PERFORMANCE OF DIFFUSION ADAPTATION FOR COLLABORATIVE OPTIMIZATION

Jianshu Chen and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

We derive an adaptive diffusion mechanism to optimize global cost functions in a distributed manner over a network of nodes. The cost function is assumed to consist of the sum of individual components, and diffusion adaptation is used to enable the nodes to cooperate locally through in-network processing in order to solve the desired optimization problem. We analyze the mean-square-error performance of the algorithm, including its transient and steady-state behavior. We illustrate one application in the context of least-mean-squares estimation for sparse vectors.

Index Terms— Distributed optimization, diffusion adaptation, in-network processing, learning, energy conservation.

1. INTRODUCTION

We consider the problem of optimizing a global cost function in a distributed manner. The cost function is assumed to consist of the sum of individual components, and spatially distributed nodes are then used to seek the common minimizer through local interactions. There are already a couple of useful techniques for the solution of such optimization problems. Most notable among them is the incremental approach [1–3]. In this approach, a cyclic path is defined over the nodes and data are processed in a cyclic manner through the network until optimization is achieved. However, determining a cyclic path that covers all nodes is an NP-hard problem and, in addition, cyclic trajectories are vulnerable to link and node failures. In earlier works [4–7], we introduced the concept of diffusion adaptation and showed how it can be used to solve least-mean-squares estimation problems in a decentralized manner very effectively. In the diffusion approach, information is processed locally at the nodes and then diffused through a real-time sharing mechanism. This paper generalizes the diffusive learning process of [4–7].

2. PROBLEM FORMULATION

The objective is to determine an optimal $M \times 1$ vector w^o that minimizes a global cost function of the form:

$$J^{\text{glob}}(w) = \sum_{l=1}^{N} J_l(w) \tag{1}$$

where $J_l(w)$, $l = 1, 2, \ldots, N$, are individual real-valued functions defined over $w \in \mathbb{R}^M$ and assumed to be differentiable and convex. We assume $J^{\text{glob}}(w)$ in (1) is strictly convex so that the minimizer w^o is unique.

Email: {jshchen, sayed}@ee.ucla.edu. This work was supported in part by NSF grants CCF-1011918 and CCF-0942936.

Using the approach of [7], we derived in [8, 9] the following class of diffusion algorithms for the distributed optimization of (1):

$$\begin{cases} \phi_{k,i-1} = \displaystyle\sum_{l=1}^{N} p_{1,l,k} w_{l,i-1} & \text{(2a)} \\ \psi_{k,i} = \phi_{k,i-1} - \mu_k \displaystyle\sum_{l=1}^{N} s_{l,k} \nabla_w J_l(\phi_{k,i-1}) & \text{(2b)} \\ w_{k,i} = \displaystyle\sum_{l=1}^{N} p_{2,l,k} \psi_{l,i} & \text{(2c)} \end{cases}$$

where $\boldsymbol{w}_{k,i}$ is the local estimate for w^o at node k and time i, μ_k is the step-size parameter at node k, and $\nabla_w J_l(w)$ is the (column) gradient vector of $J_l(\cdot)$ relative to w. Moreover, the non-negative coefficients $\{p_{1,l,k}\}$, $\{s_{l,k}\}$, and $\{p_{2,l,k}\}$ are the (l,k)-th entries of matrices P_1, S, and P_2, respectively, and are required to satisfy:

$$\boxed{\begin{array}{l} P_1^T \mathbb{1} = \mathbb{1},\ P_2^T \mathbb{1} = \mathbb{1},\ S\mathbb{1} = \mathbb{1} \\ p_{1,l,k} = 0,\ p_{2,l,k} = 0\ , s_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \end{array}} \tag{3}$$

where $\mathbb{1}$ denotes a vector with all entries equal to one, and $\mathcal{N}_k$ denotes the neighborhood of node k. Different choices for $\{P_1, P_2, S\}$ correspond to different cooperation strategies [7]. For example, $P_1 = I$, $P_2 = I$ and $S = I$ correspond to the no-cooperation case. On the other hand, $P_1 = I$, $P_2 = A$ and $S = C$ correspond to the adapt-then-combine (ATC) strategy [7], while the choice $P_1 = A$, $P_2 = I$ and $S = C$ correspond to the combine-then-adapt (CTA) strategy [6] (where A and C denote matrices with nonnegative entries that satisfy $A^T\mathbb{1} = \mathbb{1}$ and $C\mathbb{1} = \mathbb{1}$).

