2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2015, BOSTON, USA

ADAPTIVE REGULARIZED DIFFUSION ADAPTATION OVER MULTITASK NETWORKS

Sadaf Monajemi[†] *Saeid Sanei*[*] *Sim-Heng Ong*[‡] *Ali H. Sayed*[§]

[†] NUS Graduate School for Integrative Sciences and Engineering, NUS, Singapore
[*] Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, UK
[‡] Department of Electrical and Computer Engineering, NUS, Singapore
[§] Department of Electrical Engineering, University of California, Los Angeles

ABSTRACT

The focus of this paper is on multitask learning over adaptive networks where different clusters of nodes have different objectives. We propose an adaptive regularized diffusion strategy using Gaussian kernel regularization to enable the agents to learn about the objectives of their neighbors and to ignore misleading information. In this way, the nodes will be able to meet their objectives more accurately and improve the performance of the network. Simulation results are provided to illustrate the performance of the proposed adaptive regularization procedure in comparison with other implementations.

Index Terms— Distributed optimization, adaptive combination and regularization, diffusion LMS.

1. INTRODUCTION

Distributed optimization and learning over networks is an attractive research area with several applications from signal processing and optimization to modeling of biological and social networks [1–4]. Several strategies have been proposed in the literature for distributed processing over networks such as incremental strategies [5], consensus strategies [6, 7] and diffusion strategies [1, 2, 8]. It has been shown that among these strategies, the diffusion algorithm is robust, scalable, and capable of real-time adaptation and learning. Diffusion strategies also have superior performance and stability compared to consensus methods [1, 2, 9]. We therefore focus on the implementation of diffusion strategies in this article. In particular, we examine networks where different clusters of agents may be interested in different objectives [10–15]. In this case, it is important to develop algorithms that enable the agents to continuously learn which of their neighbors belong to the same cluster and which ones are from different clusters.

References [13, 15] study the important case of two objectives where agents receive data from one of two possible models. Reference [16] considers multiple clusters under the assumption that the objectives of adjacent clusters are related to each other and that agents are aware of their clusters. There are also variations of multi-task networks where agents deal with the estimation of different types of parameters [17]; one of the parameters is common to all agents and the second parameter can vary across agents.

In this paper we consider a multitask network consisting of several connected clusters with different objectives. We do not assume that the agents have access to any prior clustering information. In particular, the agents do not know which clusters they belong to. They also do not know the clusters of their neighbors. Moreover, we do not assume prior knowledge about how the objectives of the clusters are related to each other. Therefore, the proposed method is able to handle situations where there are different objectives in the network without interference among the clusters. For this purpose, we propose a multitask learning method that employs Gaussian kernel regularization. In this method, both the combination weights and the regularization coefficients are learned adaptively and continuously. In this way, the agents are able to cooperate only with neighbors that share the same objective.

2. MODELING OF MULTITASK NETWORKS

In this paper, we use plain letters to denote scalars, boldface lowercase letters to denote vectors, and boldface uppercase letters to denote matrices. Furthermore, $\mathcal{N}_k$ represents neighbors of node k, including k.

2.1. Network model

We adopt a multitask problem formulation similar to the one studied in [11,16] with some variation in the cost formulation, as explained further ahead. We consider a connected network consisting of N nodes. Each node k wants to estimate an $M \times 1$ unknown vector $\boldsymbol{\omega}_k^o$ from collected measurements. Each node k has access to a scalar measurement $d_k(i)$ and an $M \times 1$ regression vector $\boldsymbol{x}_k(i)$ at every time instant $i \geq 0$. The data at each node is assumed to be related to the unknown

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

978-1-4673-7454-5/15/$31.00

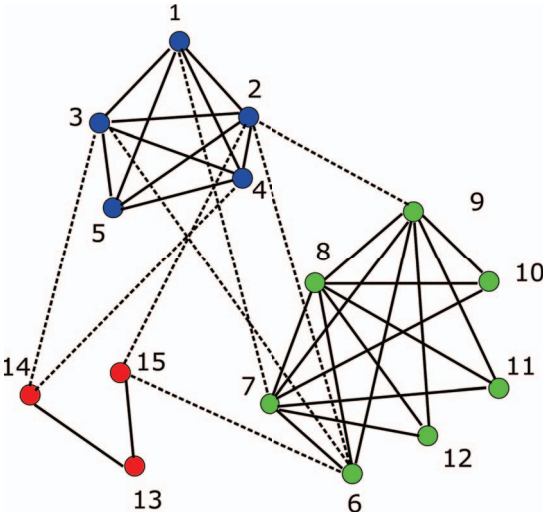

Fig. 1. An example of a multitask network consisting of $N = 15$ nodes and $Q = 3$ clusters. The solid lines are the links between the nodes of the same cluster and the dashed lines represent connections between nodes in different clusters.

