DIFFUSION LMS FOR SOURCE AND PROCESS ESTIMATION IN SENSOR NETWORKS

Reza Abdolee, Benoit Champagne∗

McGill University
Dept. of Electrical and Computer Engineering
Montreal, QC, H3A 2A7, Canada

Ali H. Sayed†

University of California
Dept. of Electrical Engineering
Los Angeles, CA, 90095, USA

ABSTRACT

We develop a least mean-squares (LMS) diffusion strategy for sensor network applications where it is desired to estimate parameters of physical phenomena that vary over space. In particular, we consider a regression model with space-varying parameters that captures the system dynamics over time and space. We use a set of basis functions such as sinusoids or B-spline functions to replace the space-variant (local) parameters with space-invariant (global) parameters, and then apply diffusion adaptation to estimate the global representation. We illustrate the performance of the algorithm via simulations.

Index Terms— Diffusion adaptation, Distributed adaptive estimation, sensor networks, space-varying parameters, population dispersal, fluid-flow.

1 Introduction

Adaptive diffusion strategies were shown to serve as efficient and powerful learning mechanisms for solving estimation problems in a distributed manner and in real-time over networks [1, 2]. These strategies were used in [3–5] to model several instances of organized behavior encountered in nature, such as bird flight formations, fish schooling, bee swarming, and bacteria mobility.

In these articles and other related works in the area of distributed optimization, the parameters of interest were assumed to be *space-invariant*. There are, however, important applications where the underlying parameters of interest vary over space, such as in the monitoring of fluid flow in underground porous media [6], the tracking of population dispersal [7], and the estimation of distributed sources and processes in physical phenomena [8].

In this paper, we present a system model with space-varying parameters that can adequately capture the behavior of such phenomena, and propose a distributed adaptive strategy of diffusion type to estimate and track the corresponding parameters. The proposed algorithm differs from previous approaches [9–11] in that it is both adaptive and distributed. The adaptive feature of the algorithm is instrumental in situations where the space-varying parameters change over time, and its distributed feature is attractive for networking and decentralized system models. Compared with previous works in distributed adaptive filtering, this work extends diffusion strategies to models with space-dependent parameters, and rank-deficient local correlation matrices.

Notation: Boldface font is used for random variables and normal font is used for deterministic quantities. For complex vectors and matrices, $(\cdot)^*$ denotes complex conjugate transposition. I_M denotes the identity matrix of size $M \times M$.

2 Problem Formulation

We consider a connected network with N nodes where each node k measures data $\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$ that satisfy a linear regression model of the form:

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i} h_k^o + \mathbf{v}_k(i) \qquad k \in \{1, 2, \dots, N\} \tag{1}$$

In this model, the indices k and i represent space and time, respectively, $\mathbf{d}_k(i) \in \mathbb{C}$ is the measurement signal, $\mathbf{u}_{k,i} \in \mathbb{C}^{1\times M}$ is the regressor, $h_k^o \in \mathbb{C}^{M\times 1}$ represents the unknown space-dependent parameter, and $\mathbf{v}_k(i) \in \mathbb{C}$ is measurement noise. This regression model is useful enough to describe the behavior of physical phenomena whose parameters are space-varying. For this model, we assume the regression data $\{\mathbf{u}_{k,i}\}$ are temporally white and independent over space with covariance matrices $R_{u,k} = \mathrm{E}[\mathbf{u}_{k,i}^* \mathbf{u}_{k,i}] > 0$. The noise $\mathbf{v}_k(i)$ is a zero-mean random process with variance $\sigma_{v_k}^2$, independent of $\mathbf{v}_\ell(j)$ for $\ell \neq k$ or $j \neq i$, and also independent of $\mathbf{u}_{\ell,j}$ for all i, j and k, ℓ.