Adaptive diffusion strategies of these forms were originally derived in [4–7] and used to solve distributed minimum mean-square-error estimation problems over networks. The special case of the CTA strategy with $C = I$ appeared later in [10, 11] to solve distributed optimization problems albeit by further requiring all nodes to reach consensus. Diffusion strategies of the form (2) are more powerful in a couple of respects. First, they do not only diffuse the local estimates, but they *can also* diffuse the local gradient vectors. Second, the combination weights $\{p_{1,l,k}, p_{2,l,k}\}$ are not required to be doubly stochastic (we are only requiring the columns of P_1 and P_2 to add up to one). Finally, the step-size parameters in (2) are not required to be vanishing; instead, they can assume constant values, and this flexibility enables continuous adaptation and learning. Multi-agent systems in nature behave in this manner; they do not require exact agreement among their agents but allow for fluctuations due to individual levels of assessment and noise.

We established in [8] that the individual estimators $\{w_{k,i}\}$ that are generated by (2) converge asymptotically to w^o under the condition of bounded Hessian matrices, as specified by (15) further ahead.

In many situations in practice, the true gradient vectors needed in (2b) may not be available. Instead, perturbed versions are available, which we model as

$$\widehat{\nabla}_w J_l(\boldsymbol{w}) = \nabla_w J_l(\boldsymbol{w}) + \boldsymbol{v}_l(\boldsymbol{w}) \tag{4}$$

where the noise term, $\boldsymbol{v}_l(\boldsymbol{w})$, may depend on $\boldsymbol{w}$ and will be required to satisfy certain conditions given by (17)–(18). We refer to the perturbation in (4) as gradient noise. In this paper, we examine the effect of gradient noise on the convergence and mean-square performance of the diffusion strategy. In particular, we characterize the steady-state mean-square-deviation (MSD) of the network.

3. MEAN-SQUARED PERFORMANCE

3.1. Error Recursions

Introduce the error vectors:

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

Then, from (2)–(4), we have

$$\begin{cases} \tilde{\boldsymbol{\phi}}_{k,i-1} = \sum_{l=1}^{N} p_{1,l,k}\tilde{\boldsymbol{w}}_{l,i-1} & \text{(5a)} \\ \tilde{\boldsymbol{\psi}}_{k,i} = \tilde{\boldsymbol{\phi}}_{k,i-1} + \mu_k \sum_{l=1}^{N} s_{l,k}\big[\nabla_w J_l(\boldsymbol{\phi}_{k,i-1}) + \boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1})\big] & \text{(5b)} \\ \tilde{\boldsymbol{w}}_{k,i} = \sum_{l=1}^{N} p_{2,l,k}\tilde{\boldsymbol{\psi}}_{l,i} & \text{(5c)} \end{cases}$$

We need to relate $\nabla_w J_l(\boldsymbol{\phi}_{k,i-1})$ in (5b) to $\tilde{\boldsymbol{\phi}}_{k,i-1}$. Assume each $J_l(w)$ has a minimizer at the same w^o. Then, from [12, p.6]:

$$\begin{aligned} \nabla_w J_l(\boldsymbol{\phi}_{k,i-1}) &= -\left[\int_0^1 \nabla_w^2 J_l\big(w^o - t\tilde{\boldsymbol{\phi}}_{k,i-1}\big)dt\right]\tilde{\boldsymbol{\phi}}_{k,i-1} \\ &\triangleq -\boldsymbol{H}_{l,k,i-1}\tilde{\boldsymbol{\phi}}_{k,i-1} \end{aligned} \tag{6}$$

Substituting (6) into (5b) leads to:

$$\tilde{\boldsymbol{\psi}}_{k,i} = \Big[I_M - \mu_k \sum_{l=1}^{N} s_{l,k}\boldsymbol{H}_{l,k,i-1}\Big]\tilde{\boldsymbol{\phi}}_{k,i-1} + \mu_k \sum_{l=1}^{N} s_{l,k}\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1}) \tag{7}$$

