DISTRIBUTED PRIMAL STRATEGIES OUTPERFORM PRIMAL-DUAL STRATEGIES OVER ADAPTIVE NETWORKS

Zaid J. Towfic and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

This work studies distributed primal-dual strategies for adaptation and learning over networks from streaming data. Two first-order methods are considered based on the Arrow-Hurwicz (AH) and augmented Lagrangian (AL) techniques. Several results are revealed in relation to the performance and stability of these strategies when employed over adaptive networks. It is found that these methods have worse steady-state mean-square-error performance than primal methods of the consensus and diffusion type. It is also found that the AH technique can become unstable under a partial observation model, while the other techniques are able to recover the unknown under this scenario. It is further shown that AL techniques are stable over a narrower range of step-sizes than primal strategies.

Index Terms— Augmented Lagrangian, Arrow-Hurwicz algorithm, primal strategies, diffusion strategies, consensus strategies

1. INTRODUCTION AND RELATED WORK

Distributed estimation is the task of estimating and tracking slowly drifting parameters by a network of agents, based solely on local interactions. In this work, we focus on distributed strategies that enable *continuous* adaptation and learning from streaming data by relying on stochastic gradient updates that employ *constant* step-sizes. The resulting networks become adaptive in nature, which means that the effect of gradient noise never dies out and seeps into the operation of the algorithms. For this reason, the design of such networks requires careful analysis in order to assess performance and provide convergence guarantees.

Many efficient algorithms have been proposed in the literature for inference over networks [1–13] such as consensus strategies [9–12] and diffusion strategies [2–8]. These strategies belong to the class of *primal* optimization techniques since they rely on estimating and propagating the primal variable. Previous studies have shown that sufficiently small step-sizes enable these strategies to learn well and in a stable manner. Explicit conditions on the step-size parameters for mean-square-error stability, as well as closed-form expressions for their steady-state mean-square-error performance already exist (see, e.g., [4, 14] and the many references therein). Besides primal methods, in the broad optimization literature, there is a second formidable class of techniques known as *primal-dual* methods such as the Arrow-Hurwicz (AH) method [15, 16] and the augmented Lagrangian (AL) method [16, 17]. These methods rely on propagating two sets of variables: the primal variable and a dual variable. The main advantage relative to primal methods is their ability to avoid ill-conditioning when solving constrained problems.

In contrast to existing useful studies on primal-dual algorithms (e.g., [18, 19]), we shall examine this class of strategies in the context of *stochastic* optimization over *adaptive* networks, where the optimization problem is *not* necessarily static anymore (i.e., its minimizer can drift with time) and where the exact form of the cost function need not be known beforehand because the statistical distribution of the data is generally unavailable. We therefore develop adaptive primal-dual distributed variants that can learn continuously from streaming data. This step is challenging because the dual function cannot be determined explicitly any longer, and, consequently, the computation of the optimal primal and dual variables cannot assume knowledge of the dual function. We address this difficulty by employing *constant* step-size adaptation and *instantaneous* data measurements to approximate the search directions.

We subsequently examine the behavior of AH and AL strategies under a partial observation model. This model refers to the important situation in which some agents may not be able to estimate the unknown parameter on their own, whereas the aggregate information from across the entire network is sufficient for the recovery of the unknown vector through local cooperation. We discover that the AH strategy can fail under this condition. More specifically, the AH network can become unstable *even* when the network has sufficient information to enable recovery of the unknown. In comparison, we show that the AL, consensus, and diffusion strategies are able to recover the unknown under the partial observation model.

We also examine the steady-state mean-square-deviation (MSD) of the primal-dual adaptive strategies and reveal that the Arrow-Hurwicz method achieves the same MSD performance as non-cooperative processing. This is a disappointing property for AH since the algorithm employs cooperation, and yet the agents are not able to achieve better performance. On the other hand, the augmented Lagrangian algorithm improves on the performance of non-cooperative processing, and can be made to approach the performance of diffusion and consensus strategies but only for large values of the regularization parameter. This means that the AL algorithm must utilize small step-sizes to approach the same performance level that other distributed (consensus and diffusion) algorithms can achieve with more reasonable parameter values. In addition, we show that the stability range for the AL algorithm is inversely proportional to the regularization parameter. This implies that in order for the AL algorithm to achieve the same MSD performance as the consensus and diffusion strategies, it is necessary for the AL strategy to utilize a smaller step-size to guarantee convergence, which in turn slows down its learning process.