Our objective is to propose a distributed and adaptive algorithm that can estimate the space-dependent parameters h_k^o, for $k \in \{1, 2, \dots, N\}$. To achieve this objective, the global cost function of the network is taken as:

$$J(h_1, \cdots, h_N) = \sum_{k=1}^{N} J_k(h_k) \tag{2}$$

where

$$J_k(h_k) = \mathrm{E}|\mathbf{d}_k(i) - \mathbf{u}_{k,i} h_k|^2 \tag{3}$$

∗This work was performed while R. Abdolee was a visiting Ph.D. student in the Adaptive Systems Lab. at the Univ. of California, Los Angeles (UCLA). The work of R. Abdolee and B. Champagne was supported by the Fonds Québécois de la recherche sur la nature et les technologies (FQRNT) and the Natural Sciences and Eng. Research Council (NSERC) of Canada.

†The work of A. H. Sayed was supported in part by NSF grants CCF-1011918 and CCF-0942936.

is the cost function at node k. To proceed, we express the m^{th} component of h_k as a linear combination of a set of N_b basis functions $\{b_j(k) : j = 1, \cdots, N_b\}$, i.e.,

$$h_k(m) = \sum_{j=1}^{N_b} W_{mj} b_j(k), \quad \text{for} \quad k \in \{1, \cdots, N\} \tag{4}$$

where W_{mj} are combination (expansion) coefficients. Observe that the basis functions can depend on the space variable, k. These functions can be chosen, e.g., as sinusoids or B-splines. Equation (4) can be expressed in matrix form by introducing

$$b_k = [b_1(k), \ldots, b_{N_b}(k)]^T \tag{5}$$

Then, $h_k = W b_k$, where

$$W \triangleq \begin{bmatrix} W_{11} & W_{12} & \ldots & W_{1N_b} \\ W_{21} & W_{22} & \ldots & W_{2N_b} \\ \vdots & \vdots & \ldots & \vdots \\ W_{M1} & W_{M2} & \ldots & W_{MN_b} \end{bmatrix} \tag{6}$$

For computational convenience, we rearrange W as a column vector w by stacking the columns of W^T on top of each other, i.e., $w = \text{vec}(W^T)$. In this way, we can write:

$$h_k = B_k w, \qquad \text{for} \quad k \in \{1, 2, \ldots N\} \tag{7}$$

where $B_k \in \mathbb{R}^{M \times MN_b}$ is the block diagonal matrix and computed as

$$B_k \triangleq I_M \otimes b_k^T \tag{8}$$

and $\otimes$ is the Kronecker product. We substitute (7) into (2) to replace the space-dependent parameter h_k with the global parameter w. Thus, we get:

$$J(w) = \sum_{k=1}^{N} \text{E}|\mathbf{d}_k(i) - \mathbf{u}_{k,i} B_k w|^2 \tag{9}$$

Now, the goal is to determine the optimal global parameter w^o that minimizes (9) in a distributed manner and arrive at $h_k^o = B_k w^o$.

3 Distributed Adaptive Optimization

Let $\mathcal{N}_k \triangleq \{\nu_{k,1}, \nu_{k,2}, \ldots \nu_{k,n_k}\}$ denote the neighborhood of node k, where $\nu_{k,\ell} \in \{1, \ldots, N\}$ denote the indices of the neighbors, and $n_k = |\mathcal{N}_k|$ is the cardinality of set $\mathcal{N}_k$. We also introduce combination matrices C and $A \in \mathbb{R}^{N \times N}$ with nonnegative entries, such that

$$c_{\ell,k} = a_{\ell,k} = 0, \quad \text{if } l \notin \mathcal{N}_k, \text{ and } \quad C\mathbb{1} = A^T\mathbb{1} = \mathbb{1} \tag{10}$$