Introduce the global error vectors and matrices:

$$\tilde{\boldsymbol{\phi}}_i = \mathrm{col}\{\tilde{\boldsymbol{\phi}}_{1,i} \cdots \tilde{\boldsymbol{\phi}}_{N,i}\}, \qquad \tilde{\boldsymbol{\psi}}_i = \mathrm{col}\{\tilde{\boldsymbol{\psi}}_{1,i} \cdots \tilde{\boldsymbol{\psi}}_{N,i}\} \tag{8}$$

$$\tilde{\boldsymbol{w}}_i = \mathrm{col}\{\tilde{\boldsymbol{w}}_{1,i} \cdots \tilde{\boldsymbol{w}}_{N,i}\} \tag{9}$$

$$\mathcal{P}_1 = P_1 \otimes I_M, \ \mathcal{P}_2 = P_2 \otimes I_M, \mathcal{S} = S \otimes I_M, \ \mathcal{M} = \Omega \otimes I_M \tag{10}$$

$$\Omega = \mathrm{diag}\{\mu_1, \ \ldots, \ \mu_N\} \tag{11}$$

$$\boldsymbol{\mathcal{D}}_{i-1} = \textstyle\sum_{l=1}^{N} \mathrm{diag}\big\{s_{l,1}\boldsymbol{H}_{l,1,i-1} \cdots, s_{l,N}\boldsymbol{H}_{l,N,i-1}\big\} \tag{12}$$

$$\boldsymbol{\mathcal{G}}_i = \textstyle\sum_{l=1}^{N} \mathrm{col}\big\{s_{l,1}\boldsymbol{v}_l(\boldsymbol{\phi}_{1,i-1}), \cdots, s_{l,N}\boldsymbol{v}_l(\boldsymbol{\phi}_{N,i-1})\big\} \tag{13}$$

Then, recursions (5a), (7), and (5c) give:

$$\boxed{\tilde{\boldsymbol{w}}_i = \mathcal{P}_2^T[I_{MN} - \mathcal{M}\boldsymbol{\mathcal{D}}_{i-1}]\mathcal{P}_1^T\tilde{\boldsymbol{w}}_{i-1} + \mathcal{P}_2^T\mathcal{M}\boldsymbol{\mathcal{G}}_i} \tag{14}$$

Assumption 1 (Bounded Hessian)**.** *There exist nonnegative real numbers $\lambda_{l,\min}$ and $\lambda_{l,\max}$ such that*

$$\lambda_{l,\min} I_M \le \nabla^2 J_l(w) \le \lambda_{l,\max} I_M \tag{15}$$

$$\sum_{l=1}^{N} s_{l,k}\lambda_{l,\min} > 0, \quad k = 1, \ldots, N \tag{16}$$

■

Assumption 2 (Gradient noise)**.** *Conditioned on the history up to time $i-1$, the noise $\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1})$ is zero mean, and its variance is upper bounded by the squared-norm of $\tilde{\boldsymbol{\phi}}_{k,i-1}$. Specifically, there exist $\alpha \ge 0$ and $\sigma_v^2 \ge 0$ such that, for all i, l, and k:*

$$\mathsf{E}\,\{\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1}) \mid \mathcal{F}_{i-1}\} = 0 \tag{17}$$

$$\mathsf{E}\left\{\|\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1})\|^2 \mid \mathcal{F}_{i-1}\right\} \le \alpha\|\tilde{\boldsymbol{\phi}}_{k,i-1}\|^2 + \sigma_v^2 \tag{18}$$

where $\mathcal{F}_{i-1} \triangleq \{\boldsymbol{w}_{k,j} : k = 1, \ldots, N \text{ and } j \le i-1\}$. ■

Lemma 1. *The matrix $\boldsymbol{H}_{l,k,i-1}$ defined in* (6) *is a nonnegative definite matrix that satisfies the following condition:*

$$\lambda_{l,\min} I_M \le \boldsymbol{H}_{l,k,i-1} \le \lambda_{l,\max} I_M \tag{19}$$

Proof. The result follows from (6) and (15). ■

Compared to the bounded gradient norm assumption in [10], Assumption 1 is more general in the sense that it allows the gradient vector $\nabla_w J_l(w)$ to have unbounded norm (as happens, for example, in the case of quadratic costs). Likewise, condition (18) allows the variance of the gradient noise to be unbounded and time-varying. This condition is more general than the "uniform bounded assumptions" in [10] (Assumptions 5.1 and 6.1), which are special cases of (18) for $\alpha = 0$. Furthermore, (18) is actually a combination of the "relative random noise" and the "absolute random noise" in [12, pp.100–102].