This work was supported in part by NSF grant CCF-1011918 and ECCS-1407712. Emails: {ztowfic, sayed}@ucla.edu

2. ADAPTIVE PRIMAL STRATEGIES

In this section, we describe the problem formulation and review the two main primal techniques: diffusion and consensus for later user. Thus, consider a connected network of N agents that wish to estimate a real $M\times 1$ parameter vector w^o in a distributed manner. Each agent $k = 1, 2, \ldots, N$ has access to real scalar observations $\boldsymbol{d}_k(i)$ and zero-mean real $1 \times M$ regression vectors $\boldsymbol{u}_{k,i}$ that are assumed to be related via the model:

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

where $\boldsymbol{v}_k(i)$ is zero-mean real scalar random noise, and i is the time index. Models of the form (1) arise in many useful contexts in applications involving channel estimation, target tracking, equalization, beamforming, and localization [20, 21]. We denote the second-order moments by

$$R_{u,k} = \mathbb{E}\boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{u}_{k,i}, \quad r_{du,k} = \mathbb{E}\boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{d}_k(i), \quad \sigma_{v,k}^2 = \mathbb{E}\boldsymbol{v}_k^2(i) \tag{2}$$

and assume that the regression and noise processes are each temporally and spatially white. We also assume that $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_\ell(j)$ are independent of each other for all k, ℓ and i, j. We allow for the possibility that some individual covariance matrices, $R_{u,k}$, are singular but assume that the sum of all covariance matrices across the agents is positive-definite:

$$\sum_{k=1}^{N} R_{u,k} > 0 \tag{3}$$

This situation corresponds to the *partial observation* scenario where some of the agents may not able to solve the estimation problem on their own, and must instead cooperate with other nodes in order to estimate w^o.

To determine w^o, we consider an optimization problem involving an aggregate mean-square-error cost function:

$$\min_{w} \frac{1}{2}\sum_{k=1}^{N}\mathbb{E}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w)^2 \tag{4}$$

It is straightforward to verify that w^o from (1) is the unique minimizer of (4). We will compare the performance of the primal-dual algorithms introduced in the next section to baseline primal algorithms to solve (4) in a distributed manner such as diffusion strategies [2, 3, 14, 21]:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu\boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) & \text{(5a)} \\ \boldsymbol{w}_{k,i} = \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} & \text{(5b)} \end{cases}$$

where $\mu > 0$ is a small step-size parameter and $\mathcal{N}_k$ denotes the neighborhood of agent k. Moreover, the coefficients $\{a_{\ell k}\}$ that comprise the matrix A are non-negative convex combination coefficients that satisfy the conditions:

$$a_{\ell k} \geq 0, \qquad \sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{6}$$

In other words, the matrix A is left-stochastic and satisfies $A^T\mathbb{1}_N = \mathbb{1}_N$. In (5a)–(5b), each agent k first updates its estimate $\boldsymbol{w}_{k,i-1}$ to an intermediate value by using its sensed data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ through (5a), and subsequently aggregates the information from the neighbors through (5b). A connected network is said to be strongly-connected when at least one a_{kk} is strictly positive; i.e., there exists at least one agent with a self-loop, which is reasonable since it means that at least one agent in the network should have some trust in its own data. We will assume that the network is strongly connected for the remainder of the article.

We will also compare the performance of the primal-dual algorithms to that of consensus-type strategies (albeit with constant step-sizes) [9–12]:

$$\begin{cases} \boldsymbol{\phi}_{k,i-1} = \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1} & \text{(7a)} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu\boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) & \text{(7b)} \end{cases}$$

It is important to note the asymmetry in the update (7b) with both $\{\boldsymbol{\phi}_{k,i-1}, \boldsymbol{w}_{k,i-1}\}$ appearing on the right-hand side of (7b), while the *same* state variable $\boldsymbol{w}_{k,i-1}$ appears on the right-hand side of the diffusion strategy (5a). This asymmetry has been shown to be a source of instability for consensus-based solutions [4, 14, 22].