where $\mathbb{1} \in \mathbb{R}^{N \times 1}$ is a column vector with unit entries. That is, C is a right-stochastic matrix and A is a left-stochastic matrix. Following the technique developed in [2], we can derive the diffusion algorithm (listed below) where nodes cooperate locally to estimate w^o. In this algorithm, $\mu_k > 0$ is the step-size parameter, and $\boldsymbol{\psi}_{k,i}$ is an intermediate estimate at node k at time i. In this implementation, at every iteration i, every node k first uses data $\{\mathbf{d}_{\ell,i}, \mathbf{u}_{\ell,i}\}$ from its neighborhood to update its existing estimate $\mathbf{w}_{k,i-1}$ to an intermediate value $\boldsymbol{\psi}_{k,i}$. Subsequently, the intermediate estimates from across the neighborhood of node k are combined to yield $\mathbf{w}_{k,i}$, from which the space-dependent parameter $\mathbf{h}_{k,i}$ is determined. In this algorithm, the weighting and combination coefficients $\{c_{\ell,k}, a_{\ell,k}\}$ can be chosen according to different rules, e.g., Metropolis or relative degree rules [2]. They can also be computed adaptively to improve the estimation quality [12].

Diffusion LMS for models with space-varying parameters

$$\boldsymbol{\psi}_{k,i} = \mathbf{w}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \big[\mathbf{u}_{\ell,i}^* \mathbf{d}_{\ell,i} \otimes b_\ell - (\mathbf{u}_{\ell,i}^* \mathbf{u}_{\ell,i} \otimes b_\ell) B_\ell \mathbf{w}_{k,i-1}\big]$$

$$\mathbf{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{\psi}_{\ell,i}$$

$$\mathbf{h}_{k,i} = B_k \mathbf{w}_{k,i}$$

4 Performance Analysis

For space limitations, we only comment here on the convergence of the algorithm in the mean sense. We note that compared to the analysis in [1,2], we now need to account for the fact that the correlation matrix $\mathcal{R}_{u,k} = (R_{u,k} \otimes b_k) B_k$ can be rank-deficient even when $R_{u,k}$ is full rank.

4.1 Mean Convergence

We define the local error vectors

$$\tilde{\mathbf{w}}_{k,i} = w^o - \mathbf{w}_{k,i} \tag{11}$$

$$\tilde{\boldsymbol{\psi}}_{k,i} = w^o - \boldsymbol{\psi}_{k,i} \tag{12}$$

and the global error vectors

$$\tilde{\boldsymbol{\psi}}_i = \text{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \cdots, \tilde{\boldsymbol{\psi}}_{N,i}\} \tag{13}$$

$$\tilde{\mathbf{w}}_i = \text{col}\{\tilde{\mathbf{w}}_{1,i}, \cdots, \tilde{\mathbf{w}}_{N,i}\} \tag{14}$$

where in this representation, col$\{\cdot\}$ stacks its arguments into a column vector. In addition, we introduce the extended weighting matrices $\mathcal{A} = A \otimes I_{MN_b}$ and $\mathcal{C} = C \otimes I_{MN_b}$. We also define

$$\mathcal{M} = \text{diag}\{\mu_1 I_{MN_b}, \cdots, \mu_N I_{MN_b}\} \tag{15}$$

$$\boldsymbol{g}_i = \mathcal{C}^T \cdot \text{col}\{(\mathbf{u}_{1,i}^* \otimes b_1)\mathbf{v}_1(i), \cdots, (\mathbf{u}_{N,i}^* \otimes b_N)\mathbf{v}_N(i)\} \tag{16}$$

$$\boldsymbol{\mathcal{D}}_i = \text{diag}\Big\{\sum_{\ell=1}^{N} c_{\ell j}(\mathbf{u}_{\ell,i}^* \mathbf{u}_{\ell,i} \otimes b_\ell) B_\ell : j = 1, \cdots, N\Big\} \tag{17}$$

Then, it can be verified that the weight-error vector across the network evolves according to the recursion:

$$\tilde{\mathbf{w}}_i = \mathcal{A}^T (I_{NMN_b} - \mathcal{M}\boldsymbol{\mathcal{D}}_i) \tilde{\mathbf{w}}_{i-1} - \mathcal{A}^T \mathcal{M} \boldsymbol{g}_i \tag{18}$$

