ENHANCED DIFFUSION LEARNING OVER NETWORKS

Ricardo Merched[⋆], *Stefan Vlaski*[†] *and Ali H. Sayed*[†]

[⋆]Department of Electronics and Computer Engineering, Universidade Federal do Rio de Janeiro
[†] Institute of Electrical Engineering, École Polytechnique Fédérale de Lausanne

ABSTRACT

This work develops a variation of diffusion learning by incorporating an adaptive construction for the combination weights through local fusion steps. This leads to an implementation with enhanced convergence rate and mean-square-error performance while maintaining the same level of complexity as standard implementations. The approach is based on formulating optimal or close-to-optimal learning and fusion steps using a proximity function rationale within neighborhoods. The first version of the algorithm employs exact fusion in the least-squares sense using inverses of uncertainty matrices. The second version replaces these matrices by diagonal approximations with reduced complexity. The result is an LMS-complexity scheme with improved performance for distributed learning over networks.

Index Terms— diffusion networks, fusion, least-squares, adaptation, combination weights.

1. INTRODUCTION

In typical implementations of consensus and diffusion strategies for learning over networks, it is customary to combine estimates from neighborhoods by relying on convex combination weights [1]–[6]. In general, these weights are fixed and chosen as (scalar) entries of left-stochastic combination matrices. There have been works in the literature where the combination weights have also been learned as part of the adaptation process. For example, the *relative variance* combination rule from [7] was derived by optimizing the instantaneous mean-square-error measure as the learning algorithm evolves over time.

In this work, we take a different route to learning the combination weights, leading to enhanced performance. We attain this objective by formulating optimal or close-to-optimal fusion steps locally under a proximity rationale, and subsequently reduce the complexity of the iterations by replacing uncertainty matrices by scalar approximations. The main difference between our derivation and existing approaches is that uncertainties in the estimates by the agents are carried from self-learning to the social learning phase, and back to self-learning again, in a continuous fashion. This results in adaptive combination weights and lead to lower mean-square-error and faster convergence compared to existing approaches.

The work of A. H. Sayed was supported in part by NSF grant CCF-1524250. Emails: merched@lps.ufrj.br, {stefan.vlaski, ali.sayed}@epfl.ch.

2. DIFFUSION ALGORITHMS REVISITED

We consider a strongly-connected network of distributed agents, represented by a collection of N nodes in Fig. 1. Each agent k receives streaming data $\{d_k(i), \boldsymbol{u}_{k,i}\}$, assumed to be related via a linear regression model of the form

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

where i is the time index, $\boldsymbol{w}_k^o$ is an unknown local parameter of size $M \times 1$, $\boldsymbol{u}_{k,i}$ is a regression (row) vector of size $1 \times M$, and $v_k(i)$ is additive zero-mean white noise, which is temporally and spatially uncorrelated with other data.

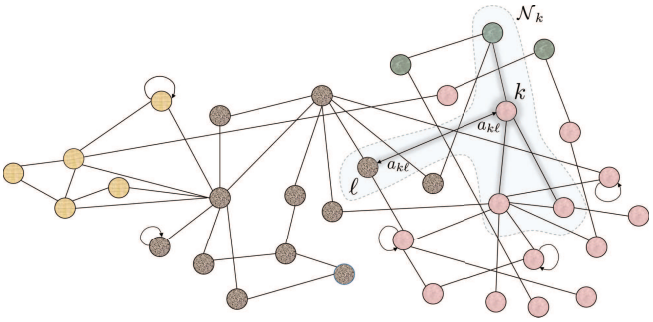

Fig. 1. Illustration of a network with $N = 34$ agents.