When the step-size parameter is sufficiently small, the steady-state deviation (MSD) of the consensus and diffusion strategies can be shown to match to first-order in μ [14]. For example, when A is doubly-stochastic, it holds that:

$$\text{MSD} \triangleq \lim_{i\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 \tag{8a}$$

$$= \frac{\mu}{2N}\text{Tr}\left(\left(\sum_{k=1}^{N}R_{u,k}\right)^{-1}\left(\sum_{k=1}^{N}\sigma_{v,k}^2 R_{u,k}\right)\right) + O(\mu^2) \tag{8b}$$

A related expression also exists for the case when A is left-stochastic and it will further involve the entries of the Perron vector of A [4,14]. Furthermore, it can be shown that the diffusion strategy (5a)–(5b) is guaranteed to converge in the mean for any connected network topology, i.e., $\mathbb{E}\boldsymbol{w}_{k,i} \to w^o$, as long as the agents are individually mean stable. This condition is satisfied for step-sizes satisfying:

$$0 < \mu < \min_{1\leq k\leq N}\left\{\frac{2}{\lambda_{\max}(R_{u,k})}\right\} \tag{9}$$

In contrast, consensus implementations can become unstable for some topologies even if all individual agents are mean stable [14,22].

3. ADAPTIVE PRIMAL-DUAL STRATEGIES

To motivate the adaptive primal dual strategy, we start by replacing (4) by the following equivalent constrained optimization problem where the variable w is replaced by w_k:

$$\min_{w} \quad \frac{1}{2}\sum_{k=1}^{N}\mathbb{E}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w_k)^2 \tag{10a}$$

$$\text{s.t.} \quad w_1 = w_2 = \cdots = w_N \tag{10b}$$

The following definition is useful [23].

Definition 1 (Incidence matrix of an undirected graph)**.** *Given a graph G, the incidence matrix $C = [c_{ek}]$ is an $E \times N$ matrix, where E is the total number of edges in the graph and N is the total number of nodes, with entries defined as follows:*

$$c_{ek} = \begin{cases} +1, & k \text{ is the lower indexed node connected to } e \\ -1, & k \text{ is the higher indexed node connected to } e \\ 0, & \text{otherwise} \end{cases}$$

Thus, $C\mathbb{1}_N = \mathbb{0}_E$. Self-loops are excluded. ■

Since the network is connected, we may rewrite (10a)-(10b) as

$$\min_{w} \quad \frac{1}{2}\sum_{k=1}^{N}\mathbb{E}(\boldsymbol{d}_k(i)-\boldsymbol{u}_{k,i}w_k)^2 \tag{11a}$$

$$\text{s.t.} \quad \mathcal{C}\mathcal{W} = \mathbb{0}_{EM} \tag{11b}$$

where we introduced the extended quantities:

$$\mathcal{C} \triangleq C \otimes I_M, \quad \mathcal{W} \triangleq \text{col}\{w_1,\dots,w_N\} \tag{12}$$

The augmented Lagrangian of the constrained problem (11a)–(11b) is given by [16, 24]:

$$f(\mathcal{W},\lambda) = \frac{1}{2}\sum_{k=1}^{N}\mathbb{E}(\boldsymbol{d}_k(i)-\boldsymbol{u}_{k,i}w_k)^2+\lambda^{\mathsf{T}}\mathcal{C}\mathcal{W}+\frac{\eta}{2}\|\mathcal{C}\mathcal{W}\|^2 \tag{13}$$

where $\lambda \in \mathbb{R}^{EM\times 1}$ is the Lagrange multiplier vector: it consists of E subvectors, $\lambda = \text{col}\{\lambda_e\}$, each of size $M \times 1$ for $e = 1, 2, \dots, E$. One subvector λ_e is associated with each edge e. Notice that the matrix $C^{\mathsf{T}}C$ is in fact the Laplacian matrix of the network [23]. It is now possible to seek a saddle-point of (13) by employing a *stochastic approximation* version of the first-order augmented Lagrangian algorithm [16, pp. 240–242] [17, p. 456]. The implementation relies on a *stochastic* gradient descent step with respect to the primal variable, $\mathcal{W}$, and a gradient ascent step with respect to the dual variable, λ, as follows:

$$\begin{cases} \boldsymbol{\mathcal{W}}_i = \boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\nabla_{\mathcal{W}}} f(\boldsymbol{\mathcal{W}}_{i-1},\boldsymbol{\lambda}_{i-1}) & \text{(14a)} \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \mu\nabla_{\lambda} f(\boldsymbol{\mathcal{W}}_{i-1},\boldsymbol{\lambda}_{i-1}) & \text{(14b)} \end{cases}$$