Taking the expectation of both sides yields:

$$\mathrm{E}[\tilde{\mathbf{w}}_i] = \mathcal{A}^T(I_{NMN_b} - \mathcal{M}\mathcal{D})\mathrm{E}[\tilde{\mathbf{w}}_{i-1}] \tag{19}$$

where $\mathcal{D} = \mathrm{E}[\boldsymbol{\mathcal{D}}_i]$ and is given by:

$$\mathcal{D} = \mathrm{diag}\Big\{\sum_{\ell=1}^{N} c_{\ell,1}\,\mathcal{R}_{u,\ell}, \ldots, \sum_{\ell=1}^{N} c_{\ell,N}\,\mathcal{R}_{u,\ell}\Big\} \tag{20}$$

Considering the fact that the matrices $\mathcal{R}_{u,\ell}$ can be rank-deficient, it can be verified that if the step-sizes are chosen to satisfy:

$$0 < \mu_k < \frac{2}{\lambda_{\max}\big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\mathcal{R}_{u,\ell}\big)}, \quad \text{for} \quad k = 1, \ldots, N \tag{21}$$

then the mean weight-error vector, $\mathrm{E}[\tilde{\mathbf{w}}_i]$, would remain bounded as $i \to \infty$.

5 Simulation Results

Example 1 (Source Estimation): In this example, we demonstrate the application of the algorithm for process estimation in PDE systems. We consider a network with $N = 10$ sensors that are uniformly placed over an oil reservoir with $L = 1$ normalized distance. Assume the dimensionless fluid pressure distribution can be described as [10]:

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial}{\partial x}\big(\theta(x)\frac{\partial f(x,t)}{\partial x}\big) + h(x,t) \tag{22}$$

where $(x,t) \in [0,L] \times [0,T]$ denote the space and time variables, respectively, $f(x,t)$ represents the fluid pressure distribution, and $\theta(x)$ is the known transmissivity parameter. The objective is to estimate the space-varying process $h(x,t)$. The boundary and initial conditions of the reservoir are

$$f(x,t)|_{t=0} = f(x,t)|_{x=0} = f(x,t)|_{x=L} = 0 \tag{23}$$

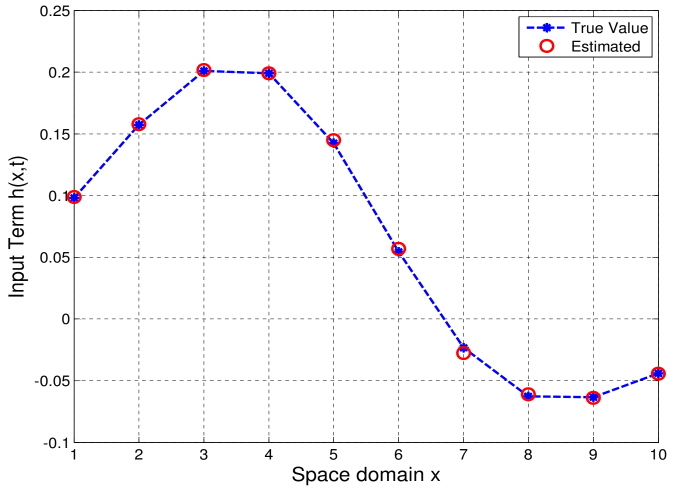
We employ the finite difference method (FDM) to discretize the PDE over time and space. Let $\Delta x = L/N$ and $x_k = k\Delta x$ for $k \in \mathcal{K} \triangleq \{0,1,2,\ldots,N\}$. Similarly, let $\Delta t = T/P$ and $t_i = i\Delta t$ for $i \in \mathcal{I} \triangleq \{0,1,2,\ldots,P\}$. We further introduce the sampled value of the pressure distribution $f_k(i) \triangleq f(x_k,t_i)$, sampled input $h_k(i) \triangleq h(x_k,t_i)(\Delta t)$, and define $\theta_k \triangleq \theta(x_k)$. In this way, we can express the measurement samples as

