DIFFUSION LMS FOR CLUSTERED MULTITASK NETWORKS

Jie Chen ⋆ *Cédric Richard* ⋆ *Ali H. Sayed* †

⋆ Université de Nice Sophia-Antipolis, CNRS, France
† University of California, Los Angeles, USA
{jie.chen, cedric.richard}@unice.fr sayed@ee.ucla.edu

ABSTRACT

Recent research works on distributed adaptive networks have intensively studied the case where the nodes estimate a common parameter vector collaboratively. However, there are many applications that are multitask-oriented in the sense that there are multiple parameter vectors that need to be inferred simultaneously. In this paper, we employ diffusion strategies to develop distributed algorithms that address clustered multitask problems by minimizing an appropriate mean-square error criterion with ℓ_2-regularization. Some results on the mean-square stability and convergence of the algorithm are also provided. Simulations are conducted to illustrate the theoretical findings.

Index Terms— Multitask learning, distributed optimization, diffusion strategy, collaborative processing, regularization

1. INTRODUCTION

Distributed adaptive learning is an attractive and challenging subject within the area of multi-agent networks. It leads to algorithms that are able to continuously adapt and learn, and that are particularly suitable for tracking concept drifts in the measured data. The resulting distributed algorithms offer an important alternative to centralized solutions with advantages resulting from scalability, robustness, and decentralization. Several useful distributed strategies for online parameter estimation have been proposed in the literature, including consensus strategies [1–3], incremental strategies [4–7], and diffusion strategies [8–13]. Incremental techniques require the determination of a cyclic path that runs across all nodes, which is generally an NP-hard problem. Besides, incremental solutions are sensitive to link failures. On the other hand, diffusion strategies are attractive since they are scalable, robust, and enable continuous adaptation and learning. In addition, for data processing over adaptive networks, diffusion strategies have been shown to have superior stability and performance ranges [14] than consensus-based implementations. Accessible overviews of recent results on diffusion adaptation can be found in [8, 9].

An inspection of the literature on distributed algorithms shows that most existing works focus primarily, though not exclusively [15–17], on the case where the nodes have to estimate a single parameter vector collaboratively. We refer to problems of this type as *single-task* problems. However, many problems of interest happen to be *multitask*-oriented in the sense that there are multiple parameter vectors to be inferred simultaneously and in a collaborative manner. Multitask learning problems have been studied by the machine learning community in several contexts, including web page categorization [18], web-search ranking [19], disease progression modeling [20], among other areas. Clearly, this concept is also relevant in the context of estimation over adaptive networks. Initial investigations along these lines for the traditional diffusion strategy appear in [15, 21]. In this article, we consider the situation where there are connected clusters of nodes, and each cluster has a parameter vector to estimate. The estimation still needs to be performed cooperatively across the network because the data across the clusters may be correlated and, therefore, cooperation across clusters can be beneficial. The aim of this paper is to derive a diffusion strategy that is able to solve the clustered multitask estimation problem, and to provide analytical results for convergence in terms of mean weight error and mean-square error.

Notation. Small letters x denote scalars, and boldface small letters $\boldsymbol{x}$ denote column vectors. Boldface capital letters $\boldsymbol{R}$ represent matrices, and the operator $(\cdot)^\top$ denotes matrix transposition. $\boldsymbol{I}_N$ denotes the $N \times N$ identity matrix. $\mathcal{N}_k$ denotes the neighbors of node k, including k, whereas $\mathcal{N}_k^-$ denotes the neighbors of node k, excluding k. $\mathcal{C}_i$ is the cluster i, i.e., index set of nodes in the i-th cluster. $\mathcal{C}(k)$ denotes the cluster to which node k belongs. Finally, $\otimes$ denotes the Kronecker product, and vec$(\cdot)$ stacks the columns of a matrix on top of each other into a vector.