Observe that we are using an approximate gradient vector in (14a) and the exact gradient vector in (14b); this is because differentiation relative to $\mathcal{W}$ requires knowledge of the data statistics, which are not available. These gradient vectors are evaluated as follows:

$$\widehat{\nabla_{\mathcal{W}}} f(\boldsymbol{\mathcal{W}},\boldsymbol{\lambda}) = \boldsymbol{h}_i + \mathcal{C}^{\mathsf{T}}\boldsymbol{\lambda} + \eta\mathcal{C}^{\mathsf{T}}\mathcal{C}\boldsymbol{\mathcal{W}} \tag{15a}$$

$$\nabla_{\lambda} f(\boldsymbol{\mathcal{W}},\boldsymbol{\lambda}) = \mathcal{C}\boldsymbol{\mathcal{W}} \tag{15b}$$

where the vector $\boldsymbol{h}_i$ amounts to an instantaneous approximation for the gradient vector of the first term on the right-hand side of (13); its $k-$th entry is given by $-\boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_k)$. Observe that (15a)–(15b) can also be written in the following form:

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \boldsymbol{w}_{k,i-1} - \mu\sum_{e=1}^{E} c_{ek}\boldsymbol{\lambda}_{e,i-1} - \mu\eta\sum_{\ell\in\mathcal{N}_k} l_{k\ell}\boldsymbol{w}_{\ell,i-1} & \text{(16a)} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \mu\boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) & \text{(16b)} \\ \boldsymbol{\lambda}_{e,i} = \boldsymbol{\lambda}_{e,i-1} + \mu(\boldsymbol{w}_{k,i-1} - \boldsymbol{w}_{\ell,i-1}) \quad [\ell > k, \ell \in \mathcal{N}_k] & \text{(16c)} \end{cases}$$

where either node connected to edge e may update $\boldsymbol{\lambda}_{e,i-1}$.

When $\eta = 0$ in (13), (14a), and (16a), we obtain the *distributed Arrow-Hurwicz* (AH) method, also considered in [25, 26] for the solution of saddle point problems for other cost functions. Reference [25] considers problems that arise in the context of reinforcement learning in response to target policies, while reference [26] considers regret analysis problems and employs *decaying* step-sizes rather than continuous adaptation. Moreover, in contrast to [12], the derived AL and AH implementations do not require the availability of special *bridge* nodes alongside the regular nodes.

4. MAIN RESULTS

In this section, we summarize the main results. Proofs are omitted due to space limitations — see [27].

4.1. Error Dynamics

We know that the optimizer of (11a)–(11b) is $w_k = w^o$ for all $k = 1, \dots, N$, where w^o was defined in (1) since (11a)–(11b) is equivalent to (4). We introduce the error vector at each agent k, $\widetilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$, and collect all errors from across the network into the block column vector:

$$\widetilde{\boldsymbol{\mathcal{W}}}_i = \text{col}\{\widetilde{\boldsymbol{w}}_{1,i}, \widetilde{\boldsymbol{w}}_{2,i}, \dots, \widetilde{\boldsymbol{w}}_{N,i}\} \tag{17}$$

We also introduce the singular-value-decomposition:

$$C = USV^{\mathsf{T}} \tag{18}$$

where $U \in \mathbb{R}^{E\times E}$ and $V \in \mathbb{R}^{N\times N}$ are orthogonal matrices and $S \in \mathbb{R}^{E\times N}$ is partitioned according to

$$S = \left[\begin{array}{c|c} S_2 & \mathbb{0}_{N-1} \\ \hline 0_{(E-N+1)\times(N-1)} & \mathbb{0}_{E-N+1} \end{array}\right] \tag{19}$$

where the square diagonal matrix $S_2 \in \mathbb{R}^{(N-1)\times(N-1)}$ contains the nonzero singular values of C along its main diagonal and is therefore non-singular. We also partition V into:

$$V = \left[\begin{array}{cc} V_2 & \frac{1}{\sqrt{N}}\mathbb{1}_N \end{array}\right] \tag{20}$$

It is then possible to establish the following result:

Lemma 1 (Error dynamics of primal-dual strategies)**.** *Let the network be connected. Then, the error dynamics of the primal-dual AH and AL algorithms* (16a)–(16c) *evolve over time as follows:*

$$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}'_{1,i} \\ \widetilde{\boldsymbol{\mathcal{W}}}'_{2,i} \\ \widetilde{\boldsymbol{\lambda}}'_{1,i} \end{bmatrix} = \boldsymbol{\mathcal{B}}'_i \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}'_{1,i-1} \\ \widetilde{\boldsymbol{\mathcal{W}}}'_{2,i-1} \\ \widetilde{\boldsymbol{\lambda}}'_{1,i-1} \end{bmatrix} - \mu \begin{bmatrix} \mathcal{V}_2^{\mathsf{T}}\boldsymbol{z}_i \\ \mathcal{V}_0^{\mathsf{T}}\boldsymbol{z}_i \\ \mathbb{0}_{(N-1)M} \end{bmatrix} \tag{21}$$

where

$$\boldsymbol{\mathcal{B}}'_i \triangleq I_{(2N-1)M} - \mu\boldsymbol{\mathcal{R}}'_i \tag{22}$$

$$\boldsymbol{\mathcal{R}}'_i \triangleq \left[\begin{array}{cc|c} \mathcal{V}_2^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i\mathcal{V}_2 + \eta\mathcal{S}_2^{\mathsf{T}}\mathcal{S}_2 & \mathcal{V}_2^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i\mathcal{V}_0 & \mathcal{S}_2^{\mathsf{T}} \\ \mathcal{V}_0^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i\mathcal{V}_2 & \mathcal{V}_0^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i\mathcal{V}_0 & 0_{M\times(N-1)M} \\ \hline -\mathcal{S}_2 & 0_{(N-1)M\times M} & 0_{(N-1)M} \end{array}\right] \tag{23}$$

$$\widetilde{\boldsymbol{\lambda}}'_i \triangleq \mathcal{U}^{\mathsf{T}}\widetilde{\boldsymbol{\lambda}}_i = \begin{bmatrix} \widetilde{\boldsymbol{\lambda}}'_{1,i} \\ \widetilde{\boldsymbol{\lambda}}'_{2,i} \end{bmatrix}, \quad \widetilde{\boldsymbol{\mathcal{W}}}'_i \triangleq \mathcal{V}^{\mathsf{T}}\widetilde{\boldsymbol{\mathcal{W}}}_i = \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}'_{1,i} \\ \widetilde{\boldsymbol{\mathcal{W}}}'_{2,i} \end{bmatrix} \tag{24}$$

and $\mathcal{U} = U \otimes I_M$, $\mathcal{S}_2 = S_2 \otimes I_M$, $\mathcal{V} = V \otimes I_M$, $\mathcal{V}_2 = V_2 \otimes I_M$, $\mathcal{V}_0 = \frac{1}{\sqrt{N}}\mathbb{1}_N \otimes I_M$, *and*

$$\boldsymbol{z}_i \triangleq \text{col}\{\boldsymbol{u}_{1,i}\boldsymbol{v}_1(i), \dots \boldsymbol{u}_{N,i}\boldsymbol{v}_N(i)\} \tag{25}$$

$$\boldsymbol{\mathcal{H}}_i \triangleq \text{blockdiag}\{\boldsymbol{u}_{1,i}^{\mathsf{T}}\boldsymbol{u}_{1,i}, \dots, \boldsymbol{u}_{N,i}^{\mathsf{T}}\boldsymbol{u}_{N,i}\} \tag{26}$$

where we denote the first $(N-1)M$ *elements of* $\widetilde{\boldsymbol{\lambda}}'_i$ *by* $\widetilde{\boldsymbol{\lambda}}'_{1,i}$ *while the remaining elements are collected into the vector* $\widetilde{\boldsymbol{\lambda}}'_{2,i}$*. Similarly, the first* $(N-1)M$ *elements of the vector* $\widetilde{\boldsymbol{\mathcal{W}}}'_i$ *are denoted by* $\widetilde{\boldsymbol{\mathcal{W}}}'_{1,i}$ *while the remaining* M *elements are denoted by* $\widetilde{\boldsymbol{\mathcal{W}}}'_{2,i}$*.* ■

4.2. Stability Results

Using the error-recursion (21), we can establish the following statement regarding mean stability ($\mathbb{E}\widetilde{\boldsymbol{w}}_{k,i} \to 0$) and mean-square-error stability ($\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ tends to a small bounded region).

