CONVEX COMBINATION OF DIFFUSION STRATEGIES OVER DISTRIBUTED NETWORKS

Danqi Jin, Jie Chen, Jingdong Chen

Centre of Intelligent Acoustics and Immersive Communications
School of Marine Science and Technology, Northwestern Polytechinical University, China
danqijin@mail.nwpu.edu.cn dr.jie.chen@ieee.org jingdongchen@ieee.org

ABSTRACT

Diffusion adaptation is a useful strategy for distributed estimation over networks. Though several information fusion strategies for the diffusion adaptation have been proposed in the literature, it can be restrictive to use a single strategy especially for networks operating in non-stationary environments. Inspired by the convex combination of adaptive filters, in this paper we propose to benefit the performance of two distinct strategies by appropriately combining their fusion coefficients. The combination coefficient on each node is determined by minimizing the overall squared estimation error in a local and online manner. Simulation results highlight favorable properties of the proposed combination scheme, with both static and dynamic fusion components.

Index Terms— Distributed optimization, diffusion strategy, convex combination, adaptive fusion strategy.

1. INTRODUCTION

Distributed adaptation algorithms endow networks with the ability to estimate and track unknown parameters from streaming data in a collaborative manner. Among various strategies [1–6], diffusion adaptation [5, 7] is an efficient strategy that is particularly attractive due to its enhanced adaptation performance and wider stability ranges [8]. Diffusion-based algorithms have been extensively studied, in respect of adaptation algorithms on agents, including diffusion LMS [9, 10], diffusion APA [11], diffusion Kalman filtering [12], diffusion RLS [13], and in respect of cooperation strategies among agents [14–16].

A typical diffusion-based adaptation algorithm consists of an adaptation step and a combination step. In the combination step, each agent collects intermediate estimates from its neighbors and performs a fusion of the collected estimates. Properly selecting the fusion weights can be crucial to guarantee an enhanced performance of a diffusion-based algorithm. While finding an optimal setting of these weights can be non-trivial, several empirical strategies with fixed coefficients have been proposed, including averaging rule, metropolis rule, relative-degree rule, etc [5]. In addition to these static strategies, adaptive algorithms have also been derived in consideration of noise levels across the network [17] or parameter relations among agents [15]. However, it is difficult to know the best algorithm to use a priori, and it is also difficult to have an algorithm that always outperforms the others, especially for networks operating in time-varying environments. It is thus important to find a way to address the limitation of using a single combination strategy.

Combination of adaptive filters inspired us with a solution to this issue. Recall that utilizing convex combination [18,19] or affine combination [20] of adaptive filters with diversity, the resulting filters possess the advantages of all component filters. Generally, the combination schemes are used to facilitate the selection of filter parameters, to increase robustness against the unknown environment, as well as to possibly enhance performance beyond the range of each component [21]. Over the past years, several algorithms have been proposed to adapt the combination coefficients of the component filters, such as cvx-LMS [18] and cvx-PN-LMS [22]. In this paper, we propose a convex combination algorithm for fusion coefficients in diffusion strategies. Each agent is designed to run two diverse diffusion strategies and combine their estimates to generate the final estimates. The time-varying combination coefficients are obtained by minimizing the overall squared instantaneous errors. Simulation results show that the proposed algorithm endows the networks with a significantly enhanced performance in the learning process.

Notation. Normal font x denotes scalars. Boldface small letters $\boldsymbol{x}$ and capital letters $\boldsymbol{X}$ denote column vectors and matrices, respectively. The superscript $(\cdot)^\top$ denotes the transpose operator. The mathematical expectation is denoted by $\mathbb{E}\{\cdot\}$. $\mathcal{N}_k$ denotes the neighbors of node k, including k.