2. NETWORK MODEL AND PROBLEM FORMULATION

2.1. Clustered multitask network

Consider a connected network consisting of N nodes. The problem is to estimate an $L \times 1$ unknown vector at each node k from collected data. Node k has access to time sequences $\{d_k(n), \boldsymbol{x}_k(n)\}$, with $d_k(n)$ representing the reference signal, and $\boldsymbol{x}_k(n)$ denoting an $L \times 1$ regression vector with covariance matrix $\boldsymbol{R}_{x,k} = E\{\boldsymbol{x}_k(n)\boldsymbol{x}_k^\top(n)\} > 0$. The data at node k are assumed to be related via the linear model:

$$d_k(n) = \boldsymbol{x}_k^\top(n)\, \boldsymbol{w}_k^\star + z_k(n) \qquad (1)$$

where $\boldsymbol{w}_k^\star$ is an unknown parameter vector at node k, and $z_k(n)$ is a zero-mean, i.i.d. noise that is independent of every other signal and has variance $\sigma_{z,k}^2$. We assume that there are Q clusters and, therefore, Q tasks to be performed. We also assume that the nodes in the same cluster perform the same estimation task. The optimum parameter vectors $\boldsymbol{w}_k^\star$ are constrained to be equal within each cluster, but similarities between neighboring clusters are allowed to exist, namely,

$$\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}_q}^\star \qquad \text{for } \forall k \in \mathcal{C}_q \qquad (2)$$

$$\boldsymbol{w}_{\mathcal{C}_p}^\star \sim \boldsymbol{w}_{\mathcal{C}_q}^\star \qquad \text{if } \mathcal{C}_p,\ \mathcal{C}_q \text{ are connected} \qquad (3)$$

where p and q denote two cluster indexes, and $\sim$ represents a similarity relationship in some sense. The reader is referred to Fig. 1(a)

This work was partly supported by the CNRS, France (Display project, Mastodons). The work of A. H. Sayed was supported in part by NSF grant CCF-1011918.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

for an illustration showing a network with $N = 15$ nodes and $Q = 3$ clusters.

2.2. Problem formulation

Clustered multitask networks require that nodes in the same cluster estimate the same coefficient vector. We associate a mean-square error cost function, $J_k(\boldsymbol{w}_{\mathcal{C}(k)})$, with each node k such that

$$J_k(\boldsymbol{w}_{\mathcal{C}(k)}) = E\big\{|d_k(n) - \boldsymbol{x}_k^\top(n)\boldsymbol{w}_{\mathcal{C}(k)}|^2\big\}. \tag{4}$$

In order to promote similarities among adjacent clusters, appropriate regularization can be used. In this paper, we simply introduce the squared ℓ_2-norm as a possible regularizer, namely,

$$\Delta(\boldsymbol{w}_{\mathcal{C}(k)}, \boldsymbol{w}_{\mathcal{C}(\ell)}) = \|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2. \tag{5}$$

Combining (4) and (5) yields the following regularization problem at the level of the entire network:

$$\begin{aligned}\overline{J^{\text{glob}}}(\boldsymbol{w}_{\mathcal{C}_1},\dots,\boldsymbol{w}_{\mathcal{C}_Q}) &= \sum_{k=1}^{N} E\big\{|d_k(n) - \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}|^2\big\} \\ &+ \frac{\tau}{2}\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} \rho_{k\ell}\,\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2\end{aligned} \tag{6}$$

where the second term on the RHS of expression (6) promotes similarities between neighboring clusters, with non-negative strength parameter τ and non-negative weights $\rho_{k\ell}$. We seek a distributed solution to (6). For that purpose, we first associate with the i-th cluster, the following cost function

$$\begin{aligned}\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}) &= \sum_{k\in\mathcal{C}_i} E\big\{|d_k(n) - \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}|^2\big\} \\ &+ \frac{\tau}{2}\sum_{k\in\mathcal{C}_i}\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} (\rho_{k\ell}+\rho_{k\ell})\,\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2\end{aligned} \tag{7}$$

Note that for given $\boldsymbol{w}_{\mathcal{C}(\ell)}$ with $\ell \in \mathcal{N}_k\backslash\mathcal{C}(k)$, the costs in (6) and (7) have the same gradient vectors relative to $\boldsymbol{w}_{\mathcal{C}_i}$. In order that each node can solve the problem autonomously and adaptively using only local interactions, we shall derive a distributed iterative algorithm for solving (6) by considering (7) since both cost functions have the same gradient information.

