SPATIO-TEMPORAL DIFFUSION MECHANISMS FOR ADAPTATION OVER NETWORKS

Jae-Woo Lee, Seong-Eun Kim†, Woo-Jin Song*†, and Ali H. Sayed‡*

*Department of Electronic and Electrical Engineering, POSTECH
†Educational Institute of Future Information Technology, POSTECH
Pohang, Gyungbuk, 790-784, Korea
‡Department of Electrical Engineering, University of California, Los Angeles, CA 90095
Email:{magic0ad,raslove,wjsong}@postech.ac.kr, ‡sayed@ee.ucla.edu

ABSTRACT

This work develops diffusion algorithms for adaptation over networks that endow nodes with both cooperation abilities and temporal processing abilities. Each node is allowed to share information locally with its neighbors. At the same time, each node filters past data and uses them to enhance the collaborative process. In this manner, the resulting algorithms consist of three stages: adaptation, spatial processing, and temporal processing. The order of these operations can be inter-changed leading to a total of six variations. The results indicate that the version that performs adaptation prior to the steps of spatial cooperation and temporal processing leads to best performance.

1. INTRODUCTION

Consider a set of N nodes that are distributed over a region in space, as shown in Fig. 1. The set of nodes that are connected to node k (including k itself) is called the neighborhood of node k and is denoted by $\mathcal{N}_k$. Each node k receives scalar measurements $\boldsymbol{d}_k(i)$ and $1\times M$ regression vectors $\boldsymbol{u}_{k,i}$ at successive time instants i. The data are assumed to satisfy a linear regression model of the form

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

where w^o is an $M \times 1$ parameter vector of interest. The nodes would like to estimate w^o in a distributed and adaptive manner through local cooperation. Each node is allowed to share information only with its neighbors. Distributed adaptive problems of this type have been proposed and studied in some detail in [1-8]. Several algorithms have been developed for this purpose, such as incremental LMS [1,2], diffusion LMS [3-5], diffusion RLS [6], and diffusion Kalman filters and smoothers [7]. Algorithms based on average consensus also have been proposed in [8]. Adaptive diffusion algorithms exploit the spatial diversity in the data to great effect. Performance and stability analyses in the aforementioned references [1]-[7] derive expressions that characterize the mean-square behavior of adaptive diffusion algorithms and establish their superior performance over non-cooperative schemes.

The purpose of this article is to add a temporal dimension to the processing at the nodes. Rather than rely solely on current data and on the data shared with the neighbors at a particular time instant, we endow each node with a local memory to filter its past data and use them in addition to the current data. In this way, the effects of previous data and estimates are used to enhance performance further. We proceed to derive the spatio-temporal diffusion algorithms and to illustrate their behavior.

This work was supported in part by the Brain Korea 21 Project in 2011 while J-W. Lee was a PhD student visitor at the UCLA Adaptive Systems Laboratory. The work of A. H. Sayed was supported in part by NSF grants CCF-1011918 and CCF-0942936.

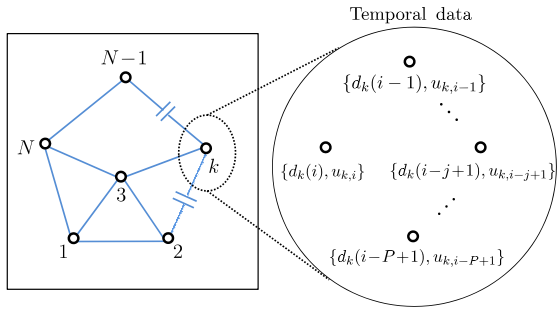

Figure 1: A spatio-temporal network, where each node has access to the P data points $\{d_k(i-j), u_{k,i-j}\}$ for $j = 0, 1, \ldots, P-1$.

2. PROBLEM FORMULATION

Each node k is assumed to have access to data from time i and to data from the previous $P-1$ time instants. Thus, consider the following global cost function:

$$J^{\text{glob}}(w) = \sum_{k=1}^{N}\left(\sum_{j=0}^{P-1} q_{k,j}\,\mathrm{E}\,|\boldsymbol{d}_k(i-j) - \boldsymbol{u}_{k,i-j}w|^2\right) \tag{2}$$

where E denotes the expectation operator, and $q_{k,j}$ is a non-negative scalar that represents the weight that node k gives to data from time $i-j$. We collect the weights $\{q_{k,j}\}$ into an $N\times P$ matrix Q and require that Q satisfy the normalization condition $Q\mathbb{1}_P = \mathbb{1}_N$,where $\mathbb{1}_k$ denotes the $k\times 1$ vector with unit entries.