2. NETWORK MODEL AND DIFFUSION LMS

2.1. Network Model

Consider a connected network consisting of N agents. The problem is to estimate unknown parameter vectors $\boldsymbol{w}_k^\star$ of length $L \times 1$ at each agent k. Agent k has access to temporal measurement sequences $\{d_{k,n}, \boldsymbol{x}_{k,n}\}$, where $d_{k,n}$ denotes a

The work of Jie Chen was supported in part by NSFC grant 61671382. The work of Jingdong Chen was supported in part by NSFC grant 61425005.

reference signal, and $\boldsymbol{x}_{k,n}$ denotes an $L \times 1$ regression vector with a covariance matrix $\boldsymbol{R}_{x,k} > 0$. The data at agent k at time instant n are characterized by the linear model:

$$d_{k,n} = \boldsymbol{x}_{k,n}^\top \boldsymbol{w}_k^\star + z_{k,n}, \tag{1}$$

where $z_{k,n}$ is a zero-mean, stationary, i.i.d. additive noise with variance $\sigma_{z,k}^2$, and independent of any other signals. To determine the unknown parameter vector $\boldsymbol{w}_k^\star$, we consider the following mean-square-error (MSE) cost at each agent k:

$$J_k(\boldsymbol{w}) = \mathbb{E}\left\{|d_{k,n} - \boldsymbol{x}_{k,n}^\top \boldsymbol{w}|^2\right\}. \tag{2}$$

It is seen from (1) that $J_k(\boldsymbol{w})$ is minimized at $\boldsymbol{w}_k^\star$. For single-task problems, each agent in the network estimate the same parameter vector, while for multi-task problems, agents may estimate distinct parameter vectors.

Note that the combination scheme to be presented will fit in any diffusion adaptation algorithms. We use the linear model (1) and MSE cost (2) for an ease of presentation of main idea of this work.

2.2. Diffusion LMS

Diffusion LMS strategies for the distributed estimation of $\boldsymbol{w}_k^\star$ were derived in [5, 6, 9, 10] by seeking the minimizer of the following aggregate cost function:

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

in a cooperative manner in order to improve estimation accuracy. Our work will be presented with the adapt-then-combine (ATC) diffusion LMS without raw data exchange, i.e.,

$$\begin{cases} \boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} + \mu_k \, \boldsymbol{x}_{k,n} \, (d_{k,n} - \boldsymbol{x}_{k,n}^\top \, \boldsymbol{w}_{k,n}) \\ \boldsymbol{w}_{k,n+1} = \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \boldsymbol{\psi}_{\ell,n+1} \end{cases} \tag{4}$$

where $\{a_{lk}\}$ are a group of coefficients satisfying:

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k. \tag{5}$$

Namely fusion coefficients $\{a_{\ell k}\}$ form to a left-stochastic matrix $\boldsymbol{A}$. Note that the steps in (4) can be written in a compact form by:

$$\boldsymbol{w}_{k,n+1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \left[\boldsymbol{w}_{\ell,n} + \mu_\ell \boldsymbol{x}_{\ell,n} \big(d_{\ell,n} - \boldsymbol{x}_{\ell,n}^\top \, \boldsymbol{w}_{\ell,n}\big)\right]. \tag{6}$$

2.3. Motivation of combining multiple matrices $\boldsymbol{A}$

Coefficient matrix $\boldsymbol{A}$ plays a crucial role in affecting the performance of a diffusion LMS algorithm. Though we can select left-stochastic matrix $\boldsymbol{A}$ freely, either in a fixed or adaptive manner, it can be restrictive to use a single combination matrix $\boldsymbol{A}$ for the following reasons:

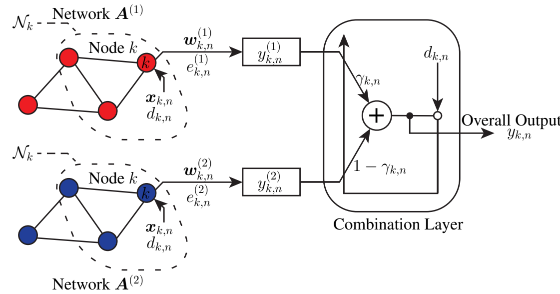


Fig. 1. Illustration of combination of two diffusion strategies.

- On one hand, it is difficult to know the best fusion strategy to use a priori, especially for networks operating in time-varying environments. For instance, an averaging rule is frequently used for single-task networks and usually outperforms the non-cooperative strategy with $\boldsymbol{A} = \boldsymbol{I}_N$. However, if parameters on agents drift to a multitask case at some instant, the non-cooperative strategy can then become preferable since an averaging rule leads to a large estimation bias.
- On the other hand, it is often difficult to have an adaptive fusion coefficient adjusting algorithm that outperforms the others in all cases, since each algorithm has its own working hypothesis, and has parameters to adjust with respect to applications.