3. DISTRIBUTED ADAPTIVE ESTIMATION ALGORITHM

3.1. Local cost decomposition and problem relaxation

We first note that a steepest-descent solution that is based on (7) will require every node in the network to have access to the statistical second-order moments of the data over its cluster. There are two problems with this scenario. First, nodes can only have access to information from their immediate neighborhood and the cluster of every node k may include nodes that are not direct neighbors of k. Second, nodes rarely have access to the data statistical moments; instead, they have access to data generated from distributions with these moments. Therefore, more is needed to enable a distributed solution that relies solely on local interactions within neighborhoods and that relies on measured data as opposed to statistical moments. To derive a distributed algorithm, we follow the approach of [9, 11]. The first step in this approach is to show how to express the cost (7) in terms of other local costs that only depend on data from neighborhoods.

We start by introducing an $N \times N$ right stochastic matrix $\boldsymbol{C}$ with non-negative entries $c_{\ell k}$ such that

$$\sum_{k=1}^{N} c_{\ell k} = 1, \quad \text{and} \quad c_{\ell k} = 0 \text{ if } k \notin \mathcal{N}_\ell \cap \mathcal{C}(\ell). \tag{8}$$

With these coefficients, we associate a local cost function of the following form with each node k:

$$J_k^{\text{loc}}(\boldsymbol{w}_{\mathcal{C}(k)}) = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k} E\big\{|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{w}_{\mathcal{C}(k)}|^2\big\} \tag{9}$$

In (9), note that $\boldsymbol{w}_{\mathcal{C}(k)} = \boldsymbol{w}_{\mathcal{C}(\ell)}$ because $\ell \in \mathcal{C}(k)$. To make the notation simpler, we shall write $\boldsymbol{w}_k$ instead of $\boldsymbol{w}_{\mathcal{C}(k)}$, and consequently $\boldsymbol{w}_k = \boldsymbol{w}_\ell$ for all $\ell \in \mathcal{C}(k)$. To take interactions among neighboring clusters into account, we modify (9) by associating a regularized local cost function with node k of the following form

$$\begin{aligned}\overline{J_k^{\text{loc}}}(\boldsymbol{w}_k) &= \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\, E\big\{|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_k|^2\big\} \\ &+ \frac{\tau}{2}\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} (\rho_{k\ell}+\rho_{\ell k})\|\boldsymbol{w}_k - \boldsymbol{w}_\ell\|^2.\end{aligned} \tag{10}$$

Observe that this local cost is now solely defined in terms of information that is available to node k from its neighbors. It can then be verified that the following relation between (10) and (7) holds:

$$\overline{J_{\mathcal{C}(k)}}(\boldsymbol{w}_k) = \overline{J_k^{\text{loc}}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{C}(k)\setminus k} \overline{J_\ell^{\text{loc}}}(\boldsymbol{w}_\ell) \tag{11}$$

Let $\boldsymbol{w}_k^o$ denote the minimizer of the local cost (10), given $\boldsymbol{w}_\ell$ for all $\ell \in \mathcal{N}_k\backslash\mathcal{C}(k)$. A completion-of-squares argument shows that, for any k, the cost $\overline{J_k^{\text{loc}}}(\boldsymbol{w}_k)$ can be expressed as

$$\overline{J_k^{\text{loc}}}(\boldsymbol{w}_k) = \overline{J_k^{\text{loc}}}(\boldsymbol{w}_k^o) + \|\boldsymbol{w}_k - \boldsymbol{w}_k^o\|^2_{\overline{\boldsymbol{R}}_k} \tag{12}$$

where

$$\overline{\boldsymbol{R}}_k = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{R}_{x,\ell} + \frac{\tau}{2}\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} (\rho_{k\ell}+\rho_{\ell k})\boldsymbol{I}_L. \tag{13}$$