When the processes $\boldsymbol{d}_k(i)$ and $\boldsymbol{u}_{k,i}$ are jointly wide-sense stationary (WSS) and the regression process $\boldsymbol{u}_{k,i}$ is also WSS, the optimal solution w^o of (2) is the solution to the following normal equations (where we assume inverses exist whenever necessary)[9]:

$$w^o = \left(\sum_{k=1}^{N} R_{u,k}\right)^{-1}\left(\sum_{k=1}^{N} R_{du,k}\right) \tag{3}$$

where $R_{u,k} = \mathrm{E}\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}$ and $R_{du,k} = \mathrm{E}\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^*$.

2.1 Local Optimization over Space

Following the approach developed in [4] to derive spatial diffusion, we introduce an $N \times N$ weighting matrix C with

individual non-negative real entries $\{c_{l,k}\}$ such that

$$c_{l,k} = 0 \quad \text{if} \quad l \notin \mathcal{N}_k, \quad C\mathbb{1}_N = \mathbb{1}_N, \quad \mathbb{1}_N^* C = \mathbb{1}_N^*. \tag{4}$$

We further consider the following local cost function at node k; it only employs data that are available to node k:

$$J_k^{\text{loc}}(w) = \sum_{l\in\mathcal{N}_k} c_{l,k} \left(\sum_{j=0}^{P-1} q_{l,j} \mathrm{E}\,|\boldsymbol{d}_l(i-j) - \boldsymbol{u}_{l,i-j} w|^2 \right). \tag{5}$$

The local optimal estimate for w^o is given by:

$$w_k^{\text{loc}} = \Gamma_k^{-1} \left(\sum_{l\in\mathcal{N}_k} c_{l,k} R_{du,l} \right) \text{ where } \Gamma_k = \sum_{l\in\mathcal{N}_k} c_{l,k} R_{u,l}. \tag{6}$$

We can rewrite (5) in terms of w_k^{loc} as follows:

$$J_k^{\text{loc}}(w) = \left\| w - w_k^{\text{loc}} \right\|_{\Gamma_k}^2 + \text{mmse}. \tag{7}$$

where mmse denotes the resulting minimum mean-square error and the notation $\|a\|_\Sigma^2 = a^*\Sigma a$ represents a weighted vector squared norm. In this manner, we can re-express the original global cost function (2) in the form:

$$J^{\text{glob}}(w) = \sum_{l=1}^{N} J_l^{\text{loc}}(w) = J_k^{\text{loc}}(w) + \sum_{l\neq k}^{N} J_l^{\text{loc}}(w). \tag{8}$$

From (5), (7) and (8), we conclude that minimizing (2) is equivalent to minimizing the alternative cost function:

$$J^{\text{glob}\prime}(w) = J_k^{\text{loc}}(w) + \sum_{l\neq k}^{N} \left\| w - w_l^{\text{loc}} \right\|_{\Gamma_l}^2. \tag{9}$$

2.2 Local Optimization over Time

As explained in [4], the equivalent expression (9) allows us to relate the global optimization problem (2) to local optimization problems of the form (5) at the local nodes. In this manner, expression (9) can be used to motivate a distributed way for solving the global problem (2) through local interactions at the nodes. But before we discuss the distributed solution, we repeat a similar argument in the time domain. Starting from (5), we can write it as:

$$J_k^{\text{loc}}(w) = \sum_{j=0}^{P-1} \sum_{l\in\mathcal{N}_k} c_{l,k} \left(q_{l,j} \mathrm{E}\,|\boldsymbol{d}_l(i-j) - \boldsymbol{u}_{l,i-j} w|^2 \right). \tag{10}$$

Expression (10) motivates us to introduce the following cost function at node k for every time instant j ($j = 0, 1, 2, \ldots, P-1$):

$$J_{k,j}^{\text{loc}}(w) = \sum_{l\in\mathcal{N}_k} c_{l,k} \left(q_{l,j} \mathrm{E}|\boldsymbol{d}_l(i-j) - \boldsymbol{u}_{l,i-j} w|^2 \right). \tag{11}$$

Then,

$$J_k^{\text{loc}}(w) = \sum_{j=0}^{P-1} J_{k,j}^{\text{loc}}(w). \tag{12}$$

By the WSS assumption on the regression process, the optimal temporal solution of (11) at node k is

$$w_{k,j}^{\text{loc}} = \Lambda_{k,j}^{-1} \left(\sum_{l\in\mathcal{N}_k} c_{l,k} q_{l,j} R_{du,l} \right) \tag{13}$$

where $\Lambda_{k,j} = \sum_{l\in\mathcal{N}_k} c_{l,k} q_{l,j} R_{u,l}$. In a manner similar to (7), we can express (11) as

$$J_{k,j}^{\text{loc}}(w) = \left\| w - w_{k,j}^{\text{loc}} \right\|_{\Lambda_{k,j}}^2 + \text{mmse}. \tag{14}$$