3.2. Variance Relations

Equating the squared *weighted* "norm" of both sides of recursion (14) and using (17), we obtain the following variance relation:

$$\boxed{\begin{aligned} &\mathsf{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathsf{E}\big\{\|\tilde{\boldsymbol{w}}_{i-1}\|_{\boldsymbol{\Sigma}'}^2\big\} + \mathsf{E}\|\mathcal{P}_2^T\mathcal{M}\boldsymbol{\mathcal{G}}_i\|_{\Sigma}^2 \\ &\boldsymbol{\Sigma}' = \mathcal{P}_1[I_{MN} - \mathcal{M}\boldsymbol{\mathcal{D}}_{i-1}]\mathcal{P}_2\Sigma\mathcal{P}_2^T[I_{MN} - \mathcal{M}\boldsymbol{\mathcal{D}}_{i-1}]\mathcal{P}_1^T \end{aligned}} \tag{20}$$

where Σ is a positive semi-definite matrix that we are free to choose. Notice that $\boldsymbol{\Sigma}'$ is a random matrix that depends on $\{\boldsymbol{H}_{l,k,i-1}\}$ via $\boldsymbol{\mathcal{D}}_{i-1}$ (see (12)). By the definition of $\boldsymbol{H}_{l,k,i-1}$ in (6), $\boldsymbol{\Sigma}'$ further depends on $\tilde{\boldsymbol{\phi}}_{k,i-1}$, which is a linear combination of $\{\tilde{\boldsymbol{w}}_{l,i-1}\}$. Therefore, the main challenge to continue from (20) is that $\boldsymbol{\Sigma}'$ now depends on $\tilde{\boldsymbol{w}}_{i-1}$. Then, we cannot apply the traditional step of replacing $\boldsymbol{\Sigma}'$ in the first equation of (20) by $\mathsf{E}\boldsymbol{\Sigma}'$ as in [13, p.345] to analyze the transient behavior of the algorithm. To address the difficulty, we modify the argument and rely first on a set of inequality recursions to bound the mean-square-error. Then, we return to (20) to evaluate the steady-state performance for small step-sizes.

First, we notice that (5a) and (5c) are convex combinations of $\{\tilde{\boldsymbol{w}}_{l,i-1}\}$ and $\{\tilde{\boldsymbol{\psi}}_{l,i}\}$, respectively. Then, by Jensen's inequality

$$\begin{cases} \mathsf{E}\|\tilde{\boldsymbol{\phi}}_{k,i-1}\|^2 \le \sum_{l=1}^{N} p_{1,l,k}\mathsf{E}\|\tilde{\boldsymbol{w}}_{l,i-1}\|^2 & \text{(21a)} \\ \mathsf{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \le \sum_{l=1}^{N} p_{2,l,k}\mathsf{E}\|\tilde{\boldsymbol{\psi}}_{l,i}\|^2 & \text{(21b)} \end{cases}$$

Next, evaluating $\mathsf{E}\|\tilde{\boldsymbol{\psi}}_{k,i}\|^2$ from (7) and using property (17), we get

$$\mathsf{E}\|\tilde{\boldsymbol{\psi}}_{k,i}\|^2 = \mathsf{E}\big\{\|\tilde{\boldsymbol{\phi}}_{k,i-1}\|_{\boldsymbol{\Sigma}_{k,i-1}}^2\big\} + \mu_k^2\mathsf{E}\Big\|\sum_{l=1}^{N} s_{l,k}\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1})\Big\|^2 \tag{22}$$

$$\boldsymbol{\Sigma}_{k,i-1} \triangleq \Big[I_M - \mu_k \sum_{l=1}^{N} s_{l,k}\boldsymbol{H}_{l,k,i-1}\Big]^2 \tag{23}$$

We call upon the following two lemmas to bound (22).