Substituting (12) into the second term on the RHS of (11), and discarding the terms depending on $\boldsymbol{w}_k^o$ since they are independent of the optimization variables in the cluster, we can consider the following equivalent alternative to (11) at node k:

$$\overline{J_{\mathcal{C}(k)}}(\boldsymbol{w}_k) = \overline{J_k^{\text{loc}}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{C}(k)\setminus k} \|\boldsymbol{w}_\ell - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell} \tag{14}$$

where it holds that $\boldsymbol{w}_k = \boldsymbol{w}_\ell$ because $\ell \in \mathcal{C}(k)$. Therefore, the gradient of (14) with respect to $\boldsymbol{w}_k$ is equivalent to that of (7). However, the second term of (14) still requires multi-hop information passing. In order to avoid this situation, we relax (14) at node k by considering only information originating form its neighbors, i.e.,

$$\overline{J_{\mathcal{C}(k)}}'(\boldsymbol{w}_k) = \overline{J_k^{\text{loc}}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} \|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell}. \tag{15}$$

Usually, the weighting matrices $\overline{\boldsymbol{R}_\ell}$ are unavailable. Following an argument based on the Rayleigh-Ritz characterization of eigenvalues, a useful strategy is to replace each matrix $\overline{\boldsymbol{R}_\ell}$ by a weighted multiple of the identity matrix, say, as

$$\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell} \approx b_{\ell k}\,\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2 \tag{16}$$

The coefficients $b_{\ell k}$ will be incorporated into a left stochastic matrix to be defined and, therefore, the designer does not need to worry about the selection of these coefficients at this stage [9]. Based on the arguments presented so far, expression (15) can then be relaxed to the following form:

$$\begin{aligned}\overline{J_{\mathcal{C}(k)}}''(\boldsymbol{w}_k) &= \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\, E\big\{|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_k|^2\big\} \\ &+ \frac{\tau}{2}\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} (\rho_{k\ell}+\rho_{\ell k})\,\|\boldsymbol{w}_k - \boldsymbol{w}_\ell\|^2 + \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2.\end{aligned} \tag{17}$$

We now use (17) to derive distributed strategies.

3.2. Stochastic approximation algorithm

Let $\boldsymbol{w}_k(n)$ denote the estimate for $\boldsymbol{w}_k$ at iteration n. Using a constant step-size μ for each node, the update relation with an instantaneous approximation for the gradient vector, takes the following form:

$$\begin{aligned}\boldsymbol{w}_k(n+1) =& \boldsymbol{w}_k(n) - \mu \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}(\boldsymbol{x}_\ell(n)^\top \boldsymbol{w}_k(n) - d_\ell(n)) \\ & - \mu\,\tau \sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} \frac{\rho_{k\ell}+\rho_{\ell k}}{2}\,(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell(n)) \\ & - \mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\,(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell^o)\end{aligned} \tag{18}$$

Among other possible forms, expression (18) can be evaluated in two successive update steps:

$$\begin{aligned}\boldsymbol{\psi}_k(n+1) = \boldsymbol{w}_k(n) - \mu\Big[&\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}(\boldsymbol{x}_\ell(n)^\top \boldsymbol{w}_k(n) - d_\ell(n)) \\ &+ \tau \sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} \frac{\rho_{k\ell}+\rho_{\ell k}}{2}\,(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell(n))\Big]\end{aligned} \tag{19}$$

$$\boldsymbol{w}_k(n+1) = \boldsymbol{\psi}_k(n+1) + \mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\,(\boldsymbol{w}_\ell^o - \boldsymbol{w}_k(n)) \tag{20}$$

Following the same line of reasoning from [9] in the single-task case, we use $\boldsymbol{\psi}_\ell(n+1)$ as a local estimate for $\boldsymbol{w}_\ell^o$ in (20), and replace $\boldsymbol{w}_k(n)$ by $\boldsymbol{\psi}_k(n+1)$. Step (20) then becomes

$$\boldsymbol{w}_k(n+1) = \Big(1-\mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\Big)\boldsymbol{\psi}_k(n+1)+\mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\,\boldsymbol{\psi}_\ell(n+1). \tag{21}$$