Using (12) and (14), we observe that minimizing the local cost function (5) is equivalent to minimizing the following cost:

$$J_k^{\text{loc}\prime}(w) = J_{k,0}^{\text{loc}}(w) + \sum_{j=1}^{P-1} \left\| w - w_{k,j}^{\text{loc}} \right\|_{\Lambda_{k,j}}^2. \tag{15}$$

Combining (9) and (15), we arrive at our initial conclusion, namely, that optimizing the cost below over w is equivalent to optimizing the original global cost function (2):

$$J^{\text{glob}\prime\prime}(w) = \sum_{l\in\mathcal{N}_k} c_{l,k} \left(q_{l,0} \mathrm{E}|\boldsymbol{d}_l(i) - \boldsymbol{u}_{l,i} w|^2 \right) + \sum_{j=1}^{P-1} \left\| w - w_{k,j}^{\text{loc}} \right\|_{\Lambda_{k,j}}^2 + \sum_{l\neq k}^{N} \left\| w - w_l^{\text{loc}} \right\|_{\Gamma_l}^2. \tag{16}$$

3. SPATIO-TEMPORAL DIFFUSION LMS

The alternative cost (16) is expressed in terms of the spatial local estimates $\{w_l^{\text{loc}}\}$ and the temporal local estimates $\{w_{k,j}^{\text{loc}}\}$. To minimize this cost in its present form, we need to have access to global information such as the moment matrices, Γ_l and $\Lambda_{k,j}$. Following the arguments in [4], a distributed implementation can be motivated by replacing Γ_l and $\Lambda_{k,j}$ by (non-negative) scaled multiples of the identity matrix and by replacing w_l^{loc} and $w_{k,j}^{loc}$ by local estimates ψ_l and $\phi_{k,j}$, respectively:

$$\begin{aligned} \Gamma_l &\Longrightarrow b_{l,k} I_M & w_l^{\text{loc}} &\Longrightarrow \psi_l \\ \Lambda_{k,j} &\Longrightarrow h_{k,j} I_M & w_{k,j}^{\text{loc}} &\Longrightarrow \phi_{k,j} \end{aligned} \tag{17}$$

where $b_{l,k} = 0$ if $l \notin \mathcal{N}_k$, $\mathbb{1}_N^* B = \mathbb{1}_N^*$ and $H\mathbb{1}_P = \mathbb{1}_N$. The matrix B is the $N \times N$ matrix with individual entries $b_{l,k}$ and the matrix H is the $N \times P$ matrix with individual entries $h_{k,j}$. The vectors ψ_l serve as intermediate estimates resulting from the spatial processing and the vectors $\phi_{k,j}$ serve as intermediate estimates resulting from the temporal processing.

In this way, each node k can proceed to minimize a cost of the following form in a distributed manner:

$$J_k^{\text{dist}}(w) = \sum_{l\in\mathcal{N}_k} c_{l,k} \left(q_{l,0} \mathrm{E}|\boldsymbol{d}_l(i) - \boldsymbol{u}_{l,i} w|^2 \right) + \sum_{j=1}^{P-1} h_{k,j} \|w - \phi_{k,j}\|^2 + \sum_{l\in\mathcal{N}_k\setminus\{k\}} b_{l,k} \|w - \psi_l\|^2. \tag{18}$$

We observe that (18) is a localized apporoximation for the global cost function (16). Taking the gradient of (18) with respect to w we obtain:

$$\left[\nabla_w J_k^{\text{dist}}(w) \right]^* = \sum_{l\in\mathcal{N}_k} c_{l,k} q_l \left(R_{u,l} w - R_{du,l} \right) + \sum_{j=1}^{P-1} h_{k,j} \left(w - \phi_{k,j} \right) + \sum_{l\in\mathcal{N}_k\setminus\{k\}} b_{l,k} \left(w - \psi_l \right) \tag{19}$$

where $q_{l,0}$ is now being denoted by q_l for the simplicity. The gradient vector in (19) is a sum of three terms, namely

$$\sum_{l\in\mathcal{N}_k} c_{l,k} q_l \left(R_{u,l} w - R_{du,l} \right), \; \sum_{j=1}^{P-1} h_{k,j} \left(w - \phi_{k,j} \right), \; \sum_{l\in\mathcal{N}_k\setminus\{k\}} b_{l,k} \left(w - \psi_l \right). \tag{20}$$