We thus propose to consider several combination strategies as components, then adjust strength of each component to achieve a properly combined strategy. This component-based combination scheme has been successfully used in convex combination of adaptive filters, as well as in multi-kernel learning [23, 24].

3. CONVEX COMBINATION OF DIFFUSION STRATEGIES

In this work, we consider the convex combination of two diffusion LMS strategies with distinct component fusion matrices, while an extension to multiple components is possible.

The combination scheme consists of two concurrent adaptive layers: adaptation with individual diffusion strategies and adaptation of the combination layer. The input of the combination layer is two groups of estimations of the same network obtained by running two distinct diffusion strategies individually. This setting is also equivalent to combine two topologically identical networks with different diffusion strategies, with the associated combination matrices denoted by $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ respectively [1], as illustrated in Fig. 1. For networks $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$, the corresponding agents have access to the identical input and reference signals, then produce two group-

[1]Without ambiguity, $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ are also used to denote the networks running corresponding diffusion strategies.

s but individual estimations of the optimal weight vectors. We then associate combination coefficients $\gamma_{k,n} \in [0,1]$, and $1-\gamma_{k,n}$ respectively for two components at agent k at time instant n. In total, a network of N agents is then associated with N pairs of convex combination coefficients. The goal of the combination layer is to learn which diffusion strategy performs better at any particular instant n at each agent, to assign them with weights that optimizes the overall network performance.

We denote the (ℓ,k)-th entries of left-stochastic matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ as $a_{\ell k}^{(1)}$ and $a_{\ell k}^{(2)}$ respectively. Then we have:

$$\boldsymbol{w}_{k,n+1}^{(i)} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}^{(i)} \left[\boldsymbol{w}_{\ell,n}^{(i)} + \mu_\ell^{(i)} \boldsymbol{x}_{\ell,n}\big(d_{\ell,n} - \boldsymbol{x}_{\ell,n}^\top \boldsymbol{w}_{\ell,n}^{(i)}\big)\right] \tag{7}$$

$$y_{k,n}^{(i)} = \boldsymbol{x}_{k,n}^\top \boldsymbol{w}_{k,n}^{(i)} \tag{8}$$

$$e_{k,n}^{(i)} = d_{k,n} - \boldsymbol{x}_{k,n}^\top \boldsymbol{w}_{k,n}^{(i)} \tag{9}$$

with $i=1$ and 2, denoting estimation quantities obtained by using two diffusion strategies with $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ respectively. Recall that $\boldsymbol{w}_{k,n+1}^{(i)}$ is the estimate of the system coefficient $\boldsymbol{w}_k^\star$ at time instant $n+1$, $e_{k,n}^{(i)}$ is a priori output estimation error of agent k at instant n, and $y_{k,n}^{(i)}$ is the estimation of the reference signal at time instant n.

By combining the estimations of two diffusion strategies at each agent k, we have the overall output $y_{k,n}$, overall estimation error $e_{k,n}$ and the overall system coefficients $\boldsymbol{w}_{k,n}$ at the combination layer for each agent k defined as follows:

$$y_{k,n} = \gamma_{k,n} y_{k,n}^{(1)} + (1-\gamma_{k,n}) y_{k,n}^{(2)} \tag{10}$$

$$e_{k,n} = \gamma_{k,n} e_{k,n}^{(1)} + (1-\gamma_{k,n}) e_{k,n}^{(2)} \tag{11}$$

$$\boldsymbol{w}_{k,n} = \gamma_{k,n} \boldsymbol{w}_{k,n}^{(1)} + (1-\gamma_{k,n}) \boldsymbol{w}_{k,n}^{(2)}, \tag{12}$$

corresponding to convex combination of estimations on each agent, with $\gamma_{k,n}\in[0,1]$. To avoid using non-natural hard-thresholding operations for $\gamma_{k,n}$ to keep the convex constraint, we utilize a nonlinear sigmoid function to calculate $\gamma_{k,n}$:

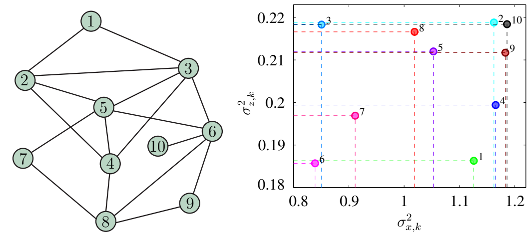
$$\gamma_{k,n} = \frac{1}{1+e^{-\alpha_{k,n}}}, \tag{13}$$

where $\alpha_{k,n}$ is an introduced auxiliary parameter.

Now the problem boils down to the selection of parameter $\gamma_{k,n}$, or equivalently $\alpha_{k,n}$, based on certain criterions. We consider the minimum mean-square-error (MMSE) criterion. The MSE of the whole network at the combination layer at instant n is defined as:

$$J_n^{\text{net}} = \frac{1}{2}\sum_{k=1}^{N} \mathbb{E}\left\{e_{k,n}^2\right\}, \tag{14}$$

which is the summation of the MSE over all agents. Then we choose $\gamma_{k,n}$ by minimizing (14). Observe from (11) that



(a) (b)

Fig. 2. Network topology and associated input variances and noise variances. (a) Network topology; (b) Agent input and noise variances.

the direct solution of $\gamma_{k,n}$ would include an unsolvable term $\mathbb{E}\{e_{k,n}^{(1)} e_{k,n}^{(2)}\}$. To avoid this problem, we adopt an adaptive scheme for the solution of $\gamma_{k,n}$. Due to the introduction of the auxiliary parameter $\alpha_{k,n}$, we adopt an iteration for $\alpha_{k,n}$ instead of updating $\gamma_{k,n}$ directly. Using a stochastic gradient search to minimize (14), and approximating the expectation terms with instantaneous values, it leads to the power-normalized LMS (PN-LMS) iteration:

$$\begin{aligned}\alpha_{k,n+1} &= \alpha_{k,n} - \frac{\mu_{\alpha_k}}{\varepsilon + p_{k,n}} \frac{\partial J_n^{\text{net}}}{\partial \alpha_{k,n}} \\ &\approx \alpha_{k,n} + \frac{\mu_{\alpha_k}}{\varepsilon + p_{k,n}} \gamma_{k,n}\left(1-\gamma_{k,n}\right) e_{k,n} \\ &\quad \times \boldsymbol{x}_{k,n}^\top\big(\boldsymbol{w}_{k,n}^{(1)} - \boldsymbol{w}_{k,n}^{(2)}\big),\end{aligned} \tag{15}$$

where ε is a small positive number to ensure non-zero division, μ_{α_k} is a positive step-size, and $p_{k,n}$ is a low-pass filtered estimation of the power of $\boldsymbol{x}_{k,n}^\top\big(\boldsymbol{w}_{k,n}^{(1)} - \boldsymbol{w}_{k,n}^{(2)}\big)$:

$$p_{k,n} = \eta\, p_{k,n-1} + (1-\eta)\big[\,\boldsymbol{x}_{k,n}^\top(\boldsymbol{w}_{k,n}^{(1)} - \boldsymbol{w}_{k,n}^{(2)})\,\big]^2, \tag{16}$$

with η being a temporal smoothing factor. We utilize power normalization trick [22] in (15) to simplify the selection of step-size μ_{α_k}. Also, we observe that the adaptation (15) would virtually stop if $\alpha_{k,n}$ is allowed to grow unbounded, as $\gamma_{k,n}$ will get close to 0 or 1, which makes the term $\gamma_{k,n}\big(1-\gamma_{k,n}\big)$ be 0. To alleviate this problem, auxiliary parameter $\alpha_{k,n}$ is restricted to be in a symmetric interval $[-\alpha^+,\alpha^+]$, which ensures a minimum level of adaptation. A common choice in the literature is to set $\alpha^+=4$ [21].