The coefficients in the above relation can be redefined as:

$$\begin{aligned} a_{kk} &\triangleq 1 - \mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k} \\ a_{\ell k} &\triangleq \mu\, b_{\ell k}, \quad \ell \in \mathcal{N}_k^- \cap \mathcal{C}(k) \\ a_{\ell k} &\triangleq 0, \quad \ell \notin \mathcal{N}_k \cap \mathcal{C}(k)\end{aligned} \tag{22}$$

Let $\boldsymbol{A}$ be a left-stochastic matrix with (ℓ, k)-th entry $a_{\ell k}$. With this notation, we arrive at the following adapt-then-combine (ATC) diffusion strategy for solving problem (6):

$$\begin{cases}\boldsymbol{\psi}_k(n+1) = \boldsymbol{w}_k(n) + \mu \displaystyle\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}[d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{w}_k(n)]\boldsymbol{x}_\ell(n) \\ \qquad\qquad + \tau \displaystyle\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} \frac{\rho_{k\ell}+\rho_{\ell k}}{2}\,(\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)) \\ \boldsymbol{w}_k(n+1) = \displaystyle\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} a_{\ell k}\,\boldsymbol{\psi}_k(n+1)\end{cases} \tag{23}$$

4. NETWORK PERFORMANCE ANALYSIS

In this section we examine the convergence properties and network performance of the adaptive diffusion strategy (23). Let us denote by $\boldsymbol{w}(n)$ and $\boldsymbol{w}^\star$ the block weight estimate vector and the block optimum weight vector, respectively, both of size $L \times 1$, i.e.,

$$\boldsymbol{w}(n) = (\boldsymbol{w}_1^\top(n), \dots, \boldsymbol{w}_N^\top(n))^\top \tag{24}$$

$$\boldsymbol{w}^\star = (\boldsymbol{w}_1^{\star\top}, \dots, \boldsymbol{w}_N^{\star\top})^\top \tag{25}$$

with $\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}(k)}^\star$. Define the weight error vector by

$$\boldsymbol{v}(n) = \boldsymbol{w}(n) - \boldsymbol{w}^\star \tag{26}$$

Introduce the block diagonal matrix $\boldsymbol{H} = \text{diag}\,\{\boldsymbol{R}_1, \dots, \boldsymbol{R}_N\}$ with

$$\boldsymbol{R}_k = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{R}_{x,\ell}, \tag{27}$$

and let $\boldsymbol{P}$ be the matrix with (k, ℓ)-th entry $\rho_{k\ell}$. Introduce also the block matrix

$$\boldsymbol{Q} = \frac{1}{2}\left[\text{diag}\{(\boldsymbol{P}+\boldsymbol{P}^\top)\boldsymbol{1}\} - (\boldsymbol{P}+\boldsymbol{P}^\top)\right] \otimes \boldsymbol{I}_L. \tag{28}$$

and

$$\boldsymbol{B} = (\boldsymbol{A}\otimes\boldsymbol{I}_L)^\top\,[\boldsymbol{I}_{LN} - \mu(\boldsymbol{H} + \tau\boldsymbol{Q})] \tag{29}$$

$$\boldsymbol{r} = (\boldsymbol{A}\otimes\boldsymbol{I}_L)^\top \boldsymbol{Q}\,\boldsymbol{w}^\star \tag{30}$$

$$\boldsymbol{G} = (\boldsymbol{A}\otimes\boldsymbol{I}_L)^\top \boldsymbol{C}_I^\top \text{diag}\{\sigma_{z,1}^2\boldsymbol{R}_{x,1}, \dots, \sigma_{z,N}^2\boldsymbol{R}_{x,N}\}\boldsymbol{C}_I(\boldsymbol{A}\otimes\boldsymbol{I}_L) \tag{31}$$

with $\boldsymbol{C}_I = \boldsymbol{C}\otimes\boldsymbol{I}$. Assume that the step-size μ is sufficiently small such that higher-order powers of μ can be neglected and let

$$\boldsymbol{K} = \boldsymbol{B}^\top \otimes \boldsymbol{B}^\top. \tag{32}$$

With these matrices and vectors, we have the following results (proofs are omitted due to space constraints).