We collect the data measured at each agent k up to time i into the quantities:

$$\boldsymbol{H}_{k,i} = \mathsf{col}\{\boldsymbol{u}_{k,1}, \boldsymbol{u}_{k,2}, \ldots, \boldsymbol{u}_{k,i}\} \tag{2}$$

$$\boldsymbol{y}_{k,i} = \mathsf{col}\{d_k(1), d_k(2), \ldots, d_k(i)\} \tag{3}$$

In a non-cooperative setting, each agent k would estimate its parameter vector $\boldsymbol{w}_k^o$ by solving a weighted least-squares problem of the form:

$$\min_{\boldsymbol{w}_k} \|\boldsymbol{y}_{k,i} - \boldsymbol{H}_{k,i}\boldsymbol{w}_k\|^2_{\boldsymbol{\Lambda}_{k,i}} \tag{4}$$

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

for some weighting matrix $\boldsymbol{\Lambda}_{k,i} = \text{diag}\{\lambda^{i-1}, \ldots, \lambda, 1\}$ where $0 \ll \lambda \leqslant 1$ is an exponential weighting factor. In many important situations, however, the individual models $\{\boldsymbol{w}_k^o\}$ are related, such as varying smoothly over a graph. In these cases, it is reasonable to encourage agents to seek estimates that are close to each other. One way to achieve this objective is to replace problem (4) by a regularized problem that enforces coupling among agents. To exploit this idea, let $\text{bdiag}(\cdot)$ denote a block diagonal operator and introduce the following extended quantities :

$$\begin{aligned}
\boldsymbol{\mathcal{W}}^o &= \text{col}\{\boldsymbol{w}_1^o, \boldsymbol{w}_2^o, \ldots, \boldsymbol{w}_N^o\} && (5)\\
\boldsymbol{y}_i &= \text{col}\{\boldsymbol{y}_{1,i}, \boldsymbol{y}_{2,i}, \ldots, \boldsymbol{y}_{N,i}\} && (6)\\
\boldsymbol{v}_i &= \text{col}\{\boldsymbol{v}_{1,i}, \boldsymbol{v}_{2,i}, \ldots, \boldsymbol{v}_{N,i}\} && (7)\\
\boldsymbol{\mathcal{H}}_i &= \text{bdiag}\{\boldsymbol{H}_{1,i}, \boldsymbol{H}_{2,i}, \ldots, \boldsymbol{H}_{N,i}\} && (8)\\
\boldsymbol{\Lambda}_i &= (1/N)\text{bdiag}\{\boldsymbol{\Lambda}_{1,i}, \ldots, \boldsymbol{\Lambda}_{N,i}\} && (9)
\end{aligned}$$

Definitions (5)-(9) allow us to write a global linear model for the data collected across the network up to time i as follows:

$$\boldsymbol{y}_i = \boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{W}}^o + \boldsymbol{v}_i \tag{10}$$

Introduce the extended parameter vector:

$$\boldsymbol{\mathcal{W}} = \text{col}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\} \tag{11}$$

Now, one way to enforce coupling is via a proximity function. This is usually achieved by considering the following regularized cost

$$\min_{\boldsymbol{\mathcal{W}}} \|\boldsymbol{y}_i - \boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{W}}\|^2_{\boldsymbol{\Lambda}_i} + \rho_i(\boldsymbol{\mathcal{W}}) \tag{12}$$

for some regularizer $\rho_i(\boldsymbol{\mathcal{W}})$ whose purpose is to encourage proximity to $\boldsymbol{\mathcal{W}}^o$. However, since $\boldsymbol{\mathcal{W}}^o$ is unknown, we shall instead encourage proximity with respect to the best guess available at that moment. We will explain how to obtain this estimate in the sequel. For now, let us denote its entries by $\overline{\boldsymbol{w}}_{k,i}$ at agent k; i.e., this is the estimate for $\boldsymbol{w}_k^o$ at agent k at time i. We shall also explain how to associate with this estimate a local uncertainty matrix denoted by $\overline{\boldsymbol{P}}_{k,i}$: the better the quality of $\overline{\boldsymbol{w}}_{k,i}$, the smaller $\overline{\boldsymbol{P}}_{k,i}$ will be so that the inverse matrix $\overline{\boldsymbol{P}}_{k,i}^{-1}$ serves as a measure of the uncertainty in the estimate $\overline{\boldsymbol{w}}_{k,i}$.