Lemma 2. *$\boldsymbol{\Sigma}_{k,i-1}$ is a positive semi-definite matrix that satisfies:*

$$0 \leq \boldsymbol{\Sigma}_{k,i-1} \leq \gamma_k^2 I_M \tag{24}$$

where

$$\gamma_k = \max\{|1-\mu_k\sigma_{k,\max}|, |1-\mu_k\sigma_{k,\min}|\} \tag{25}$$

$$\sigma_{k,\max} = \sum_{l=1}^{N} s_{l,k}\lambda_{l,\max}, \quad \sigma_{k,\min} = \sum_{l=1}^{N} s_{l,k}\lambda_{l,\min} \tag{26}$$

Proof. The proof follows from (19) and (23). ■

Lemma 3. *The second term on the right hand side of* (22) *satisfies:*

$$\mathsf{E}\Big\|\sum_{l=1}^{N} s_{l,k}\boldsymbol{v}_l(\boldsymbol{\phi}_{k,i-1})\Big\|^2 \leq \|S\|_1^2 \cdot \Big[\alpha\mathsf{E}\|\tilde{\boldsymbol{\phi}}_{k,i-1}\|^2 + \sigma_v^2\Big] \tag{27}$$

where $\|S\|_1$ denotes the maximum absolute column sum of matrix S.

Proof. The result follows by applying Jensen's inequality and is omitted for brevity – see [9]. ■

Substituting (24) and (27) into (22), we obtain:

$$\mathsf{E}\|\tilde{\boldsymbol{\psi}}_{k,i}\|^2 \leq (\gamma_k^2 + \mu_k^2\alpha\|S\|_1^2)\cdot\mathsf{E}\|\tilde{\boldsymbol{\phi}}_{k,i-1}\|^2 + \mu_k^2\|S\|_1^2\sigma_v^2 \tag{28}$$

Finally, introduce the global quantities:

$$\mathcal{X}_i = \mathrm{col}\big\{\mathsf{E}\|\tilde{\boldsymbol{\phi}}_{1,i}\|^2 \cdots \mathsf{E}\|\tilde{\boldsymbol{\phi}}_{N,i}\|^2\big\} \tag{29}$$

$$\mathcal{Y}_i = \mathrm{col}\big\{\mathsf{E}\|\tilde{\boldsymbol{\psi}}_{1,i}\|^2 \cdots \mathsf{E}\|\tilde{\boldsymbol{\psi}}_{N,i}\|^2\big\} \tag{30}$$

$$\mathcal{W}_i = \mathrm{col}\big\{\mathsf{E}\|\tilde{\boldsymbol{w}}_{1,i}\|^2 \cdots \mathsf{E}\|\tilde{\boldsymbol{w}}_{N,i}\|^2\big\} \tag{31}$$

$$\Gamma = \mathrm{diag}\left\{\gamma_1^2 + \mu_1^2\alpha\|S\|_1^2,\ \ldots\ ,\gamma_N^2 + \mu_N^2\alpha\|S\|_1^2\right\} \tag{32}$$

Then, (21) and (28) can be written as

$$\mathcal{X}_{i-1} \preceq P_1^T\mathcal{W}_{i-1},\ \ \mathcal{Y}_i \preceq \Gamma\mathcal{X}_{i-1} + \sigma_v^2\|S\|_1^2\Omega^2\mathbb{1},\ \ \mathcal{W}_i \preceq P_2^T\mathcal{Y}_i \tag{33}$$

where the notation $x \preceq y$ denotes that the components of vector x are no greater than the corresponding components of vector y. Then, expression (33) can be shown to lead to:

$$\boxed{\mathcal{W}_i \preceq P_2^T\Gamma P_1^T\mathcal{W}_{i-1} + \sigma_v^2\|S\|_1^2\cdot P_2^T\Omega^2\mathbb{1}} \tag{34}$$

3.3. Mean-Square Stability

Based on (34), we can now argue that, under certain conditions on the step-size parameters $\{\mu_k\}$, the mean-square-error vector $\mathcal{W}_i$ is bounded as $i \to \infty$, and we use this result in the next subsection to evaluate the steady-state value for the mean-square error for sufficiently small step-sizes. We can also give an estimate for the convergence rate by examining the spectral radius of the matrix $P_2^T\Gamma P_1^T$.