By ordering these gradients, we arrive at combinations for evaluating the gradient vector (19). For example, one incremental algorithm (see [3,5]) to update the local estimate can

be accomplished in three steps by incorporating the gradient vectors (20) in the following order:

$$\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k} q_l \left(R_{du,l} - R_{u,l} w_{k,i-1}\right) \quad (21)$$

$$\psi_{k,i} = \phi_{k,i} + \lambda_k \sum_{j=1}^{P-1} h_{k,j} \left(\phi_{k,i-j} - \phi_{k,i}\right) \quad (22)$$

$$w_{k,i} = \psi_{k,i} + \nu_k \sum_{l\in\mathcal{N}_k\setminus\{k\}} b_{l,k} \left(\psi_{l,i} - \psi_{k,i}\right) \quad (23)$$

where we replace $\phi_{k,j}$ and ψ_l by the intermediate estimates $\phi_{k,i-j}$ and $\psi_{l,i}$ which are available at time i. We can rewrite (22) and (23) as

$$\psi_{k,i} = (1-\lambda_k+h_{k,0}\lambda_k)\,\phi_{k,i} + \lambda_k \sum_{j=1}^{P-1} h_{k,j}\phi_{k,i-j} \quad (24)$$

$$w_{k,i} = (1-\nu_k+b_{k,k}\nu_k) + \nu_k \sum_{l\in\mathcal{N}_k\setminus\{k\}} b_{l,k}\psi_{l,i}. \quad (25)$$

We define $a_{k,k} = (1-\nu_k+b_{k,k}\nu_k)$ and $a_{l,k} = \nu_k b_{l,k}$ for $l \neq k$, then $N \times N$ weighting matrix A with individual entries $\{a_{l,k}\}$ satisfies $\mathbb{1}^T A = \mathbb{1}^T$. We also define $g_{k,0} = (1-\lambda_k+h_{k,0}\lambda_k)$ and $g_{k,j} = \lambda_k h_{k,j}$ for $j \neq 0$. We then use the instantaneous approximations $R_{u,k} \approx u_{k,i}^* u_{k,i}$ and $R_{du,k} \approx d_k(i) u_{k,i}^*$ to obtain the following diffusion algorithm:

$$\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k} q_l u_{l,i}^* \left(d_l(i) - u_{l,i} w_{k,i-1}\right) \quad (26a)$$

$$\psi_{k,i} = \sum_{j=0}^{P-1} g_{k,j}\phi_{k,i-j} \quad (26b)$$

$$w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i}. \quad (26c)$$

The above algorithms involves three steps: (a) an adaptation step (A) represented by (26a); (b) a temporal filtering step (T) represented by (26b), and a spatial combination step (S) represented by (26c). We use the letters ATS to designate the order of these steps and, therefore, refer to the algorithm as the ATS diffusion version. In a similar manner, we can obtain six different combinations of diffusion algorithms by changing the order in which the temporal and spatial combinations are performed in relation to the adaptation step. The results are summarized in Table 1.

4. PERFORMANCE ANALYSIS

In this section we analyze the ATS diffusion algorithm (26). The other combinations can be analyzed in a similar manner. We follow the approach of [4]. We define the error quantities $\tilde{\boldsymbol{w}}_{k,i} = w^o - w_{k,i}$, $\tilde{\boldsymbol{\psi}}_{k,i} = w^o - \psi_{k,i}$ and $\tilde{\boldsymbol{\phi}}_{k,i} = w^o - \phi_{k,i}$, and the global vectors:

$$\tilde{\boldsymbol{w}}_i = \begin{bmatrix} \tilde{\boldsymbol{w}}_{1,i} \\ \vdots \\ \tilde{\boldsymbol{w}}_{N,i} \\ \tilde{\boldsymbol{\phi}}_{1,i} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i} \\ \wr \\ \tilde{\boldsymbol{\phi}}_{1,i-P+2} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i-P+2} \end{bmatrix}, \; \tilde{\boldsymbol{\phi}}_i = \begin{bmatrix} \tilde{\boldsymbol{\phi}}_{1,i} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i} \\ \tilde{\boldsymbol{\phi}}_{1,i-1} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i-1} \\ \wr \\ \tilde{\boldsymbol{\phi}}_{1,i-P+1} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i-P+1} \end{bmatrix}, \; \tilde{\boldsymbol{\psi}}_i = \begin{bmatrix} \tilde{\boldsymbol{\psi}}_{1,i} \\ \vdots \\ \tilde{\boldsymbol{\psi}}_{N,i} \\ \tilde{\boldsymbol{\phi}}_{1,i} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i} \\ \wr \\ \tilde{\boldsymbol{\phi}}_{1,i-P+2} \\ \vdots \\ \tilde{\boldsymbol{\phi}}_{N,i-P+2} \end{bmatrix}.$$

Table 1: Six combinations of spatio-temporal diffusion. The letters A, T, and S refer to the steps of adaptation, temporal filtering, and spatial combination. The order of the letters indicate the order by which the steps are performed. For example, in TSA, temporal processing is followed by spatial processing and then by adaptation.