parameter vector $\boldsymbol{\omega}_k^o$ via a linear regression model:

$$d_k(i) = \boldsymbol{x}_k^T(i)\boldsymbol{\omega}_k^o + n_k(i), \tag{1}$$

where $n_k(i)$ is the measurement noise at node k and time instant i.

We assume that the network consists of Q different clusters and we write $\mathcal{C}_q$ to represent the set of nodes in cluster q. Each cluster is a collection of nodes that are interested in the same parameter vector:

$$\boldsymbol{\omega}_k^o = \boldsymbol{\omega}_{\mathcal{C}_q}^o, \text{ for all } \quad k \in \mathcal{C}_q. \tag{2}$$

Nodes of different clusters can be connected to each other but nodes do not have prior information about the clusters that their neighbors belong to. This means that during the initial stages of adaptation, nodes do not know whether their neighbors are following the same objective as them. An example of such a network is shown in Figure 1. In this network, the total number of agents is $N = 15$ and there are $Q = 3$ clusters. The solid lines represent links between nodes of the same cluster while the dashed lines represent connections between nodes in different clusters. In general, processing data from neighbors without considering their clusters can lead to adverse performance. In the next section we formulate an optimization problem for such a network and propose an adaptive method to solve this problem.

3. PROBLEM FORMULATION

In order to solve the estimation problem over the multitask network, we assign a local cost function, $J_k(\boldsymbol{\omega}_{\mathcal{C}(k)})$, to each node k where $\mathcal{C}(k)$ represents the cluster that node k belongs to:

$$J_k(\boldsymbol{\omega}_{\mathcal{C}(k)}) = \mathbb{E}\left\{|d_k(i) - \boldsymbol{x}_k^T(i)\boldsymbol{\omega}_{\mathcal{C}(k)}|^2\right\}. \tag{3}$$

The nodes that belong to the same cluster have a mutual interest in estimating the same parameter vector. Therefore, in order to encourage this type of cooperation, we can use an appropriate regularization to promote similarities among agents with similar objectives. There are several possible regularization terms that can be used for this particular purpose. Here, following [11, 16], we consider the squared Euclidean distance as a similarity regularizer:

$$\Delta(\boldsymbol{\omega}_{\mathcal{C}(k)} - \boldsymbol{\omega}_{\mathcal{C}(\ell)}) \triangleq \|\boldsymbol{\omega}_{\mathcal{C}(k)} - \boldsymbol{\omega}_{\mathcal{C}(\ell)}\|^2, \tag{4}$$

where $\|\boldsymbol{\omega}_{\mathcal{C}(k)} - \boldsymbol{\omega}_{\mathcal{C}(\ell)}\|$ is the Euclidean distance between the parameter vectors for nodes ℓ and k. Combining equations (3) and (4) results in the regularized global cost function:

$$J^{\text{glob}}(\boldsymbol{\omega}_{\mathcal{C}(1)}, ..., \boldsymbol{\omega}_{\mathcal{C}(Q)}) = \sum_{k=1}^{N} \mathbb{E}\left\{|d_k(i) - \boldsymbol{x}_k^T(i)\boldsymbol{\omega}_{\mathcal{C}(k)}|^2\right\}$$
$$+ \beta \sum_{k=1}^{N} \sum_{\ell \in \mathcal{N}_k} \gamma_{\ell k} \|\boldsymbol{\omega}_{\mathcal{C}(k)} - \boldsymbol{\omega}_{\mathcal{C}(\ell)}\|^2, \tag{5}$$

where $\beta \geq 0$ is a strength parameter. Moreover, the weights $\gamma_{\ell k} \geq 0$ adjust the role of the regularization term between the two nodes k and ℓ and they satisfy the conditions:

$$\sum_{\ell=1}^{N} \gamma_{\ell k} = 1, \text{ and } \gamma_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{6}$$

It is important to note that similar to the case studied in [16], the role of adding the second term on the right hand side of (5) is to promote similarity among nodes with similar objectives. However, unlike [16], here the nodes have no prior information about the objectives of their neighbors. Therefore, the internal summation in this term is over *all* the neighbors of node k. Consequently, the regularization weights $\gamma_{\ell k}$ must be adjusted such that they allocate more weight to neighbors with similar objectives while giving less weight to the neighbors from different clusters.