$$\mathbf{z}_k(i) = f_k(i) + \mathbf{v}_k(i) \tag{24}$$

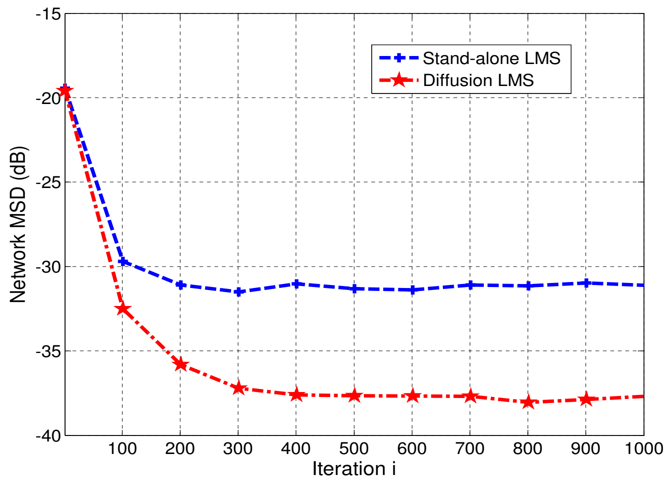
For estimation of $h_k(i)$ in the discretized model, we arrive at the optimization problem given by (2) with $J_k(h_k(i)) = \mathrm{E}|\mathbf{d}_k(i) - h_k(i)|^2$ and $M = 1$ where $\mathbf{d}_k(i) = \mathbf{z}_k(i) - p_{k,i}g_k$. In this equation, g_k and $p_{k,i}$, for $k = 1,\ldots,N$, are defined as:

$$g_k \triangleq [g_k(1), g_k(2), g_k(3)]^T \tag{25}$$

$$p_{k,i} \triangleq [f_{k-1}(i-1), f_k(i-1), f_{k+1}(i-1)] \tag{26}$$



(a) Estimated source and sink over the field.



(b) Network learning behavior.

Fig. 1. Performance of diffusion LMS in process estimation.

where the entries of g_k are $g_k(1) = \rho(\frac{1}{4}\theta_{k-1} + \theta_k - \frac{1}{4}\theta_{k+1})$, $g_k(2) = 1 - 2\rho\theta_k$ and $g_k(3) = \rho(-\frac{1}{4}\theta_{k-1} + \theta_k + \frac{1}{4}\theta_{k+1})$ with $\rho \triangleq \frac{\Delta t}{\Delta x^2}$ and $\theta_k = k/N$. For this example, suppose the unknown input $h_k(i)$ comprises of a sink $h_{1,k}$ and a source $h_{2,k}$ that are independent of time. We therefore can omit the time index in $h_k(i)$ and write $h_k = h_{1,k} + h_{2,k}$. Moreover, assume the distribution of the sink and the source over space are given as

$$h_{1,k} = 0.25e^{-0.1(k-4)^2}, \quad h_{2,k} = -0.125e^{-0.1(k-8)^2} \tag{27}$$

To represent h_k as a constant global vector over the network, we use equation (4) with $N_b = 16$ sinusoidal basis functions, i.e., $b_j(k) = \cos\frac{(j-1)k\pi}{2N}$ for odd j, and $\sin\frac{jk\pi}{2N}$ otherwise. Now, the objective is to estimate the parameter vector $w^o = [w_1,\ldots,w_{N_b}]$ by using the developed diffusion algorithm. For this, we select C and A according to the Metropo-

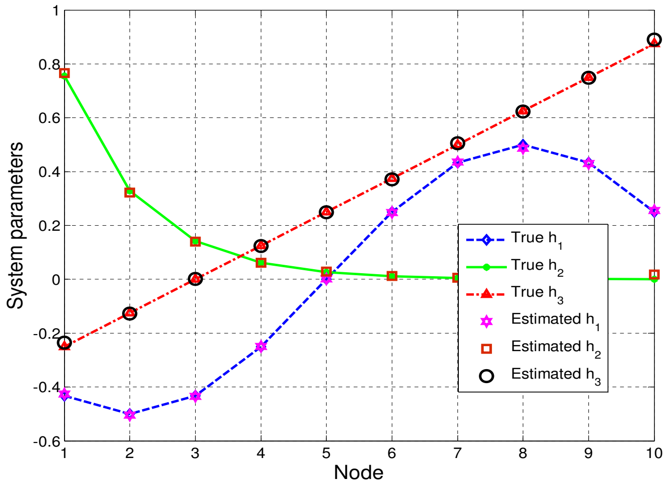


Fig. 2. Performance of the algorithm in the estimation of h_k.