1. TSA diffusion
$\phi_{k,i-1} = \sum_{j=0}^{P-1} g_{k,j} w_{k,i-j-1}$
$\psi_{k,i-1} = \sum_{l\in\mathcal{N}_k} a_{l,k}\phi_{l,i-1}$
$w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i}\psi_{k,i-1}\right)$
2. STA diffusion
$\phi_{k,i-1} = \sum_{l\in\mathcal{N}_k} a_{l,k} w_{l,i-1}$
$\psi_{k,i-1} = \sum_{j=0}^{P-1} g_{k,j}\phi_{k,i-j-1}$
$w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i}\psi_{k,i-1}\right)$
3. TAS diffusion
$\phi_{k,i-1} = \sum_{j=0}^{P-1} g_{k,j} w_{k,i-j-1}$
$\psi_{k,i} = \phi_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i}\phi_{k,i-1}\right)$
$w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i}$
4. SAT diffusion
$\phi_{k,i-1} = \sum_{l\in\mathcal{N}_k} a_{l,k} w_{l,i-1}$
$\psi_{k,i} = \phi_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i}\phi_{k,i-1}\right)$
$w_{k,i} = \sum_{j=0}^{P-1} g_{k,j}\psi_{k,i-j}$
5. ATS diffusion
$\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i} w_{k,i-1}\right)$
$\psi_{k,i} = \sum_{j=0}^{P-1} g_{k,j}\phi_{k,i-j}$
$w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i}$
6. AST diffsuion
$\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} q_l c_{l,k} u_{l,i}^* \left(d_l(i) - u_{l,i} w_{k,i-1}\right)$
$\psi_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\phi_{l,i}$
$w_{k,i} = \sum_{j=0}^{P-1} g_{k,j}\psi_{k,i-j}$

We further introduce:

$$\begin{aligned}
\mathcal{M} &= \text{diag}\left\{\mu_1 I_M, \ldots, \mu_N I_M\right\} \\
\mathcal{A} &= A \otimes I_M, \quad \mathcal{C} = C \otimes I_M \\
\boldsymbol{\mathcal{D}}_i &= \text{diag}\left\{\sum_{l=1}^{N} c_{l,1} q_l \boldsymbol{u}_{l,i}^* \boldsymbol{u}_{l,i}, \ldots, \sum_{l=1}^{N} c_{l,N} q_l \boldsymbol{u}_{l,i}^* \boldsymbol{u}_{l,i}\right\} \\
\boldsymbol{\mathcal{G}}_i &= \mathcal{C}^T \text{col}\left\{q_1 \boldsymbol{u}_{1,i}^* \boldsymbol{v}_1(i), \ldots, q_N \boldsymbol{u}_{N,i}^* \boldsymbol{v}_N(i)\right\} \\
\mathcal{D} &= \text{E}\boldsymbol{\mathcal{D}}_i = \text{diag}\left\{\sum_{l=1}^{N} c_{l,1} q_l R_{u,l}, \cdots, \sum_{l=1}^{N} c_{l,N} q_l R_{u,l}\right\} \\
\mathcal{G} &= \text{E}\left[\boldsymbol{\mathcal{G}}_i \boldsymbol{\mathcal{G}}_i^*\right] = \mathcal{C}^T \text{diag}\left\{q_1^2 \sigma_{v,1}^2 R_{u,1}, \cdots, q_N^2 \sigma_{v,N}^2 R_{u,N}\right\} \mathcal{C}.
\end{aligned}$$

where the notation col$\{\cdots\}$ denotes a column vector and the notation diag$\{\cdots\}$ denotes a diagonal matrix. Also,

$$\begin{aligned}
\hat{\boldsymbol{\mathcal{D}}}_i &= \begin{bmatrix} I_{MN} - \mathcal{M}\boldsymbol{\mathcal{D}}_i & \boldsymbol{0} \\ \boldsymbol{0} & I_{(P-1)MN} \end{bmatrix} \quad \hat{\boldsymbol{\mathcal{G}}}_i = \begin{bmatrix} \mathcal{M}\boldsymbol{\mathcal{G}}_i \\ \boldsymbol{0} \end{bmatrix} \\
\hat{\mathcal{A}} &= \begin{bmatrix} \mathcal{A}^T & \boldsymbol{0} \\ \boldsymbol{0} & I_{(P-1)MN} \end{bmatrix} \quad \hat{\mathcal{B}} = \begin{bmatrix} K \\ I_{(P-1)MN} \quad \boldsymbol{0} \end{bmatrix} \\
K &= [\text{diag}\{g_{1,0},\cdots,g_{N,0}\},\cdots,\text{diag}\{g_{1,P-1},\cdots,g_{N,P-1}\}] \otimes I_M.
\end{aligned}$$