Using these intermediate estimates (to be constructed later), we replace the cost in (12) by one of the form

$$J_i(\boldsymbol{\mathcal{W}}) = \|\boldsymbol{y}_i - \boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{W}}\|^2_{\boldsymbol{\Lambda}_i} + \sum_{m=1}^{i} \lambda^{i-m} \|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_m\|^2_{\overline{\boldsymbol{\mathcal{P}}}_m^{-1}} \tag{13}$$

where $\overline{\boldsymbol{\mathcal{W}}}_m = \text{col}\{\overline{\boldsymbol{w}}_{1,m}, \overline{\boldsymbol{w}}_{2,m}, \ldots, \overline{\boldsymbol{w}}_{N,m}\}$, and

$$\overline{\boldsymbol{\mathcal{P}}}_m^{-1} = \text{bdiag}\left\{\overline{\boldsymbol{P}}_{1,m}^{-1}, \overline{\boldsymbol{P}}_{2,m}^{-1}, \cdots, \overline{\boldsymbol{P}}_{N,m}^{-1}\right\} \tag{14}$$

One possible choice for $\overline{\boldsymbol{w}}_{k,i}$ is $\boldsymbol{w}_{k,i-1}$, the estimate obtained at time $i-1$. Observe that (13) pushes $\boldsymbol{\mathcal{W}}$ towards the estimates $\overline{\boldsymbol{\mathcal{W}}}_m$ by computing an exponentially weighted error measure from time $m = 1$ up to time $m = i$. Since this objective function is quadratic, it can be expressed in terms of its minimizer and Hessian matrix as

$$J_i(\boldsymbol{\mathcal{W}}) = \|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}_i\|^2_{\boldsymbol{\mathcal{P}}_i^{-1}} + c \tag{15}$$

for some constant c, which can be ignored, and where

$$\boldsymbol{\mathcal{W}}_i = \arg\min_{\boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}_i\|^2_{\boldsymbol{\mathcal{P}}_i^{-1}} \tag{16}$$

$$\boldsymbol{\mathcal{P}}_i^{-1} = \nabla^2 J(\boldsymbol{\mathcal{W}}) = \boldsymbol{\mathcal{H}}_i^* \boldsymbol{\Lambda}_i \boldsymbol{\mathcal{H}}_i + \sum_{m=1}^{i} \lambda^{i-m} \overline{\boldsymbol{\mathcal{P}}}_m^{-1} \tag{17}$$

with * denoting complex conjugate transposition. Of course, this reformulation is not useful in finding the minimizer of (13) since it involves $\boldsymbol{\mathcal{W}}_i$ itself. It does however allow us to obtain a recursive algorithm for the solution. To see this, let $\{\boldsymbol{d}_i, \boldsymbol{\mathcal{U}}_i\}$ contain the most recent data at time i:

$$\begin{aligned}
\boldsymbol{d}_i &= \text{col}\{d_1(i), d_2(i), \ldots, d_N(i)\} && (18)\\
\boldsymbol{\mathcal{U}}_i &= \text{bdiag}\{\boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,i}, \ldots, \boldsymbol{u}_{N,i}\} && (19)
\end{aligned}$$

Then, (13) can be expressed as

$$\begin{aligned}
J_i(\boldsymbol{\mathcal{W}}) &= \|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \boldsymbol{\mathcal{W}}\|^2 + \|\boldsymbol{y}_{i-1} - \boldsymbol{\mathcal{H}}_{i-1} \boldsymbol{\mathcal{W}}\|^2_{\lambda \boldsymbol{\Lambda}_{i-1}}\\
&\quad + \lambda \sum_{m=1}^{i-1} \lambda^{i-m-1} \|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_m\|^2_{\overline{\boldsymbol{\mathcal{P}}}_m^{-1}} + \|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_i\|^2_{\overline{\boldsymbol{\mathcal{P}}}_i^{-1}}\\
&= \|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \boldsymbol{\mathcal{W}}\|^2 + \lambda \underbrace{\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}_{i-1}\|^2_{\boldsymbol{\mathcal{P}}_{i-1}^{-1}}}_{J_{i-1}(\boldsymbol{\mathcal{W}})} + \|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_i\|^2_{\overline{\boldsymbol{\mathcal{P}}}_i^{-1}}
\end{aligned} \tag{20}$$