Theorem 1 (Mean-Square Stability)**.** *If the step-sizes $\{\mu_k\}$ satisfy the following condition:*

$$\boxed{0 < \mu_k < \min\left\{\frac{2\sigma_{k,\max}}{\sigma_{k,\max}^2 + \alpha\|S\|_1^2}, \frac{2\sigma_{k,\min}}{\sigma_{k,\min}^2 + \alpha\|S\|_1^2}\right\}} \tag{35}$$

for $k = 1, \ldots, N$. Then, as $i \to \infty$, the following bound holds:

$$\boxed{\lim_{i\to\infty}\|\mathcal{W}_i\|_\infty \leq \frac{\left(\max\limits_{1\leq k\leq N}\mu_k^2\right)\cdot\|S\|_1^2\sigma_v^2}{1 - \max\limits_{1\leq k\leq N}(\gamma_k^2 + \mu_k^2\alpha\|S\|_1^2)}} \tag{36}$$

where $\|x\|_\infty$ denotes the maximum absolute entry of vector x.

Proof. Omitted due to space limitation (see [9]). ■

3.4. Steady-State Performance

Expression (36) gives a bound on how large the steady-state value of $\mathcal{W}_i$ can be. Now, we derive an approximate expression for the steady-state value for small step-sizes — see (45) further ahead. We further assume the following condition on the gradient noise.

Assumption 3 (Gradient noise model)**.** *Assume the gradient noise vector $\boldsymbol{\mathcal{G}}_i$ defined in* (13) *satisfies:*

$$\mathsf{E}\{\boldsymbol{\mathcal{G}}_i\boldsymbol{\mathcal{G}}_i^T\} = \alpha\mathsf{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2\cdot Q_{i-1}^o + R_v \tag{37}$$

where $\|Q_{i-1}^o\| \leq 1$, and R_v is a constant matrix. ■

Assumption (37) is a matrix analog of (18), where the two terms on the right-hand side correspond to the "relative random noise" and "absolute random noise" parts [12], respectively. As a result,

$$\mathsf{E}\|\mathcal{P}_2^T\mathcal{M}\boldsymbol{\mathcal{G}}_i\|_\Sigma^2 = \mathsf{E}\boldsymbol{\mathcal{G}}_i^T\mathcal{M}\mathcal{P}_2\Sigma\mathcal{P}_2^T\mathcal{M}\boldsymbol{\mathcal{G}}_i =$$
$$\alpha\mathsf{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2\mathrm{Tr}\Big(\Sigma\mathcal{P}_2^T\mathcal{M}Q_{i-1}^o\mathcal{M}\mathcal{P}_2\Big) + \mathrm{Tr}\Big(\Sigma\mathcal{P}_2^T\mathcal{M}R_v\mathcal{M}\mathcal{P}_2\Big) \tag{38}$$

In order to evaluate the steady-state performance, we first approximate (20) at small step-sizes. From (36), as the algorithm reaches steady-state, the mean-squared value of $\{\tilde{\boldsymbol{w}}_{k,i}\}$ is small at small step-sizes. Hence, $\tilde{\boldsymbol{\phi}}_{k,i-1}$ is also small because it is a convex combination of $\{\tilde{\boldsymbol{w}}_{k,i-1}\}$ (see (5a)). Then, by definition (6), $\boldsymbol{H}_{l,k,i-1}$ can be approximated by $\boldsymbol{H}_{l,k,i-1} \approx \nabla_w^2 J_l(w^o)$. Thus,

$$\boldsymbol{\mathcal{D}}_{i-1} \approx \mathcal{D}_\infty \triangleq \sum_{l=1}^{N}\mathrm{diag}\big\{s_{l,1}\nabla^2 J_l(w^o), \cdots, s_{l,N}\nabla^2 J_l(w^o)\big\} \tag{39}$$

Substituting (38)–(39) into (20), an approximate energy conservation relation is obtained at small step-sizes:

$$\mathsf{E}\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 \approx \mathsf{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma''}^2 + \mathrm{Tr}\left(\Sigma\mathcal{P}_2^T\mathcal{M}R_v\mathcal{M}\mathcal{P}_2\right) \tag{40}$$

$$\Sigma'' \approx \mathcal{P}_1[I_{MN} - \mathcal{M}\mathcal{D}_\infty]\mathcal{P}_2\Sigma\mathcal{P}_2^T[I_{MN} - \mathcal{M}\mathcal{D}_\infty]\mathcal{P}_1^T \tag{41}$$