4. SIMULATION RESULTS

Now we present simulation results to illustrate the proposed combination scheme. Consider a non-stationary system identification scenario with $\boldsymbol{w}_k^\star$ varied over time. The network was consisted of $N=10$ agents with connection topology depicted in Fig. 2. The regressors were generated from multivariate Gaussian distribution with zero mean and covariance

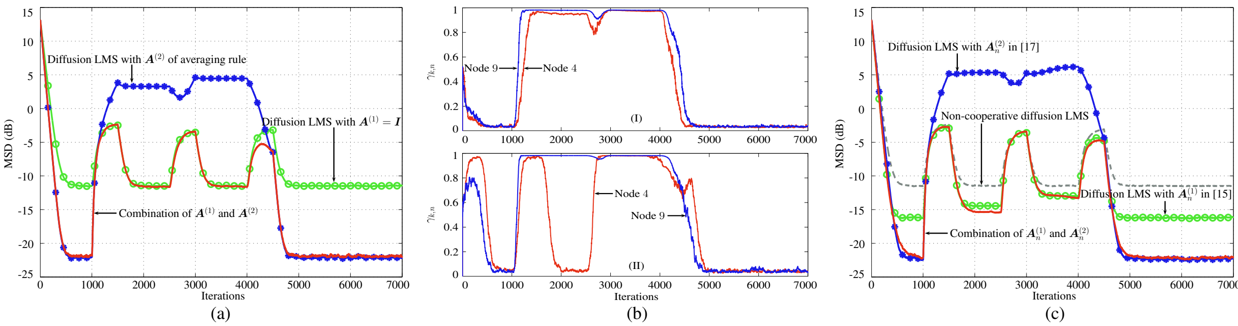


Fig. 3. Simulation results of experiments described in Sec. 4. (a) Network MSD learning curves associated to the combined strategy and two fixed component fusion matrices; (b) The evolutions of convex combination coefficients $\gamma_{k,n}$ of agent 4 and 9 in experiment 1 (top) and experiment 2 (bottom); (c) Network MSD learning curves associated to the combined strategy and two dynamic fusion matrices.

matrix $\boldsymbol{R}_k = \sigma_{x,k}^2 \boldsymbol{I}_L$, with $L = 50$. The noise signals were generated from Gaussian distributions $\mathcal{N}(0, \sigma_{z,k}^2)$. Besides, $\sigma_{x,k}^2$ and $\sigma_{z,k}^2$ of each agent generated randomly are depicted in Fig. 2. The evolutions of system coefficients $\boldsymbol{w}_k^\star$ were divided into four stationary stages and three transient episodes. For stationary stages, the system coefficients $\boldsymbol{w}_k^\star$ were generated from standard Gaussian distribution. During stationary stages, we set $\boldsymbol{w}_k^\star$ of each agent so that from time instant $n = 1$ to 1000 and from instant $n = 4500$ to 7000, the whole network pursued the same target, while from instant $n = 1500$ to 2500 and from $n = 3000$ to 4000, the network split to pursue 2 and 3 targets respectively. The fragments between two adjacent stationary stages were so-called transient episodes, which were designed by using linear interpolation over 500 time instants. We consider two experiments to validate the combination scheme and simulation results are obtained by averaging over 100 Monte Carlo runs. In addition, in all experiments, μ_{α_k} was set to 4, ε was set to 0.05, and η was set to 0.95.

In the first experiment, we consider the combination of two fixed fusion matrices: non-cooperative ATC strategy with $\boldsymbol{A}^{(1)} = \boldsymbol{I}$ and ATC strategy with a matrix $\boldsymbol{A}^{(2)}$ defined by the averaging rule. The resulting MSD learning curves are illustrated in Fig. 3(a). Besides, the evolutions of the combination coefficients of agents 4 and 9 are depicted in Fig. 3(b). We observe from Fig. 3(a) that the ATC strategy with matrix $\boldsymbol{A}^{(2)}$ outperforms ATC strategy with $\boldsymbol{A}^{(1)}$ from instant $n = 1$ to 1100 and from $n = 4400$ to 7000, since the whole network pursues the same target, cooperation between agents would be beneficial. While from instant $n = 1100$ to 4400, information fusion matrix $\boldsymbol{A}^{(2)}$ with the averaging rule leads to a poor performance due to large bias when estimating distinct targets. As expected, the proposed convex combination scheme results in a MSD learning curve approaching the best of each component strategies at different stages. Namely, it behaves in a similar way to the non-cooperative algorithm when estimating multiple targets, while approaches the strategy with $\boldsymbol{A}^{(2)}$ when estimating the same target. Besides, the evolutions of combination coefficients in Fig. 3(b) coincide with the MSD behavior of combination scheme. Large weights are assigned to put emphasize on the use of $\boldsymbol{A}^{(1)}$ from $n = 1100$ to 4400, and on the use of $\boldsymbol{A}^{(2)}$ for the other instants.

