2 \leq \alpha \\ 0, & \text{otherwise} \end{cases}$$

   $f_{\ell k}(i) = \nu \times f_{\ell k}(i-1) + (1-\nu) \times b_{\ell k}(i)$

   $e_{\ell k}(i) = \lfloor f_{\ell k}(i) \rceil$

  end for

  select $a_{\ell k}(i)$ according to (7)

$$w_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}(i)\psi_{\ell,i}$$

end for

for any $\ell \in \mathcal{N}_k$. These probabilities are given by:

$$P_1 = \Pr(\boldsymbol{f}_{\ell k}(i) < 0.5 | w_\ell^o = w_k^o). \tag{13}$$

$$P_2 = \Pr(\boldsymbol{f}_{\ell k}(i) \geq 0.5 | w_\ell^o \neq w_k^o). \tag{14}$$

We are assuming sufficient training time has elapsed so that these probabilities can be assumed to be independent of time. In order to find bounds for P_1 and P_2, we examine the probability distribution of the trust variable $\boldsymbol{f}_{\ell k}(i)$, now treated as a random variable. Using (9), we have:

$$\boldsymbol{f}_{\ell k}(i) = \nu^{i+1}\boldsymbol{f}_{\ell k}(-1) + (1-\nu)\sum_{j=0}^{i} \nu^j \boldsymbol{b}_{\ell k}(i-j) \tag{15}$$

where $\boldsymbol{b}_{\ell k}(i)$ is modelled as a Bernoulli random variable. The assignment of $\boldsymbol{b}_{\ell k}(i)$ to one corresponds to the event described by

$$\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \leq \alpha. \tag{16}$$

We denote respectively the probabilities of true and false assignments (again under the assumption of time-independence) by

$$P_d = \Pr(\boldsymbol{b}_{\ell k}(i) = 1 | w_\ell^o = w_k^o). \tag{17}$$

$$P_f = \Pr(\boldsymbol{b}_{\ell k}(i) = 1 | w_\ell^o \neq w_k^o). \tag{18}$$

These probabilities also satisfy:

$$(1 - P_d) = \Pr(\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 > \alpha | w_\ell^o = w_k^o). \tag{19}$$

$$P_f = \Pr(\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \leq \alpha | w_\ell^o \neq w_k^o). \tag{20}$$

Using arguments similar to [1], it can be argued that approximately the following two time-independent probabilities hold

$$P_1 \leq \frac{1-\nu}{1+\nu} . \frac{P_d(1-P_d)}{(P_d - 0.5)^2}. \tag{21}$$

$$P_2 \leq \frac{1-\nu}{1+\nu} \cdot \frac{P_f(1-P_f)}{(0.5-P_f)^2}. \tag{22}$$

Moreover, arguments similar to [13] indicate that after sufficient iterations and for small enough α and step-sizes,

$$(1-P_d) \leq O(e^{-c_1/\mu_{\max}}) \tag{23}$$

$$P_f \leq O(e^{-c_2/\mu_{\max}}) \tag{24}$$

for some constants $c_1, c_2 > 0$, where the maximum step-size across the agents is denoted by $\mu_{\max}$. It is then seen that the probabilities P_1 and P_2 are expected to approach zero for vanishing step-sizes.

5.2. Learning Curves

The transient mean-square deviation (MSD) of the network at each time instant i is defined by:

$$\text{MSD}_{\text{network}}(i) \triangleq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\|w_k^o - \boldsymbol{w}_{k,i}\|^2. \tag{25}$$

Both types of clustering errors affect the network MSD, but it is clear that errors of type II have the worst effect because of the sharing of data from different models. We define the $N \times N$ true clustering matrix E^o, where each entry is given by

$$e_{\ell k}^o = \begin{cases} 1, & w_\ell^o = w_k^o \text{ and } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

The normalized clustering errors of types I and II by each agent k at time instant i are given, respectively, by

$$v_{1,k}(i) \triangleq \frac{(\mathbb{1} - [E_i]_{:,k})^\mathsf{T} \times ([E^o]_{:,k} - [E_i]_{:,k})}{(n_k - 1)} \tag{27}$$

$$v_{2,k}(i) \triangleq \frac{[E_i]_{:,k}^\mathsf{T} \times ([E_i]_{:,k} - [E^o]_{:,k})}{(n_k - 1)} \tag{28}$$

where $n_k \triangleq |\mathcal{N}_k|$ is the number of agent k's neighbors.