Note that $J_i(\boldsymbol{\mathcal{W}})$ is composed of three components. The first term, $\|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \boldsymbol{\mathcal{W}}\|^2$, fits $\boldsymbol{\mathcal{W}}$ to the most recent data; the second term, $\lambda \|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}_{i-1}\|^2_{\boldsymbol{\mathcal{P}}_{i-1}^{-1}}$, incorporates past information; and the last term $\|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_i\|^2_{\overline{\boldsymbol{\mathcal{P}}}_i^{-1}}$ promotes smoothness and closeness to the intermediate estimate $\overline{\boldsymbol{\mathcal{W}}}_i$ yet to be specified.

Using completion of squares in (20), we can combine the first two terms and write the minimization problem as:

$$\min_{\boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}} - \widehat{\boldsymbol{\mathcal{W}}}_i\|^2_{\widehat{\boldsymbol{\mathcal{P}}}_i^{-1}} + \|\boldsymbol{\mathcal{W}} - \overline{\boldsymbol{\mathcal{W}}}_i\|^2_{\overline{\boldsymbol{\mathcal{P}}}_i^{-1}} \tag{21}$$

in terms of the updated local estimates:

$$\begin{aligned}
\widehat{\boldsymbol{\mathcal{W}}}_i &= \boldsymbol{\mathcal{W}}_{i-1} + \widehat{\boldsymbol{\mathcal{P}}}_i \boldsymbol{\mathcal{U}}_i^* (\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \boldsymbol{\mathcal{W}}_{i-1}) && (22)\\
\widehat{\boldsymbol{\mathcal{P}}}_i^{-1} &= \lambda \boldsymbol{\mathcal{P}}_{i-1}^{-1} + \boldsymbol{\mathcal{U}}_i^* \boldsymbol{\mathcal{U}}_i && (23)
\end{aligned}$$

where $\widehat{\boldsymbol{\mathcal{W}}}_i = \text{col}\{\widehat{\boldsymbol{w}}_{1,i}, \widehat{\boldsymbol{w}}_{2,i}, \ldots, \widehat{\boldsymbol{w}}_{N,i}\}$, and

$$\widehat{\boldsymbol{\mathcal{P}}}_{i-1}^{-1} = \text{bdiag}\left\{\widehat{\boldsymbol{P}}_{1,i-1}, \widehat{\boldsymbol{P}}_{2,i-1}, \ldots, \widehat{\boldsymbol{P}}_{N,i-1}\right\} \tag{24}$$

We are now left to define $\{\overline{\boldsymbol{w}}_{k,i}, \overline{\boldsymbol{P}}_{k,i}^{-1}\}$. While $\widehat{\boldsymbol{w}}_{k,i}$, is an improved estimate for $\boldsymbol{w}_k^o$ over $\boldsymbol{w}_{k,i-1}$, since it includes the most recent data, we proceed one step further by allowing for an exchange of information within neighborhoods. This can

be obtained by designing $\overline{\boldsymbol{w}}_i$ as a fusion in the neighborhood $\mathcal{N}_k$ of the agent's estimates $\{\widehat{\boldsymbol{w}}_{k,i}, \widehat{\boldsymbol{P}}_{k,i}\}$, defined in (22) and (23), in a weighted least-squares manner, however, without including agent k. This is because the contribution of agent k will be added by the first term of (21). Now, since (21) is equivalent to minimizing (15), instead of designing $\overline{\boldsymbol{w}}_i$, we can design the result of their fusion, say, $\{\boldsymbol{\mathcal{W}}_i, \boldsymbol{\mathcal{P}}_i\}$, directly as the solution to

$$\min_{\boldsymbol{w}'_k} \sum_{\ell\in\mathcal{N}_k}^{N} a_{k\ell}\|\boldsymbol{w}'_k - \widehat{\boldsymbol{w}}_{\ell,i}\|^2_{\widehat{\boldsymbol{P}}^{-1}_{\ell,i}} \tag{25}$$

$$= \min_{\boldsymbol{w}'_k} \left\|\boldsymbol{w}'_k - \boldsymbol{w}'_{k,i}\right\|^2_{\boldsymbol{P}'^{-1}_{k,i}} \ , \quad k=1,2,\ldots,N \tag{26}$$