Let $\sigma = \mathrm{vec}(\Sigma)$ denote the vectorization operation; it stacks the columns of Σ on top of each other. We shall use both notation $\|\tilde{w}\|_\sigma^2$ and $\|\tilde{w}\|_\Sigma^2$ interchangeably. Using the property $\mathrm{vec}(U\Sigma V) = (V^T\otimes U)\mathrm{vec}(\Sigma)$, we can vectorize Σ'' in (41) as $\sigma'' \triangleq \mathrm{vec}(\Sigma'') \approx F\sigma$ where

$$\boxed{F \triangleq \big(\mathcal{P}_1[I_{MN} - \mathcal{M}\mathcal{D}_\infty]\mathcal{P}_2\big)\otimes\big(\mathcal{P}_1[I_{MN} - \mathcal{M}\mathcal{D}_\infty]\mathcal{P}_2\big)} \tag{42}$$

and where we used the fact that $\mathcal{D}_\infty$ is symmetric. Furthermore, using the property $\mathrm{Tr}(\Sigma X) = \mathrm{vec}(X^T)^T\sigma$, we can rewrite (40) as

$$\mathsf{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 \ \approx \mathsf{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{F\sigma}^2 + \Big[\mathrm{vec}\left(\mathcal{P}_2^T\mathcal{M}R_v\mathcal{M}\mathcal{P}_2\right)\Big]^T\sigma \tag{43}$$

Finally, letting $i \to \infty$, we get

$$\boxed{\lim_{i\to\infty}\mathsf{E}\|\tilde{\boldsymbol{w}}_i\|_{(I-F)\sigma}^2 \approx \Big[\mathrm{vec}\left(\mathcal{P}_2^T\mathcal{M}R_v\mathcal{M}\mathcal{P}_2\right)\Big]^T\sigma} \tag{44}$$

To evaluate steady-state performance from (44), we need $I - F$ to be invertible, which is guaranteed by (35).

Then, from (44), we can evaluate various performance metrics by choosing proper weighting matrices Σ (or σ), as was done in [6, 7]. For example, let $m_k = \mathrm{vec}(\mathrm{diag}(e_k)\otimes I_M)$, where e_k is a vector whose kth entry is one and zeros elsewhere. Then, letting

$\sigma = (I-F)^{-1}m_k$ in (44), we get the MSD for node k, which is the kth entry of the vector

$$\boxed{\lim_{i\to\infty}\mathcal{W}_i = \left\{I_N \otimes \left(\left[\mathrm{vec}\left(\mathcal{P}_2^T \mathcal{M} R_v \mathcal{M} \mathcal{P}_2\right)\right]^T (I-F)^{-1}\right)\right\} m} \quad (45)$$

where $m = \mathrm{col}\{m_1, \ldots, m_N\}$. And the average network MSD is

$$\mathrm{MSD}^{\mathrm{network}} = \lim_{i\to\infty} \frac{1}{N} \mathbb{1}^T \mathcal{W}_i \quad (46)$$

4. SIMULATION RESULTS

We consider a randomly generated 10-node network ($N = 10$). Each node k has access to data $\{\boldsymbol{U}_{k,i}, \boldsymbol{d}_{k,i}\}$, satisfying the model $\boldsymbol{d}_{k,i} = \boldsymbol{U}_{k,i} w^o + \boldsymbol{v}_{k,i}$, where the entries of each $\boldsymbol{U}_{k,i}$ are i.i.d. Gaussian random variables with zero mean and unit variance, and $\boldsymbol{v}_{k,i} \sim \mathcal{N}(0, \sigma^2 I_K)$ is independent of $\boldsymbol{U}_{k,i}$. Our objective is to estimate w^o from the data set $\{\boldsymbol{U}_{k,i}, \boldsymbol{d}_{k,i}\}$ in a distributed manner. We assume the vector w^o is sparse so that it has only a few nonzero entries, such as $w^o = [1\ 0\ \ldots\ 0\ 1]^T$. This application motivates us to consider a global cost of the form:

$$J^{\mathrm{glob}}(w) = \sum_{l=1}^{N} \left[\mathsf{E}\|\boldsymbol{d}_{l,i} - \boldsymbol{U}_{l,i} w\|_2^2 + \frac{\gamma}{N} R(w)\right] = \sum_{l=1}^{N} J_l(w) \quad (47)$$

where $R(w)$ and γ are the regularization function and regularization factor, respectively. A popular choice is $R(w) = \|w\|_1$, which helps enforce sparsity and is convex. However, it is non-differentiable. Instead, we use a twice-differentiable approximation for $\|w\|_1$, namely,

$$R(w) = \sum_{m=1}^{M} \sqrt{[w]_m^2 + \epsilon^2} \quad (48)$$

where $[w]_m$ denotes the mth entry of w, and ϵ is a small number.