6. SIMULATION RESULTS

We consider a fully connected network with 100 randomly distributed agents. The statistical profile of the noise across the agents is $\sigma_k^2 \in [0.8, 0.22]$, for $k = 1, \ldots, N$. The regressors are of size $M = 2$, zero-mean Gaussian variables, independent in time and space, and have diagonal covariance matrices $R_{u,k} = \lambda_k I_M$ where $\lambda_k \in [0.8, 1.2]$. We chose $\{\mu, \alpha, \nu\} = \{0.05, 0.01, 0.98\}$. The maximum number of neighbors is $n_k = 6$. The agents observe data originating from three different models $C = 3$, each model $w_{\mathcal{C},c}^o \in \mathbb{R}^{M \times 1}$ is generated as follows: $w_{\mathcal{C},c}^o = [w_{r_1}, \ldots, w_{r_M}]^\mathsf{T}$ where $w_{r_m} \in [1, -1]$. The assignment of agents to models is random. We use a uniform combination policy to generate the coefficients $a_{\ell k}(i)$. The simulation results are obtained by averaging over 100 independent experiments. Figure 3 shows the topology of one of these experiments and the estimated

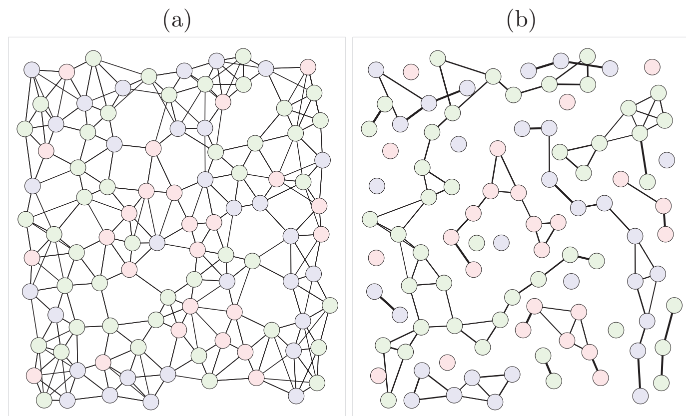

Fig. 3: (a) Network topology. (b) Estimated cluster structure.

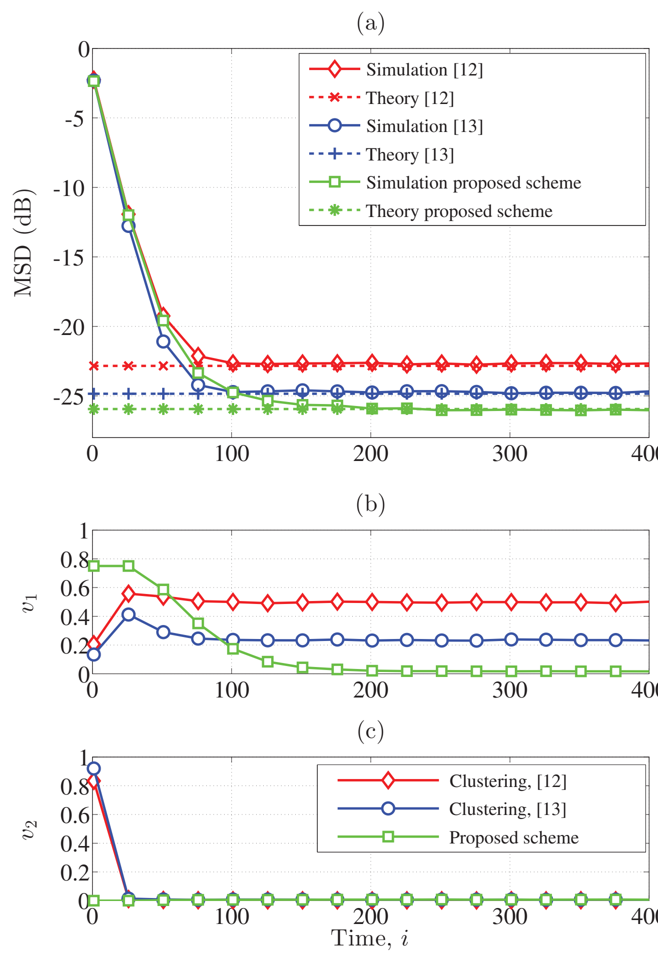

Fig. 4: (a) Network MSD; (b) clustering error v_1 and (c) v_2.