$$= \min_{\boldsymbol{\mathcal{W}}'} \left\|\boldsymbol{\mathcal{W}}' - \boldsymbol{\mathcal{W}}'_i\right\|^2_{\boldsymbol{\mathcal{P}}'^{-1}_i} \tag{27}$$

for positive scalars $\{a_{k\ell}\}$. The solution to (25) is given by

$$\boldsymbol{w}'_{k,i} \ = \ \sum_{\ell\in\mathcal{N}_k} \mathbf{A}_{k\ell,i}\widehat{\boldsymbol{w}}_{\ell,i} \ , \quad \mathbf{A}_{k\ell,i} \stackrel{\Delta}{=} a_{k\ell}\boldsymbol{P}'_{k,i}\widehat{\boldsymbol{P}}^{-1}_{\ell,i} \tag{28}$$

if $\ell\in\mathcal{N}_k$, while $\mathbf{A}_{k\ell,i}=\mathbf{0}$ if $\ell\notin\mathcal{N}_k$. The quantity $\boldsymbol{P}'^{-1}_{k,i}$ is the uncertainty that results from theses estimates,

$$\boldsymbol{P}'^{-1}_{k,i} \ = \ \sum_{\ell\in\mathcal{N}_k} a_{k\ell}\widehat{\boldsymbol{P}}^{-1}_{\ell,i} \tag{29}$$

Hence, in extended vector form, this yields $\{\boldsymbol{\mathcal{W}}'_i, \boldsymbol{\mathcal{P}}'_i\}$. Now, observe that if we select $\boldsymbol{\mathcal{W}}_i = \boldsymbol{\mathcal{W}}'_i$ and $\boldsymbol{\mathcal{P}}_i = \boldsymbol{\mathcal{P}}'_i$, the costs (27) and (15) will have the same form. In other words, by selecting (27) and (15) to have the same minimizer with the same uncertainty, we are able to propagate these quantities from self-learning to social learning in a true recursion.

Note that while we could assume for simplicity that all agents in $\mathcal{N}_k$ are equally important in the fusion process, say, $a_{k\ell}=\gamma_k\geqslant 0$, node k itself can be assigned a different weight a_{kk}, relative to its neighbors, implying that it can have more or less certainty of its own estimate. In order for these coefficients to add up to one, we select

$$a_{kk} = 1-\gamma_k(n_k-1) \tag{30}$$

where n_k is the degree of the neighborhood. Moreover, by associating the scalars $a_{k\ell}$ to entries of a $N\times N$ matrix $\boldsymbol{A}$, then $\boldsymbol{A}$ is referred to a Laplacian matrix.

Finally, defining $\boldsymbol{\mathcal{A}} = \boldsymbol{A}\otimes\boldsymbol{I}_M$, where $\otimes$ denotes the Kronecker product, then, in extended matrix notation, it holds that

$$\boldsymbol{\mathcal{W}}_i = \mathbf{A}_i\widehat{\boldsymbol{\mathcal{W}}}_i \tag{31}$$

where, using (24),

$$\mathbf{A}_i \ = \ \boldsymbol{\mathcal{P}}_i\boldsymbol{\mathcal{A}}\widehat{\boldsymbol{\mathcal{P}}}^{-1}_i \ , \ \ \boldsymbol{\mathcal{P}}^{-1}_i = \mathsf{bdiag}\left[\boldsymbol{\mathcal{A}}\widehat{\boldsymbol{\mathcal{P}}}^{-1}_i(\mathbb{1}\otimes\boldsymbol{I}_M)\right] \tag{32}$$

It is easily verified that $\mathbf{A}_i\mathbb{1} = (\mathbb{1}\otimes\boldsymbol{I}_M)$ is block right-stochastic by construction. The diffusion recursions derived so far are listed in Table 1, and constitute what we shall refer to as the *Adapt-and-Fuse* (AAF) diffusion, which extends the usual description of diffusion strategy known as *Adapt-then-Combine* (ATC), where $\mathbf{A}_i=\boldsymbol{\mathcal{A}}$. Figure 2 illustrates the equivalent global transmission scheme, where we have defined $\boldsymbol{\mathcal{G}}_i = \widehat{\boldsymbol{\mathcal{P}}}_i\boldsymbol{\mathcal{U}}^*_i$.