In the following section, we propose an adaptive diffusion strategy that enables agents to learn the regularization coefficients, as well as the combination weights, in such a way that they end up assigning relatively larger weights to the neighbors with similar objectives.

4. DIFFUSION ADAPTATION STRATEGY

Without loss of generality, we employ the adapt-then-combine (ATC) diffusion algorithm to solve the optimization problem in equation (5) due to its superior performance even in comparison to consensus strategies [1]. The result of applying

ATC to equation (5) leads to the following distributed strategy:

$$\boldsymbol{\psi}_k(i) = \boldsymbol{\omega}_k(i-1) + \mu_k \boldsymbol{x}_k(i)[d_k(i) - \boldsymbol{x}_k^T(i)\boldsymbol{\omega}_k(i-1)] + \mu_k\beta \sum_{\ell\in\mathcal{N}_k} \gamma_{\ell k}(i)\,(\boldsymbol{\omega}_\ell(i-1) - \boldsymbol{\omega}_k(i-1)), \tag{7}$$

$$\boldsymbol{\omega}_k(i) = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}(i)\boldsymbol{\psi}_\ell(i), \tag{8}$$

where $\mu_k > 0$ is the step-size parameter used by node k, and the coefficients $a_{\ell k}(i)$ are non-negative entries of an $N \times N$ combination matrix $\boldsymbol{A}_i$ at time instant i. It is important to note that since $\boldsymbol{A}_i$ has to be left-stochastic, we have:

$$\boldsymbol{A}_i^T \mathbb{1} = \mathbb{1},\ \ a_{\ell k}(i) = 0 \text{ if } \ell \notin \mathcal{N}_k, \tag{9}$$

where $\mathbb{1}$ is an $N \times 1$ vector with all entries equal to one. Observe in (7) that we are allowing the regularization coefficients $\gamma_{\ell k}(i)$ to vary with time because they will be adapted as well.

4.1. Selection of Regularization Weights

As indicated previously, selection of the regularization coefficients $\gamma_{\ell k}$ in equation (5) has a significant impact on the performance of the network. These coefficients must be estimated in an adaptive manner so that agents can be clustered more accurately. Now since the nodes do not have prior information about the clusters of their neighbors, the weights must be estimated in such a way that they allocate higher weight to neighbors sharing similar objectives. In other words, the regularization penalty term in equation (5) must be omitted when the objectives of nodes k and ℓ are not similar. Therefore, the regularization weights $\gamma_{\ell k}(i)$ must be inversely, but not necessarily linearly, proportional to the distance between the objectives of two nodes, i.e., $\|\boldsymbol{\omega}_k^o - \boldsymbol{\omega}_\ell^o\|^2$.

Among several possible adaptive regularization terms that have been used in the literature, we select $\gamma_{\ell k}(i)$ proportional to $\exp(-\|\boldsymbol{\omega}_k^o - \boldsymbol{\omega}_\ell^o\|^2/h)$ [18, 19]. It can be seen that this method is analogous to Gaussian kernel regularization with sufficient flexibility when $h > 0$:

$$\gamma_{\ell k}(i) = \frac{\exp(-\|\boldsymbol{\omega}_k^o - \boldsymbol{\omega}_\ell^o\|^2/h)}{\sum_{n\in\mathcal{N}_k} \exp(-\|\boldsymbol{\omega}_k^o - \boldsymbol{\omega}_n^o\|^2/h)} \tag{10}$$

It is still not feasible to evaluate the regularization weights based on equation (10). This is because the nodes do not know the true objectives $\boldsymbol{\omega}_k^o$ and $\boldsymbol{\omega}_\ell^o$. Therefore, we replace these objectives by the best available estimates at each time instant and reformulate the weights as:

$$\gamma_{\ell k}(i) \approx \begin{cases} \frac{\exp(-\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\omega}_\ell(i-1)\|^2/h)}{\sum_{n\in\mathcal{N}_k} \exp(-\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\omega}_n(i-1)\|^2/h)}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Algorithm 1 Regularized ATC Diffusion LMS for clustered multitask networks with adaptive regularization weights.