Theorem 1 (Stability of the AL algorithm)**.** *Under* (3) *and over connected networks, there exists* $\bar{\eta}$ *such that for all* $\eta > \bar{\eta}$, *the matrix* $\mathcal{H} + \eta \cdot \mathcal{C}^{\mathsf{T}}\mathcal{C}$ *is positive-definite and the AL algorithm is mean and mean-square stable for small* μ. ■

We conclude from Theorem 1 that the AL algorithm can be guaranteed to be stable for large enough η and small enough μ. Observe that Theorem 1 may not apply to the AH algorithm since for that algorithm stability must be guaranteed for $\eta = 0$. For such a case, we can obtain the following stability guarantee.

Theorem 2 (Stability of the AL and AH algorithms)**.** *Assume that* $R_{u,k} > 0$ *for all* k *and let the network be connected. Then, the AL and AH algorithms are mean and mean-square stable for small* μ. ■

Observe that Theorem 2 does not apply to the partial observation model (3) since it requires each covariance matrix to satisfy $R_{u,k} > 0$. In fact, there are cases where the AH is not stable under the partial observation model while the AL algorithm can be made to converge for large enough η [27]. Another aspect we need to consider is how the stability range of μ depends on the regularization parameter η and the network topology. It is already known that the mean stability range for diffusion strategies (9) is independent of the network topology [21]. We can also obtain the required step-size range for convergence of the AL algorithm for large values of η.

Theorem 3. *Let the network be connected and let* $R_{u,k} = R_u > 0$ *for all* k. *Then, for large* $\eta > 0$, *we have that the step-size range to guarantee mean stability of the AL algorithm is*

$$\mu < \frac{2}{\eta \cdot \rho(C^{\mathsf{T}}C)} \tag{27}$$

where C *is the incidence matrix of the network topology and* $\rho(C^{\mathsf{T}}C)$ *denotes the spectral radius of* $C^{\mathsf{T}}C$. ■

Clearly, as $\eta \to \infty$, the upper-bound on the step-size approaches zero. This means that the algorithm is sensitive to both the regularization parameter η and the topology (through $\rho(C^{\mathsf{T}}C)$).

4.3. Mean-Square-Error Performance

Let $\mathcal{H} \triangleq \text{blockdiag}\{R_{u,1}, \ldots, R_{u,N}\}$, then we have the following.

Theorem 4 (MSD performance of AH algorithm)**.** *Assuming each* $R_{u,k}$ *is positive-definite and the network is connected, the network* MSD *for the AH algorithm for small step-sizes is given by:*

$$\text{MSD} = \mu \frac{M}{2} \frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 + O(\mu^2) \tag{28}$$

■

Expression (28) is equal to the average performance across a collection of N *non-cooperative* agents (see, e.g., [14, 21]). In this way, Theorem 4 is a surprising result for the AH algorithm since even with cooperation, the network is unable to improve over non-cooperation. This result does not carry over to the AL algorithm.

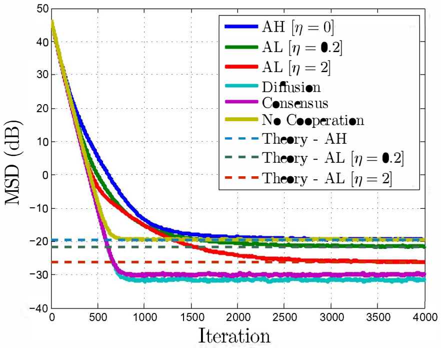

Fig. 1. Simulation result for diffusion, consensus, AH, and AL.