Then, expression (26) leads to:

$$\begin{aligned}
\tilde{\boldsymbol{\phi}}_i &= \hat{\boldsymbol{\mathcal{D}}}_i \tilde{\boldsymbol{w}}_{i-1} - \hat{\boldsymbol{\mathcal{G}}}_i \\
\tilde{\boldsymbol{\psi}}_i &= \hat{\mathcal{B}}\tilde{\boldsymbol{\phi}}_i \\
\tilde{\boldsymbol{w}}_i &= \hat{\mathcal{A}}\tilde{\boldsymbol{\psi}}_i
\end{aligned} \quad (27)$$

or, equivalently,

$$\tilde{\boldsymbol{w}}_i = \hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\boldsymbol{\mathcal{D}}}_i\tilde{\boldsymbol{w}}_{i-1} - \hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\boldsymbol{\mathcal{G}}}_i. \quad (28)$$

Let

$$\hat{\mathcal{D}} = \mathrm{E}\hat{\boldsymbol{\mathcal{D}}}_i = \begin{bmatrix} I_{MN} - \mathcal{M}\mathcal{D} & \mathbf{0} \\ \mathbf{0} & I_{(P-1)MN} \end{bmatrix}$$

$$\hat{\mathcal{G}} = \mathrm{E}\left[\hat{\boldsymbol{\mathcal{G}}}_i\hat{\boldsymbol{\mathcal{G}}}_i^*\right] = \begin{bmatrix} \mathcal{M}\mathcal{G}\mathcal{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

4.1 Mean-Square Analysis

We analyze the mean-square performance of the algorithm by following the energy conservation argument of [9]. Evaluating the weighted norm of $\tilde{\boldsymbol{w}}_i$ in (28) we obtain:

$$\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathrm{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\hat{\boldsymbol{\mathcal{D}}}_i\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\Sigma\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\boldsymbol{\mathcal{D}}}_i}^2 + \mathrm{E}\left[\hat{\boldsymbol{\mathcal{G}}}_i^*\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\Sigma\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\boldsymbol{\mathcal{G}}}_i\right] \quad (29)$$

where Σ is any Hermitian positive definite matrix. For tractable analysis, we introduce the independence assumption:

Assumption: All regressors $\boldsymbol{u}_{k,i}$ are spatially and temporally independent.

Using the above assumption, we can rewrite (29) as:

$$\mathrm{E}\,\|\tilde{w}_i\|_{\Sigma}^2 = \mathrm{E}\,\|\tilde{w}_{i-1}\|_{\Sigma'}^2 + \mathrm{Tr}\left[\Sigma\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\mathcal{G}}\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\right] \quad (30)$$

where $\Sigma' = \mathrm{E}\left(\hat{\boldsymbol{\mathcal{D}}}_i\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\Sigma\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\boldsymbol{\mathcal{D}}}_i\right)$. Let

$$\sigma = \mathrm{vec}\,(\Sigma) \quad \text{and} \quad \Sigma = \mathrm{vec}^{-1}\,(\sigma) \quad (31)$$

where the vec$\{\cdot\}$ notation replaces an $M \times M$ arbitrary matrix by an $M^2 \times 1$ column vector or an $M^2 \times 1$ column vector by an $M \times M$ matrix. To represent σ' as a function of σ, we use the following kronecker product property:

$$\mathrm{vec}\,(P\Sigma Q) = \left(Q^T \otimes P\right)\mathrm{vec}\,(\Sigma). \quad (32)$$

Using (32), we have a relation between σ' and σ as follows:

$$\sigma' = \mathrm{vec}\,(\Sigma') = F\sigma \quad (33)$$

where

$$F = \mathrm{E}\left(\hat{\boldsymbol{\mathcal{D}}}_i^T\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T \otimes \hat{\boldsymbol{\mathcal{D}}}_i^T\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\right). \quad (34)$$

Moving forward, we use the simpler notation $\|\tilde{w}\|_{\sigma}^2$ instead of $\|\tilde{w}\|_{\Sigma}^2$. Using the property $\mathrm{Tr}\,(\Sigma X) = \mathrm{vec}\,(X^T)^T\sigma$, in steady-state, (30) is rewritten as:

$$\mathrm{E}\,\|\tilde{w}_{\infty}\|_{(I-F)\sigma}^2 = \left[\mathrm{vec}\left(\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\mathcal{G}}^T\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\right)\right]^T\sigma. \quad (35)$$