Theorem 1 *(Stability in the mean) Assume data model* (1) *and that the regression data* $\boldsymbol{x}_k(n)$ *is temporally white and independent over space. Then, for any initial condition, the diffusion multitask strategy* (23) *asymptotically converges in the mean if the step-size is chosen to satisfy*

$$0 < \mu < \frac{2}{\max_k\{\lambda_{\max}(\boldsymbol{R}_k)\} + 2\tau\max_k\{\boldsymbol{Q}_{kk}\}} \tag{33}$$

where $\lambda_{\mathit{max}}(\cdot)$ *denotes the maximum eigenvalue of the matrix arguement. In addition, we have*

$$\lim_{n\to\infty} E\{\boldsymbol{v}(n)\} = \mu\tau(\boldsymbol{B} - \boldsymbol{I}_{LN})^{-1}\boldsymbol{r}. \tag{34}$$

Theorem 2 *(Mean-square stability) Assume conditions in Theorem 1 hold. Then, the diffusion multitask strategy* (23) *is mean-square stable if the matrix* $\boldsymbol{K}$ *is stable, which is guaranteed by sufficiently small step-sizes that also satisfy* (33).

Theorem 3 *(Transient MSD) Considering a sufficiently small step-size* μ *that ensures mean and mean-square stability, the network MSD learning curve, defined by* $\zeta(n) = \frac{1}{N}E\{\|\boldsymbol{v}(n)\|\}^2$, *evolves according to the following recursions for* $n \geq 0$:

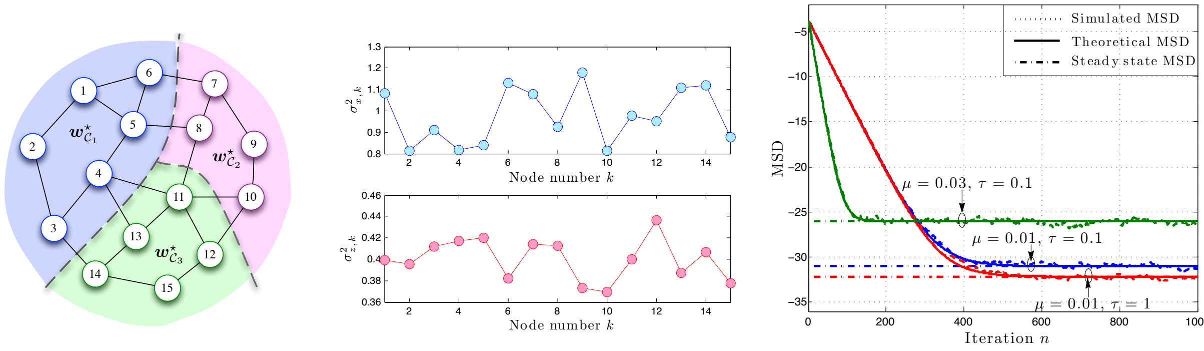
$$\begin{aligned}\zeta(n+1) = \zeta(n) + \frac{1}{N}\Big(&\mu^2\,\text{vec}(\boldsymbol{G}^\top)^\top\,\boldsymbol{K}^n\text{vec}(\boldsymbol{I}_{LN}) \\ &- E\{\|\boldsymbol{v}(0)\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n\text{vec}(\boldsymbol{I}_{LN})}\} + \mu^2\tau^2\|\boldsymbol{r}\|^2_{\boldsymbol{K}^n\text{vec}(\boldsymbol{I}_{LN})} \\ &- 2\mu\tau\,(\boldsymbol{\Gamma}(n) + \left[(\boldsymbol{B}\,E\{\boldsymbol{v}(n)\})^\top\otimes\boldsymbol{r}^\top\right]\text{vec}(\boldsymbol{I}_{LN})\Big) \\ \boldsymbol{\Gamma}(n+1) = \boldsymbol{\Gamma}(n)\,\boldsymbol{K} + &\left[(\boldsymbol{B}\,E\{\boldsymbol{v}(n)\})^\top\otimes\boldsymbol{r}^\top\right](\boldsymbol{K} - \boldsymbol{I}_{(LN)^2})\end{aligned} \tag{35}$$

with initial condition $\zeta(0) = \frac{1}{N}\|\boldsymbol{v}(0)\|^2$ *and* $\boldsymbol{\Gamma}(0) = \boldsymbol{0}_{(LN)^2}$.