Initialization: $\boldsymbol{w}_0=\mathbf{0}$, $\boldsymbol{\mathcal{P}}^{-1}_0=\epsilon\boldsymbol{I}$ for small ϵ
$\widehat{\boldsymbol{\mathcal{P}}}^{-1}_i = \lambda\boldsymbol{\mathcal{P}}^{-1}_{i-1} + \boldsymbol{\mathcal{U}}^*_i\boldsymbol{\mathcal{U}}_i$
$\boldsymbol{\mathcal{P}}^{-1}_i = \mathsf{bdiag}\left[\boldsymbol{\mathcal{A}}\widehat{\boldsymbol{\mathcal{P}}}^{-1}_i(\mathbb{1}\otimes\boldsymbol{I}_M)\right]$
$\mathbf{A}_i = \boldsymbol{\mathcal{P}}_i\boldsymbol{\mathcal{A}}\widehat{\boldsymbol{\mathcal{P}}}^{-1}_i$
$\widehat{\boldsymbol{\mathcal{W}}}_i = \boldsymbol{\mathcal{W}}_{i-1} + \widehat{\boldsymbol{\mathcal{P}}}_i\boldsymbol{\mathcal{U}}^*_i(\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i\boldsymbol{\mathcal{W}}_{i-1})$
$\boldsymbol{\mathcal{W}}_i = \mathbf{A}_i\widehat{\boldsymbol{\mathcal{W}}}_i$

Table 1. AAF Diffusion Adaptation.

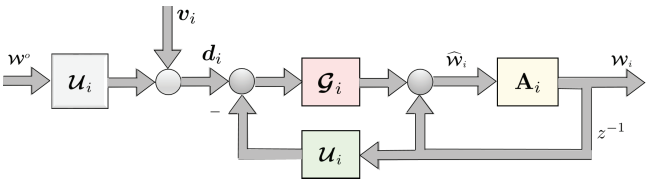

Fig. 2. Global Description of the AAF scheme.

3. SIMPLIFIED AAF RECURSIONS

Because the optimal coefficients of $\mathbf{A}_i$ and all its defining covariances w.r.t. node k are complex and matrix-valued, they require complex matrix $\times$ matrix operations and inversions at each time i. These computations can be cumbersome considering that agents should perform elementary operations when fusing their estimates within $\mathcal{N}_k$. By assuming uncorrelated input regressors, we can simplify the AAF recursions by restricting the covariances to diagonal matrices, i.e., we set $\boldsymbol{P}_{k,i}\approx\sigma^2_k(i)\boldsymbol{I}$, and $\widehat{\boldsymbol{P}}_{k,i}\approx\widehat{\sigma}^2_\ell(i)\boldsymbol{I}$. With these approximations, the $N\times N$ block entry of $\mathbf{A}_i$ in (32) simplifies to

$$[\mathbf{A}_i]_{k\ell} \ = \ a_{k\ell}(i)\boldsymbol{I} = a_{k\ell}\sigma^2_k(i)\widehat{\sigma}^{-2}_\ell(i)\boldsymbol{I}$$

The resulting algorithm is shown in Table 2, which we refer to as the Simplified-AAF (SAAF) algorithm.

Initialization: $\boldsymbol{w}_{\ell,0}=\mathbf{0}$, $\sigma^{-2}_k(0)=\epsilon$ for small ϵ
for $k=1$ to N:
$\widehat{\sigma}^{-2}_k(i) = \lambda\sigma^{-2}_k(i-1) + \lvert u_\ell(i)\rvert^2$
$\sigma^{-2}_k(i) = \sum_{\ell\in\mathcal{N}_k} a_{k\ell}\widehat{\sigma}^{-2}_{\ell k}(i)$
$a_{\ell k}(i) = a_{k\ell}\sigma^2_k(i)\widehat{\sigma}^{-2}_\ell(i)$
$\widehat{\boldsymbol{w}}_{k,i} = \boldsymbol{w}_{k,i-1} + \hat{\sigma}^2_k(i)\boldsymbol{u}^*_{k,i}[d_k(i)-\boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}]$
$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}(i)\widehat{\boldsymbol{w}}_{\ell,i}$

Table 2. SAAF Algorithm.