The mean-square deviation (MSD) and the excess mean-square error (EMSE) at node k are defined by:

$$\mathrm{MSD}_k = \mathrm{E}\,\|\boldsymbol{w}_{k,i} - w^o\|^2 \quad \mathrm{EMSE}_k = \mathrm{E}\,|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{w}}_{k,i-1}|^2 \quad (36)$$

as $i \to \infty$. We can calculate these values by using proper weighting matrices. Let us define the vectorized version of the matrices for the MSD and the EMSE as follows:

$$m_k = \mathrm{vec}\begin{bmatrix} \mathrm{diag}\,(e_k) \otimes I_M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$r_k = \mathrm{vec}\begin{bmatrix} \mathrm{diag}\,(e_k) \otimes R_{u,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

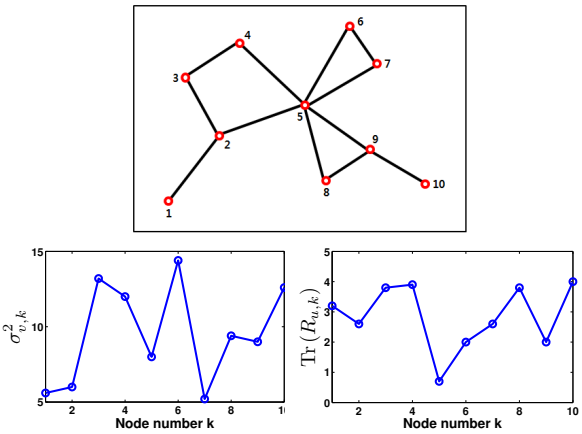

Figure 2: Network topology (top), noise variance $\sigma_{v,k}^2$ (bottom, left) and trace of regressor covariance $\mathrm{Tr}\,(R_{u,k})$ (bottom, right) for $N = 10$ nodes.

where e_k is $N \times 1$ column vector which has a unit entry at position k and zeros elsewhere. Then the MSD and the EMSE become:

$$\mathrm{MSD}_k = \mathrm{E}\,\|\tilde{w}_{\infty}\|_{m_k}^2 = \left[\mathrm{vec}\left(\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\mathcal{G}}^T\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\right)\right]^T(I-F)^{-1}m_k \quad (37)$$

$$\mathrm{EMSE}_k = \mathrm{E}\,\|\tilde{w}_{\infty}\|_{r_k}^2 = \left[\mathrm{vec}\left(\hat{\mathcal{A}}\hat{\mathcal{B}}\hat{\mathcal{G}}^T\hat{\mathcal{B}}^T\hat{\mathcal{A}}^T\right)\right]^T(I-F)^{-1}r_k. \quad (38)$$

The network MSD and EMSE are defined as the average MSD and EMSE over all nodes in the network [4]:

$$\mathrm{MSD}^{\mathrm{network}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{MSD}_k = \frac{1}{N}\mathrm{E}\,\|\tilde{\boldsymbol{w}}_{\infty}\|^2\begin{bmatrix} I_{MN} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (39)$$

$$\mathrm{EMSE}^{\mathrm{network}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{EMSD}_k = \frac{1}{N}\mathrm{E}\,\|\tilde{\boldsymbol{w}}_{\infty}\|^2\begin{bmatrix} \mathrm{diag}\{R_{u,1},\cdots,R_{u,N}\} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (40)$$

In (37) and (38), the weighting matrix $\hat{\mathcal{B}}$ is added compared to the conventional diffusion LMS [4]. When we use only one previous weight vector for update (i.e., $P = 1$), $\hat{\mathcal{B}}$ becomes the identity matrix and (37) and (38) reduce to the same form as in diffusion LMS [4].

5. SIMULATIONS

In this section, we illustrate the performance of the proposed spatio-temporal diffusion LMS algorithms and also compare them with the spatial diffusion LMS algorithm from [4]. For the simulation, we assume a channel identification scenario of an FIR model with channel length of 10 for every node. Fig. 2 depicts the network topology with $N = 10$ nodes and the network statistical profile of noise variance and signal power. step-size μ_k is set to 0.05 and all simulations are obatined by averaging 500 independent experiments. The number of temporal memory P is 3 for Figs. 3 and 5. We use relative-degree weights for the diffusion matrix A and metropolis weights for the adaptation matrix C, which are the same choices used in [4]. Also, we set $q_l = 1/P$ for every l and $g_{k,j} = 1/P$ for every k and j. In the proposed algorithms, we set μ_k to $0.05P$ to make the adaptation step