lis and relative degree criteria [2]. All nodes are initialized at zero and operate with step-size of $\mu_k = 0.05$. The SNR at each node is randomly chosen from the interval $[25, 30]$dB.

Fig.1(a), compares the estimated and the true values of the pressure and the locations of the source and sink in steady-state. As it is shown, the estimated and the true value of h_k coincide well. Fig. 1(b) shows the network transient behavior in terms of MSD, and compares the results of diffusion LMS with that of the average of stand-alone LMS filters over the network. Evidently, the performance of diffusion-LMS is superior.

Example 2 (Parameter Estimation): We consider a network as in Example 1, with the same topology and same parameters N, C, A and μ_k. However, for this example, we use the system model given in (1), and choose

$$h_k = \big[-0.5\sin(2\pi k\Delta x), 4e^{-10k\Delta x}, (1.5k\Delta x - 0.5)\big]^T \quad (28)$$

where $\Delta x = 1/N$. Here, we consider quadratic spline basis functions with $N_b = 20$ controlling points, and initialize each filter at zero. For $k \in \{1, \cdots, N\}$, we choose $\mathrm{Tr}(R_{u,k})$ as

$$[1, 1.4, 4, 1.9, 1, 2.4, 2.8, 2, 3.2, 3.9] \quad (29)$$

and the network SNR= $[18, 16, 15, 24, 22, 20, 15, 22, 19, 23]$dB . The SNR at each that is computed by using

$$\mathrm{SNR}(k) = \mathrm{E}\|\mathbf{u}_k h_k\|^2/\sigma^2_{v_k} \quad (30)$$

For this example, the results are presented in Fig. 5. From this figure, we observe that the estimated system parameters coincide well with the true values.

6 Conclusion

We proposed a diffusion strategy for the distributed estimation of space-varying parameters. We illustrated the operation of the algorithm by considering two examples.

7 References

[1] C. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, pp. 3122–3136, July 2008.

[2] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035–1048, Mar. 2010.

[3] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. on Signal Processing*, vol. 59, pp. 2038–2051, May 2011.

[4] S. Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Selected Topics on Signal Processing*, vol. 5, pp. 649–664, Aug. 2011.

[5] J. Li and A. H. Sayed, "Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing," *EURASIP Journal on Advances in Signal Processing*, 2012:18, doi:10.1186/1687-6180-2012-18,2012.

[6] T. Lee and J. Seinfeld, "Estimation of two-phase petroleum reservoir properties by regularization," *J. of Computational Physics*, vol. 69, pp. 397–419, Feb. 1987.

[7] E. Holmes, M. Lewis, J. Banks, and R. Veit, "Partial differential equations in ecology: Spatial interactions and population dynamics," *Ecology*, vol. 75, pp. 17–29, Jan. 1994.

[8] M. Özisik and H. Orlande, *Inverse Heat Transfer: Fundamentals and Applications*. Hemisphere Pub., 2000.

[9] G. Richter, "Numerical identification of a spatially-varying diffusion coefficient," *Math. Comput.*, vol. 36, pp. 375–386, Apr. 1981.

[10] C. Chung and C. Kravaris, "Identification of spatially discontinuous parameters in second-order parabolic systems by piecewise regularisation," *Inverse Problems*, vol. 4, pp. 973–994, Oct. 1988.

[11] V. Isakov and S. Kindermann, "Identification of the diffusion coefficient in a one-dimensional parabolic equation," *Inverse Problems*, vol. 16, pp. 665–680, June 2000.

[12] 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. 8, pp. 4795–4810, Sep. 2010.