Note that the computational complexity for updating the combination matrix in the SAAF recursions is of $\mathcal{O}(NM)$.

This is in contrast to the relative-variance rule of [7] which requires $\mathcal{O}(N\mathcal{N}M)$ computations per iteration in order to accomplish a similar task, where $\mathcal{N} = \sum_{k=1}^{N} \mathcal{N}_k$. That is, the uncertainties of each node in [7] are assumed to vary over the neighborhoods, while in the proposed construction, the nodes uncertainties are absolute; they are updated independently, and combined through the fixed Laplacian matrix $\boldsymbol{A}$.

4. SIMULATIONS

In order to illustrate the performance of the proposed AAF based algorithms in comparison with existing cooperative-based schemes, we consider a topology with $N = 20$ agents with unknown vectors of size $M = 10$, and compare: (i) the power normalized LMS-based algorithms employing a fixed combination policy; (ii) a normalized LMS version of the adaptive relative variance diffusion algorithm of [7], and (iii) the RLS-based diffusion of, e.g., [11].

In order to set one possible theoretical benchmark for comparison on the minimum mean-square-deviation (MSD), we consider the one corresponding to the well established diffusion LMS algorithm. The network MSD in this case, for sufficiently small μ is given by (see [8], pp. 606)

$$\mathsf{MSD}_{\text{dist,av}} = \frac{M}{2}\left(\sum_{k=1}^{N} \mu_k^2 p_k^2 \sigma_{u,k}^2 (\sigma_{v,k}^2 + \sigma_{u,k}^2 \|\boldsymbol{w}^o - \boldsymbol{w}_k^o\|^2)\right) \cdot \left(\sum_{k=1}^{N} \mu_k p_k \sigma_{u,k}^2\right)^{-1} \tag{33}$$

in terms of the input and noise variances, $\sigma_{u,k}^2$ and $\sigma_{v,k}^2$ respectively, and the entries p_k of the Perron vector associated to the combination matrix, here chosen as the optimal relative-variance rule (for the cases when $\boldsymbol{w}_k^o = \boldsymbol{w}^o$). This is illustrated by a thick straight line.

♦ ***Scenario 1 (Performance of the SAAF algorithm)***: Figure 3 shows typical ensemble average learning curves for uncorrelated inputs. We set the step-size as $\mu = 0.002$ for the LMS-based algorithms, $\lambda = 1$ for the RLS algorithms, and $\gamma_k = 0.0024$ in (30) for the proposed recursions.

We see that the proposed simplified algorithm outperforms LMS-based algorithms, and with reduced complexity compared to the relative-variance policy. We clearly see a difference in terms of the MSD attained. Moreover, it exhibits approximately the same performance of existing RLS-based recursions. It is worth noting that for the latter, the use of a Metropolis combination rule $\boldsymbol{C}$ does not yield improvement against the case when $\boldsymbol{C} = \boldsymbol{I}$.

♦ ***Scenario 2 (Performance of the full AAF algorithm for colored input and $\lambda < 1$ for all RLS algorithms)***: Figure 4 shows the curves for the RLS algorithms when $\lambda = 0.99$, and for a slightly colored AR process, with pole at 0.5.

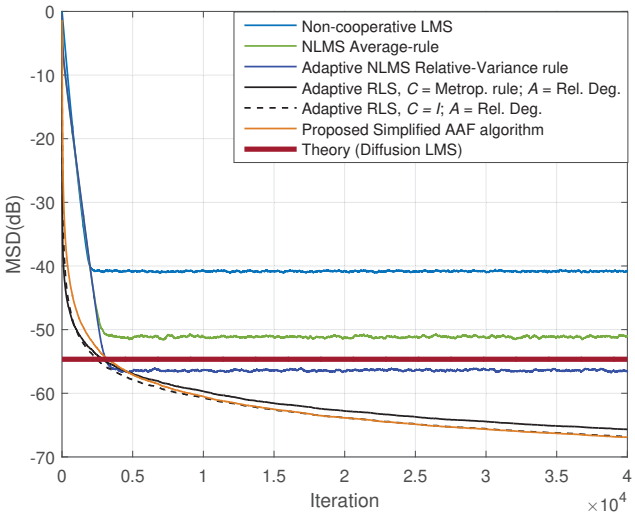

Fig. 3. Comparison among algorithms, uncorrelated input.