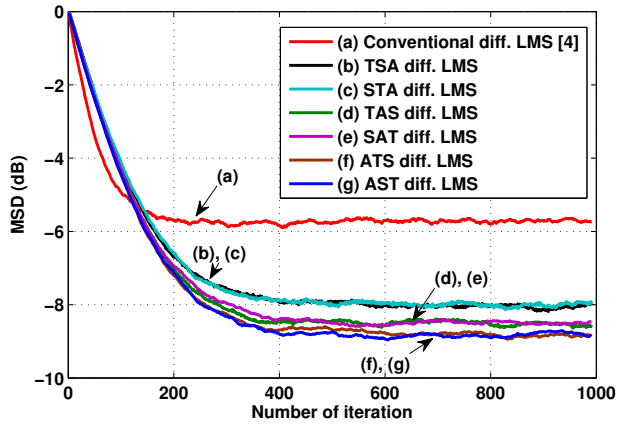

Figure 3: Transient network MSD for the conventional diffusion LMS and the 6 combinations of spatio-temporal diffusion LMS.

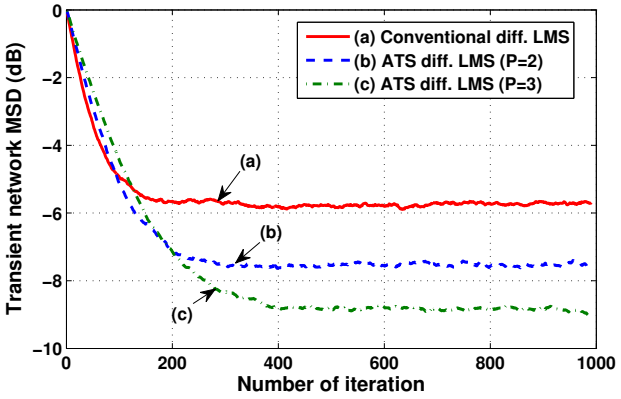

Figure 4: Trandient network MSD of the proposed ATS diffusion LMS with various $P = 2, 3$, compared to the conventional diffusion LMS algorithm.

have the same weighting as in [4] for comparison purposes ($\mu_k q_l = 0.05$).

Fig. 3 shows the transient network MSD curves for the proposed diffusion algorithms. The performance is dependent on the order by which the A, T, and S steps are performed. Algorithms (f) and (g), which perform adaptation first have the lowest steady-state error regardless of the order of the S and T steps. In contrast, algorithms (b) and (c), which perform adaptation last, are more efficient than the conventional diffusion LMS in steady-state although the convergence rate is a little slower.

In Fig. 4, we show the effect of P. We use the ATS diffusion LMS algorithm, which is the best combination for the simulation. As P increases, the steady-state error decreases but the convergence rate becomes slightly slower.

Fig. 5 shows the steady-state MSD at every node in the spatial network, and compares with the theoretical results from expression (37). In this figure, we also simulate with ATS diffusion LMS. The steady-state values are obtained by averaging over 500 samples after convergence. The simulation results match well the theoretical values.

6. CONCLUSIONS

We proposed distributed adaptation algorithms that are able to process local data both spatially and temporally. Several variations are possible depending on how the operations

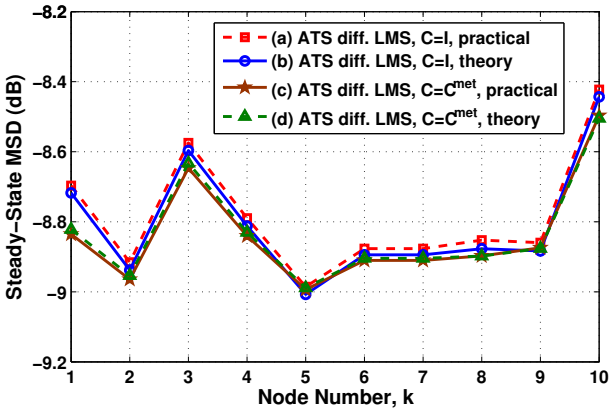

Figure 5: Steady-state MSD of the proposed ATS diffusion LMS at each node.

of adaptation, spatial combination, and temporal combination are ordered. The algorithm that adapts first, and then combines the data (regardless of the order of the spatial or temporal processing) performs the best.

REFERENCES

[1] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. on Fund. of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 8, pp. 1504–1510, August 2007.

[2] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. on Signal Processing*, vol. 55, no.8, pp. 4064–4077, August 2007.

[3] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[4] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[5] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, September 2010.

[6] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[7] F. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, September 2010.

[8] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. on Signal Processing*, vol. 57, no. 1, pp. 355-369, January 2009.

[9] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.