We then combine two adaptive information fusion matrices in the second experiment, with matrix $\boldsymbol{A}_n^{(1)}$ derived in [15] and matrix $\boldsymbol{A}_n^{(2)}$ derived in [17], where subscript n is added to emphasize the time-varying property of these fusion matrices. The MSD learning curves are illustrated in Fig. 3(c), and the evolutions of the combination coefficients are depicted in the bottom subfigure of Fig. 3(b). Similar to the first experiment, the proposed combination strategy successfully benefits the performance of each component fusion matrix via an appropriate combination of the two dynamic fusion matrices. Interestingly, from time instant $n = 1900$ to 2500, the combined curve outperforms each component. The behavior of the combination coefficient on agent 4 during this stage is not coincident with intuition, and this particular behavior however may contribute to the positive effect on the performance. Similar favorable property is also reported in the study of the convex combination of classical adaptive filters. These results show the effectiveness of the proposed strategy when applied to dynamic fusion matrices.

5. CONCLUSIONS

In this paper, we propose a scheme for the convex combination of two distributed diffusion strategies. By assigning a group of convex combination coefficients to each nodes of the network, and by minimizing the MSE of the whole network, as well as utilizing the adaptation scheme to adjust the convex combination coefficients, we obtain a combined strategy maintaining the advantages of both component diffusion strategies simultaneously, and sometimes even better.

6. REFERENCES

[1] P. Braca, S. Marano, and V. Matta, “Enforcing consensus while monitoring the environment in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3375–3380, Jul. 2008.

[2] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[3] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[4] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[5] A. H. Sayed, “Diffusion adaptation over networks,” in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., vol. 3, pp. 322–454. Elsevier, 2014.

[6] A. H. Sayed, “Adaptive networks,” *Proc. of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[7] 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.

[8] S.-Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive netowrks,” *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[9] 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.

[10] 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.

[11] L. Li and J. Chambers, “Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology,” in *Proc. IEEE SSP*, 2009, pp. 757–760.

[12] F. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[13] F. 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.

[14] 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.

[15] 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.

[16] 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, 2017.

[17] X. Zhao and A. H. Sayed, “Clustering via diffusion adaptation over networks,” in *Proc. Int. Workshop Cognitive Inf. Process. (CIP)*, May. 2012, pp. 1–6.

[18] J. Arenas-Garcia, M. Martinez-Ramon, A. Navia-Vazquez, and A. R. Figueiras-Vidal, “Plant identification via adaptive combination of transversal filters,” *Signal Process.*, vol. 86, no. 9, pp. 2430 – 2438, 2006.

[19] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, “Mean-square performance of a convex combination of two adaptive filters,” *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[20] N. J. Bershad, J. C. M. Bermudez, and J. Y. Tourneret, “An affine combination of two LMS adaptive filters—transient mean-square analysis,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1853–1864, May 2008.

[21] J. Arenas-Garcia, L. A. Azpicueta-Ruiz, M. T. M. Silva, V. H. Nascimento, and A. H. Sayed, “Combinations of adaptive filters: Performance and convergence properties,” *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 120–140, Jan. 2016.

[22] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, “A normalized adaptation scheme for the convex combination of two adaptive filters,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2008, pp. 3301–3304.

[23] R. Alain, B. Francis, C. Stephane, and G. Yves, “Simple MKL,” *Journal of Machine Learning Research*, vol. 9, no. 3, pp. 2491–2521, 2008.

[24] J. Chen, C. Richard, and P. Honeine, “Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model,” *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 480–492, Jan. 2013.