Theorem 5 (MSD performance of AL algorithm)**.** *Assume that the matrix* $\mathcal{H} + \eta \cdot \mathcal{C}^{\mathsf{T}}\mathcal{C}$ *is positive-definite (guaranteed by Theorem 1). Then, for sufficiently small step-sizes, the network* MSD *for the AL algorithm for large* η *is given by*

$$\text{MSD} = \frac{\mu}{2N} \text{Tr}\left(\left(\sum_{k=1}^{N} R_{u,k} \right)^{-1} \left(\sum_{k=1}^{N} \sigma_{v,k}^2 R_{u,k} \right) \right) + \frac{\mu}{2N\eta} \text{Tr}\left(\mathcal{R}_z (\mathcal{C}^{\mathsf{T}}\mathcal{C})^{\dagger} \right) + O\left(\frac{\mu}{N^2 \eta} \right) + O(\mu^2) \tag{29}$$

where $(\mathcal{C}^{\mathsf{T}}\mathcal{C})^{\dagger}$ *denotes the pseudoinverse of* $\mathcal{C}^{\mathsf{T}}\mathcal{C}$. ■

By examining (29), we learn that the performance of the AL algorithm for large η approaches the performance of the diffusion strategy given by (8b). However, recalling the fact that the step-size range required for convergence, under the large η regime and the assumption that $R_{u,1} = \ldots = R_{u,N} = R_u > 0$ in (27), is inversely proportional to η, we conclude that the AL algorithm can only approach the performance of the diffusion strategy as $\mu \to 0$ and $\eta \to \infty$. In addition, the performance of the AL algorithm depends explicitly on the network topology through the matrix $\mathcal{C}^{\mathsf{T}}\mathcal{C}$. Observe that this is not the case in (8b) for the primal strategies. Thus, even for large η, AL is sensitive to the network topology.

5. SIMULATION

Consider a network of $N = 20$ agents and $M = 5$. We generate a positive-definite matrix $R_u > 0$ with eigenvalues $1 + \boldsymbol{x}_m$, where $\boldsymbol{x}_m$ is a uniform random variable. We let $\mathcal{H} = I_N \otimes R_u$ with $\mu = 0.01$, which allows all algorithms to converge. The diffusion and consensus strategies utilize a doubly-stochastic matrix generated through the Metropolis rule [4]. We note that the diffusion and consensus algorithms can improve their MSD performance by designing the combination matrix based on the Hastings rule [28, 29], but we assume that the nodes are noise-variance agnostic. For (14a)–(14b), we simulate three values of η: 0, 0.2, and 2. This will allow us to validate our analysis results where an increase in η yields improvement in the MSD (Theorem 5). We observe in Fig. 1 that as η is increased, the performance of the AL algorithm improves, but is still worse than that of the consensus algorithm (7a)–(7b) and the diffusion strategy (5a)–(5b). Furthermore, the convergence rate of the AH algorithm is worse than that of non-cooperation, even though both algorithms achieve the same MSD performance. It is possible to further increase η in order to make the performance of the AL algorithm match better with that of the consensus and diffusion strategies. However, it is important to note that if η is increased too much, the algorithm will diverge (recall (27)).

6. REFERENCES

[1] K. I. Tsianos and M. G. Rabbat, "Distributed strongly convex optimization," in *Proc. Allerton Conf.*, Allerton, IL, Oct., 2012, pp. 593–600.

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

[3] 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.

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

[5] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[6] 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.

[7] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Sig. Proc. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[8] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

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

[10] 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.

[11] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Sel. Topics. Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[12] I. D Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[13] D. H. Dini and D. P. Mandic, "Cooperative adaptive estimation of distributed noncircular complex signals," in *Proc. Asilomar Conference*, Pacific Grove, CA, Nov., 2012, pp. 1518–1522.

[14] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[15] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Non-linear Programming*, Stanford University Press, CA, 1958.

[16] B. Polyak, *Introduction to Optimization*, Optimization Software, NY, 1987.

[17] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, MA, 1999.

[18] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," *in Academic Press Library in Signal Processing,* vol. 2, R. Chellapa and S. Theodoridis, *Eds.,* pp.329–408, Elsevier, 2014.

[19] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jul. 2011.

[20] A. H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.

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

[22] 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.

[23] R. B. Bapat, *Graphs and Matrices*, Springer, NY, 2010.

[24] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, NY, 2004.

[25] S. Valcarcel Macua, J. Chen, S. Zazo, and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," to appear in *IEEE Trans. Aut. Control*, also available as *arXiv:1312.7606v1*, Dec. 2013.

[26] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," in *Proc. IEEE ICASSP*, Florence, Italy, May, 2014, pp. 8292–8296.

[27] Z. J. Towfic and A. H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *submitted for publication,* available as *arXiv:1408.3693*, Aug. 2014.

[28] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[29] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, Dec. 2004.