We verify that existing diffusion RLS algorithms have their performance degraded. The exact AAF-RLS, despite the computational complexity, outperforms in speed and MSE level, which continues to decrease beyond $8 \cdot 10^4$ iterations.

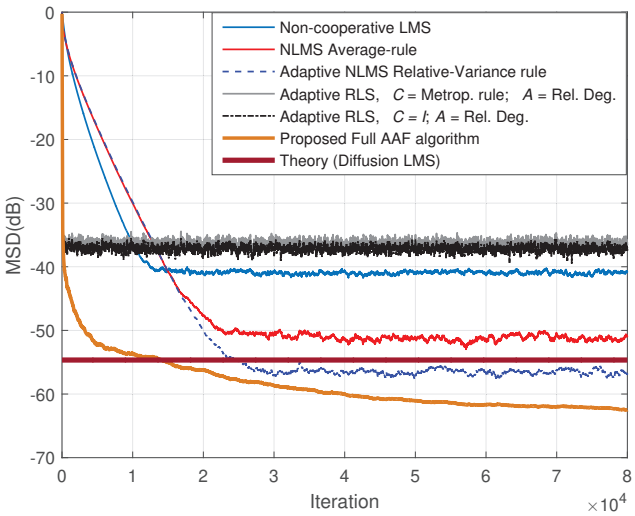

Fig. 4. AAF performance with colored input and for $\lambda < 1$.

5. CONCLUSIONS

We have proposed a construction for optimized diffusion networks which fuses estimates and uncertainties at every node in the LS sense. The proposed AAF recursions outperform existing diffusion algorithms, especially when $\lambda < 1$, and for colored inputs. For uncorrelated data, we verified that its simplified version outperforms all other algorithms, and exhibiting the same convergence performance of existing RLS diffusion schemes, however, under LMS complexity.

6. REFERENCES

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

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

[3] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. 29, no. 1, pp. 42-50, Jan. 1984.

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48- 61, Jan. 2009.

[5] 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.

[6] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674-690, Aug. 2011.

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

[8] A. H. Sayed, Adaptation, Learning, and Optimization over Networks, Foundations and Trends in Machine Learning, vol. 7, issue 4-5, NOW Publishers, Boston-Delft, 518pp, 2014. ISBN 978-1-60198-850-8, DOI 10.1561/2200000051.

[9] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1535-1542, Monticello, IL, October 2012.

[10] R. G. P. Pinto and R. Merched "A compressed sensing approach to block-iterative equalizers", *IEEE Trans. on Signal Processing*, vol. 66, no. 4, pp. 1007-1022, Dec. 2017.

[11] A. H. Sayed and C. Lopes, "Distributed recursive least-squares strategies over adaptive networks," *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 233-237, October-November, 2006.

[12] H. Salami, B. Ying, and A. H. Sayed, "Diffusion social learning over weakly-connected graphs," *Proc. IEEE ICASSP*, pp. 4119-4123, Shanghai, China, March 2016.

[13] S. Vlaski, L. Vandenberghe, and A. H. Sayed, "Diffusion stochastic optimization with non-smooth regularizers," *Proc. IEEE ICASSP*, pp. 4149–4153, Shanghai, China, March 2016.

[14] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning – Part I: Algorithm development," *IEEE Trans. Signal Processing*, vol. 67, no. 3, pp. 708-723, Feb. 2019.

[15] R. Nassif, S. Vlaski, and A. H. Sayed, "Learning over multitask graphs —- Part I: Stability analysis," available as arXiv:1805.08535v1, May 2018.