Require: $\omega_{k,0} = \psi_{k,0} = 0, \gamma_{\ell k}(0) = 0$ for all k

for $i \geq 1$ **do**

$$\gamma_{\ell k}(i) \approx \begin{cases} \frac{\exp(-\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\omega}_\ell(i-1)\|^2/h)}{\sum_{n\in\mathcal{N}_k} \exp(-\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\omega}_n(i-1)\|^2/h)}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$$

$$\boldsymbol{\psi}_k(i) = \boldsymbol{\omega}_k(i-1) + \mu_k \boldsymbol{x}_k(i)[d_k(i) - \boldsymbol{x}_k^T(i)\boldsymbol{\omega}_k(i-1)] + \mu_k\beta \sum_{\ell\in\mathcal{N}_k} \gamma_{\ell k}(i)\,(\boldsymbol{\omega}_\ell(i-1) - \boldsymbol{\omega}_k(i-1))$$

$$a_{\ell k}(i) \approx \begin{cases} \frac{\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\psi}_\ell(i)\|^{-2}}{\sum_{n\in\mathcal{N}_k} \|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\psi}_n(i)\|^{-2}}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$$

$\boldsymbol{\omega}_k(i) = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}(i)\boldsymbol{\psi}_\ell(i)$

end for

4.2. Estimation of Combination Weights

In order to estimate the combination weights, we follow the same procedure developed in [11, 13] and introduce the instantaneous MSD of the network at time i:

$$\text{MSD}(i) \triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\|\widetilde{\boldsymbol{\omega}}_k(i)\|^2, \tag{12}$$

where $\widetilde{\boldsymbol{\omega}}_k(i) \triangleq \boldsymbol{\omega}_k^o - \boldsymbol{\omega}_k(i)$ is the error vector at node k at time instant i. Then, the combination coefficients $a_{\ell k}(i)$ can be obtained by solving the optimization problem:

$$\min_{\boldsymbol{A}_i} \text{MSD}(i) \tag{13}$$

subject to (9). It was shown in [13] that the optimal solution can be approximated by:

$$a_{\ell k}(i) \approx \begin{cases} \frac{\|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\psi}_\ell(i)\|^{-2}}{\sum_{n\in\mathcal{N}_k} \|\boldsymbol{\omega}_k(i-1) - \boldsymbol{\psi}_n(i)\|^{-2}}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

One important conclusion derived from equation (14) is that the combination coefficient $a_{\ell k}(i)$ is inversely proportional to the distance between the estimate of node k and the intermediate estimate $\boldsymbol{\psi}_\ell(i)$ of node ℓ. In other words, this combination approach enables the agents to continuously learn about the objective of their neighbors so that they can distinguish between useful and misleading information. This method helps the nodes of multitask networks to acquire an effective cooperative strategy. Estimating both combination and regularization weights in this manner results in an adaptive multitask diffusion algorithm, which helps the agents benefit from cooperation by ignoring misleading information.

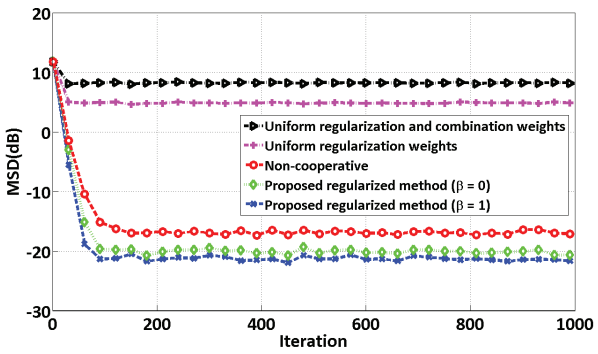

Fig. 2. Performance of the network for different learning strategies. The MSD curves are shown for different diffusion strategies.

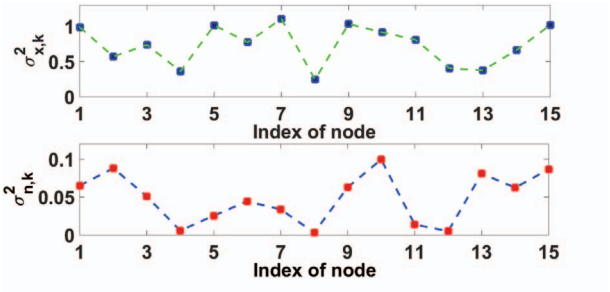

Fig. 3. Variances of the network input and noise for each node of the network.