In the simulation, we set $M = 50$, $K = 5$, $\sigma^2 = 1$, and w^o as the $M \times 1$ vector given previously. We apply both diffusion and incremental methods to solve the problem; the results are averaged over 100 trials. The step-sizes for ATC and CTA are set to $\mu = 10^{-3}$, and the step-size for the incremental algorithm [3] is $10^{-3}/N$. This is because the incremental algorithm goes through all N nodes during each iteration, and we need to ensure the same convergence rate for both incremental and diffusion algorithms for a fair comparison. We use the network $\mathrm{MSD}^{\mathrm{network}}$ to measure performance. Fig. 1 shows the learning curves of different algorithms for $\gamma = 2$ and $\epsilon = 10^{-3}$. We see that diffusion and incremental schemes have similar performance, both having 10 dB gain over the non-cooperative case. To examine the impact of the parameter ϵ and regularization factor γ, we show the steady-state MSD versus γ and ϵ in Fig. 2. When ϵ is small ($\epsilon = 10^{-4}$), adding a reasonable regularization ($\gamma = 1 \sim 4$) decreases the steady-state MSD (even for the individual case). However, when ϵ is large ($\epsilon = 1$), this choice of $R(w)$ is no longer a good approximation for $\|w\|_1$, and regularization does not improve the MSD.

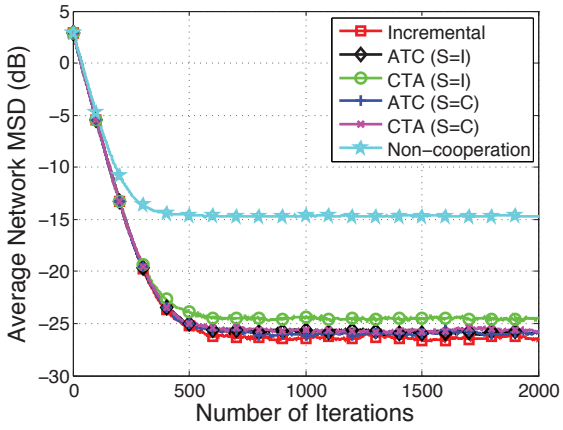

Fig. 1. Learning curves for different algorithms ($\gamma = 2$, $\epsilon = 10^{-3}$).

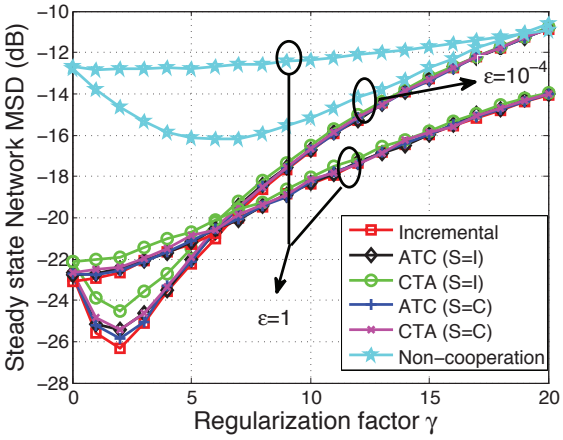

Fig. 2. Steady-state MSD for different values of ϵ and γ ($\mu = 10^{-3}$).

5. REFERENCES

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

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

[3] 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.

[4] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, MA, June 2006.

[5] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. of Electron., Commun. and Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, 2007.

[6] 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, July 2008.

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

[8] J. Chen, S.-Y. Tu, and A. H. Sayed, "Distributed optimization via diffusion adaptation," in *Proc. IEEE International Workshop on Comput. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, Puerto Rico, Dec. 2011.

[9] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *submitted for publication (also available at Arxiv preprint arXiv:1111.0034)*, Oct. 2011.

[10] 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.

[11] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz, "Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates," in *Proc. IEEE ICASSP*, Prague, Czech, May 2011, pp. 3764–3767.

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

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