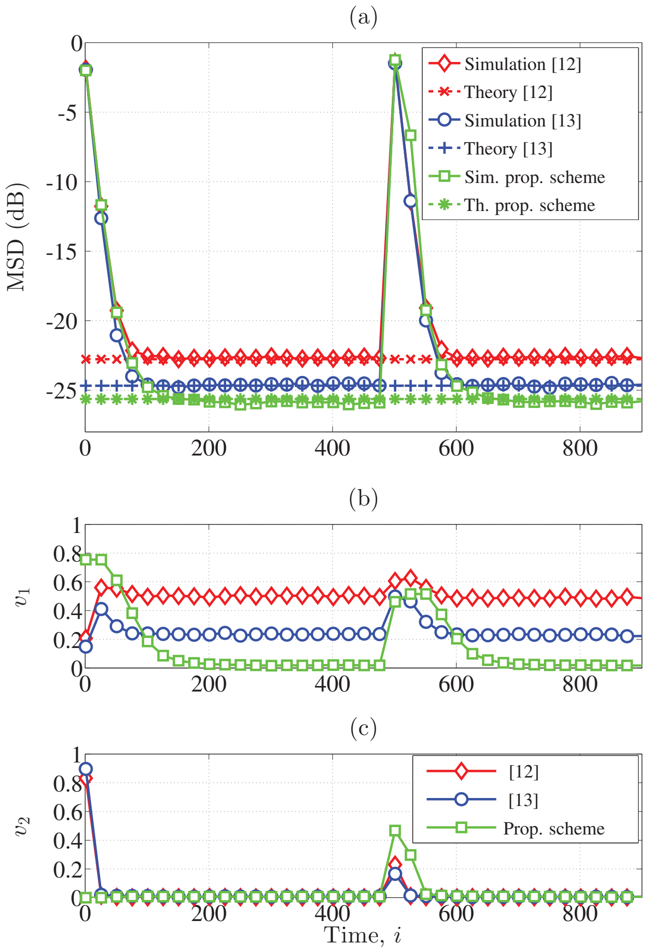

Fig. 5: (a) Network MSD; (b) clustering error v_1 and (c) v_2.

cluster structure. Figure 4a shows the MSD learning curves and the theoretical steady-state values. The normalized clustering errors over the network are shown in Figure 4b. The results are seen to enhance performance in terms of clustering errors. Figure 5 shows results for the case in which the model assignments change at time $i = 500$ for a network of size $N = 50$. This simulation shows the ability of the algorithm to track drifts in the models.

7. CONCLUSION

In this paper we proposed a distributed algorithm that carries out the tasks of estimation and clustering simultaneously with exponentially decaying error probabilities for false decisions. We showed how the agents choose a set of their neighbors to cooperate with and turn off the suspicious links with the remaining neighbors. The simulations illustrate the performance of the proposed strategy and compare with other related works.

REFERENCES

[1] S. Y. Tu and A. H. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, Mar. 2014.

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

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

[4] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive netowrks," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 7223–7227.

[5] 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, 2007.

[6] 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, 2011.

[7] I. Francis and S. Chatterjee, "Classification and estimation of several multiple regressions," *The Annals of Statistics*, vol. 2, no. 3, pp. 558–561, 1974.

[8] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, 1996.

[9] V. Cherkassky and Y. Ma, "Multiple model regression estimation," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 785–798, July 2005.

[10] L. Jacob, F. Bach, and J.-P. Vert, "Clustered multi-task learning: A convex formulation," in *Proc. Neural Inform. Process. Systems. (NIPS)*, Vancouver, Canada, Dec. 2008, pp. 1–8.

[11] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. International Workshop on Cognitive Inform. Process. (CIP)*, Baiona, Spain, May 2012, pp. 1–6.

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

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

[14] 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.