5. SIMULATION RESULTS

We present a numerical example in order to illustrate the behavior of the proposed algorithm over a multitask network. In this example, we consider the network of 15 nodes, i.e., $N = 15$, shown in Figure 1. As can be seen in the figure, the network is divided into 3 clusters: $C_1 = \{1, 2, 3, 4, 5\}$, $C_2 = \{6, 7, 8, 9, 10, 11, 12\}$, and $C_3 = \{13, 14, 15\}$. The parameter vector for each cluster is two-dimensional and chosen as $\boldsymbol{\omega}^o_{C_1} = [0.5, -0.4]^T$, $\boldsymbol{\omega}^o_{C_2} = [-1, 3]^T$, and $\boldsymbol{\omega}^o_{C_3} = [5.19, 2.81]^T$. Moreover, the regression input signals $\boldsymbol{x}_k(i)$ are 2×1 zero-mean Guassian random vectors with covariance matrices $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_M$ where $\sigma^2_{x,k}$ is shown in Figure 3. Additionally, the measurement noises $n_k(i)$ are zero-mean random variables with a Guassian distribution and their variances $\sigma^2_{n,k}$ are also shown in Figure 3.

The regularization weights $\gamma_{\ell k}(i)$ and the combination weights $a_{\ell k}(i)$ are estimated using (11) and (14), respectively. The results of different strategies have been averaged over 100 Monte-Carlo runs for $\mu = 0.1$, as shown in Figure 2. The performance of the proposed method is compared to other learning methods: (a) the non-cooperative algorithm, where each node of the network attempts to estimate the required parameter vector without using any information from other nodes. In this case, a cluster can be assigned to each node and we set: $\boldsymbol{A}_i = \boldsymbol{I}_N$, $\beta = 0$; (b) the regularized multitask algorithm with uniform regularization weights, where the nodes employ a diffusion strategy with uniform regularization weights: $\gamma_{k\ell} = |\mathcal{N}_k|^{-1}$; (c) the regularized multitask algorithm with uniform combination and regularization weights, where both the regularization weights $\gamma_{k\ell}(i)$ and combination weights $a_{k\ell}(i)$ are set equal to $|\mathcal{N}_k|^{-1}$; (d) the regularized multitask algorithm with $\beta = 0$, where the nodes of the network solve the multitask problem cooperatively and with adaptive weights, but the regularization term is omitted.

Comparing the five learning strategies, we can see that the proposed regularized method with adaptive weights enables the nodes of the network to achieve superior performance. On the other hand, in cases where the regularization and combination weights are uniform and have not been estimated adaptively, the nodes are unable to distinguish misleading information from beneficial data. Therefore, most of the nodes are not successful in estimating their own objectives and their performance is even worse than the case where there is no cooperation. In fact, this result clearly reveals the challenge of cooperation in multitask networks since it shows that in the cases where the nodes allocate equal weights to all of their neighbors without considering their objectives, the performance of the network may be even worse than the case where the nodes do not cooperate at all. Finally, comparing the performance of the proposed method for two regularization factors of $\beta = 0$ and $\beta = 1$ shows that adding the regularization term can be beneficial.

6. CONCLUSION

In this paper, we have considered the multitask estimation problem where the nodes of the network have different objectives. We developed a regularized adaptive learning algorithm involving Gaussian kernels, which guides the nodes to learn the beneficial information and ignore the misleading data received over time.

7. REFERENCES

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

[2] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," *in* Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., Elsevier, 2014*, vol. 2, pp. 329–408.

[4] S. Monajemi, S. Sanei, and S.H. Ong, "Advances in bacteria motility modelling via diffusion adaptation," in Proc. EUSIPCO, pp. 2335–2339.

[5] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

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

[8] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

[9] 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.

[10] C-K. Yu, M. van der Schaar, and A. H. Sayed, "Cluster formation over adaptive networks with selfish agents," in Proc. EUSIPCO*, Marrakech, Morocco*, September 2013, pp. 1–5.

[11] 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.

[12] J. Chen and C. Richard, "Performance analysis of diffusion LMS in multitask networks," in *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Saint Martin*, December 2013, pp. 137–140.

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

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

[15] 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.

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

[17] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5382–5397, 2014.

[18] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.

[19] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*, pp. 144–158. Springer, 2003.