Theorem 4 *(Steady-state MSD) If convergence is achieved, then the steady-state MSD for the diffusion network* (23) *is given by*

$$\zeta^*=\Big[\mu^2\text{vec}(\boldsymbol{G}^\top)^\top-2\mu\tau((\boldsymbol{B}E\{\boldsymbol{v}(\infty)\})^\top\otimes\boldsymbol{r}^\top)\Big]\,\sigma^o+\mu^2\tau^2\|\boldsymbol{r}\|^2_{\boldsymbol{\sigma}^o} \tag{36}$$

where $\sigma^o = \frac{1}{N}(\boldsymbol{I}_{(LN)^2} - \boldsymbol{K})^{-1}\text{vec}(\boldsymbol{I}_{LN})$ *and* $E\{\boldsymbol{v}(\infty)\}$ *is determined by expression* (34).

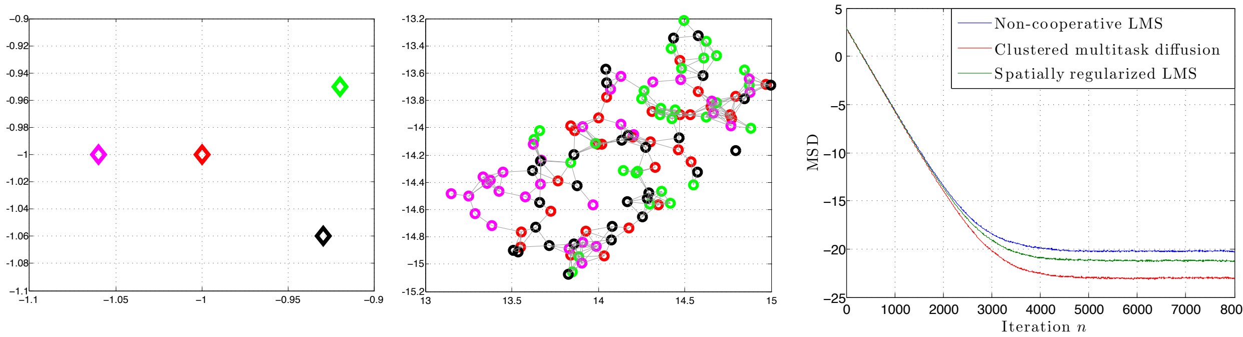


(a) Network topology.

(b) Input and noise variances.

(c) Convergence illustration.

Fig. 1. Network configuration and result illustration for Sec. 5.1.



(a) Target locations.

(b) Network topology.

(c) MSD evolution.

Fig. 2. Network configuration and result illustration for Sec. 5.2.

5. SIMULATIONS

5.1. Model validation

In this subsection we provide an illustrative example to show how the algorithm converges, and to illustrate theoretical models. We consider a network consisting of 15 nodes with connection and cluster structures shown in Fig. 1(a). The parameter vectors to be estimated in each cluster are $\boldsymbol{w}^\star_{\mathcal{C}_1} = (0.5238, -0.4008)^\top$, $\boldsymbol{w}^\star_{\mathcal{C}_2} = (0.5065, -0.3965)^\top$ and $\boldsymbol{w}^\star_{\mathcal{C}_3} = (0.4963, -0.3855)^\top$ respectively. Inputs $\boldsymbol{x}(n)$ were zero-mean 2×1 random vectors governed by a Gaussian distribution with covariance matrix $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_L$. The noises $z_k(n)$ were i.i.d. zero-mean Gaussian random variables, independent of any other signal with variances $\sigma^2_{z,k}$. Variances $\sigma^2_{x,k}$ and $\sigma^2_{z,k}$ used in this experiment are depicted in Fig. 1(b). Denoting the set cardinality by $|\cdot|$, regularization weights $\rho_{k\ell}$ were uniformly chosen as $\rho_{k\ell} = |\mathcal{N}_k \backslash \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \backslash \mathcal{C}(k)$. We considered the diffusion algorithm with measurement diffusion governed by an identity matrix $\boldsymbol{C} = \boldsymbol{I}_N$, and a uniform combination matrix $\boldsymbol{A}$ such that $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$. The algorithm was run with different step sizes and regularization parameters (μ, τ) such as $(0.01, 0.1)$, $(0.03, 0.1)$ and $(0.01, 1)$. Simulation results were obtained by averaging 100 Monte-Carlo runs. Transient MSD curves were obtained by (35). Steady-state MSD values were obtained by expression (36). Fig. 1(c) shows the evolutions of MSD and confirms theoretical analysis.

5.2. Multi-target localization

In this subsection we address an application of the problem of multi-target localization. Existing localization methods based on the diffusion strategy assume point targets [9]. However, in some situations, several distinct targets should be located. In this simulation, the objective is to estimate coordinates of three nearby targets as shown in Fig. 2(a) by a network composed by 120 nodes, with approximately 20 distance units away from targets. Each node randomly selected a target $i \in \{1, 2, 3, 4\}$. Nodes that selected the same target belong to the same cluster. The network connectivity and cluster structures are illustrated in Fig. 2(b). Noise standard deviations were set to $\sigma_{\alpha,k} = 0.1$, $\sigma_{\beta,k} = 0.01$ and $\sigma_{v,k} = 0.3$ (refer to [9] for the interpretation of these parameters). The proposed algorithm was run on each node with $\boldsymbol{C} = \boldsymbol{I}_N$, $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$, and $\rho_{k\ell} = |\mathcal{N}_k \backslash \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \backslash \mathcal{C}(k)$. The step size was set to $\mu = 0.1$. The regularization strength was set to $\tau = 0.01$. If each node is considered as a cluster, then algorithm (23) becomes a spatially regularized LMS, which was tested with the same parameter setting as the proposed algorithm. Non-cooperative LMS was also tested. MSD evolution curves were obtained by averaging over 100 Monte Carlo runs, as shown in Fig. 2(c). The benefit of cooperating and clustering is evidently illustrated.

6. CONCLUSION AND PERSPECTIVES

In this paper we derived a diffusion adaptation strategy for regularized learning over clustered multitask networks, and provided some convergence properties of the algorithm. However it can be seen that due to the summation over all nodes by (6), the problem inevitably leads to a symmetric regularization between pairs of nodes despite the fact that $\rho_{k\ell} \neq \rho_{\ell k}$. In order to benefit from additional flexibility, we will study the asymmetric regularized learning multitask problem in future work.

7. REFERENCES

[1] 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.

[2] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[3] 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.

[4] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optimiz.*, vol. 7, no. 4, pp. 913–926, Nov. 1997.

[5] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. of Sel. Topics Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[6] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with constant step size," *SIAM J. Optimiz.*, vol. 18, no. 1, pp. 29–51, Feb. 2007.

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

[8] A. H. Sayed, S.-Y Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Sig. Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[9] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., pp. 322–454. Elsevier, 2013. Also available as arXiv:1205.4220 [cs.MA], May 2012.

[10] 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.

[11] 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.

[12] 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.

[13] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

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

[15] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. CIP*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[16] S.-Y. Tu and A. H. Sayed, "Adaptive decision making over complex networks," in *Proc. ASILOMAR*, Pacific Grove, CA. USA, Nov. 2012, pp. 525–530.

[17] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013, pp. 5425–5429.

[18] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for leaning shared structures from muliple tasks," in *Proc. ICML*, Montreal, Canada, Jun. 2009, pp. 137–144.

[19] O. Chapelle, P. Shivaswmy, K. Q. Vadrevu, S. Weinberger, Y. Zhang, and B. Tseng, "Multi-task learning for boosting with application to web search ranking," in *Proc. SIGKDD*, Washington DC, USA, Jul. 2010, pp. 1189–1198.

[20] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *Proc. SIGKDD*, San Diego, CA, USA, Aug. 2011, pp. 814–822.

[21] J. Chen and C. Richard, "Performance analysis of diffusion LMS in multitask networks," in *Proc. IEEE CAMSAP*, Saint Martin, France, Dec. 2013, pp. 1–4.