GRAMIAN-BASED ADAPTIVE COMBINATION POLICIES FOR DIFFUSION LEARNING OVER NETWORKS

Y. Efe Erginbas ⋆, *Stefan Vlaski* † *and Ali H. Sayed* †

ABSTRACT

This paper presents an adaptive combination strategy for distributed learning over diffusion networks. Since learning relies on the collaborative processing of the stochastic information at the dispersed agents, the overall performance can be improved by designing combination policies that adjust the weights according to the quality of the data. Such policies are important because they would add a new degree of freedom and endow multi-agent systems with the ability to control the flow of information over their edges for enhanced performance. Most adaptive and static policies available in the literature optimize certain performance metrics related to steady-state behavior, to the detriment of transient behavior. In contrast, we develop an adaptive combination rule that aims at optimizing the transient learning performance, while maintaining the enhanced steady-state performance obtained using policies previously developed in the literature.

Index Terms— distributed learning, diffusion strategy, combination weights, adaptive network.

1. INTRODUCTION

We consider a strongly-connected network of N nodes with a predefined topology. We denote the neighborhood of node k (including node k itself) by $\mathcal{N}_k$ and the degree of node k by n_k. A local risk function $J_k(w) = \mathbb{E}_x \mathcal{Q}_k(w; \boldsymbol{x})$ is associated with each node, where $\mathcal{Q}_k(w; \boldsymbol{x})$ is some loss function. The objective is to generate an estimate, in a collaborative and distributed manner, for the unknown vector $w^o \in \mathbb{R}^M$ that minimizes the global cost:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{1}$$

Each $J_k(w)$ is a real valued function defined over $w \in \mathbb{R}^M$, and assumed to be differentiable and strongly convex. Consequently, $J^{\text{glob}}(w)$ is also a strongly convex and the minimizer w^o is unique [1]. In this work, we focus on the important case where each of the local cost functions $\{J_k(w)\}$ are also minimized at the same w^o.

The solution to this problem can be pursued in a decentralized and iterative manner by generating local estimates $w_{k,i}$ at each node k and time $i \geq 0$. The iterates can be constructed using a variety of decentralized strategies, including incremental [2], consensus [3–5], diffusion [6, 7], primal-dual [8, 9], proximal [10], augmented Lagrangian [11], or gradient tracking methods [12]. Here, we focus on the *Adapt-then-Combine* (ATC) diffusion strategy, which has been shown to have superior mean-square error performance and stability range in adaptive scenarios [7].

Since the statistical distribution of the data $\boldsymbol{x}$ is not known beforehand in most cases of interest, the exact gradient vectors $\nabla_w J_k(w)$ are not available or easily obtainable. Motivated by this consideration, we follow the stochastic gradient descent construction for diffusion algorithms and utilize gradient approximations in our analysis. We refer to the perturbation as a random additive noise component and write the approximate gradient vector in the form:

$$\widehat{\nabla_w J}_k(w) = \nabla_w J_k(w) + \boldsymbol{s}_{k,i}(w) \tag{2}$$

where $\boldsymbol{s}_{k,i}(w)$ denotes the gradient noise term. Note that we use a boldface symbol to highlight its stochastic nature. Using the perturbed gradient vectors, the ATC diffusion strategy is given by:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) & \text{(3a)} \\ \boldsymbol{w}_{k,i} = \displaystyle\sum_{\ell \in \mathcal{N}_k} \boldsymbol{a}_{\ell k,i} \boldsymbol{\psi}_{\ell,i} & \text{(3b)} \end{cases}$$

where $\boldsymbol{\psi}_{k,i}$ is an intermediate local estimation vector at node k, μ_k is the step size at node k and $\{\boldsymbol{a}_{\ell k,i}\}$ are possibly time-varying, real-valued combination weights. In most of the prior works on decentralized processing, the combination weights are pre-selected and treated as known deterministic variables. However, in this work, we focus on the case where these combination weights are constructed on the fly, and will therefore be data-dependent. They will nevertheless be always normalized to satisfy:

$$\sum_{\ell \in \mathcal{N}_k} \boldsymbol{a}_{\ell k,i} = 1, \quad \boldsymbol{a}_{\ell k,i} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{4}$$

Although the literature on diffusion strategies mostly includes works where the combination weights are constrained to be non-negative [6,7], there also exist works that allow negative values [13]. We will not impose the non-negativity requirement as well.

Since the choice of the combination weights $\{\boldsymbol{a}_{\ell k,i}\}$ plays an important role in the performance of decentralized strategies, different static and adaptive combination policies have been proposed in previous studies. The static combination policies include Uniform [14], Laplacian [15, 16], Maximum-Degree [17], Metropolis [16], Hastings [18] and Relative-Variance [19] rules. To allow for data-aware processing, adaptive combination policies have also been developed. These policies learn data statistics during the operation of the algorithm and adapt combination weights accordingly. Examples include the adaptive version of Relative-Variance rule [19], Phase-Switching algorithm [20], and the policy proposed in [21].

Previous works on adaptive combiners mostly concentrate on improving the steady-state performance of the nodes by incorporating the information obtained during the learning process [19, 20]. However, these algorithms do not explicitly aim at improving the transient performance. Some empirical results indicate that they can be outperformed by simple static combination policies (such as the averaging rule) in the transient phase [6, 19]. We address this problem in the context of networks with a general cost structure described previously, and propose an adaptive combination policy that will improve the transient performance of the decentralized learning algorithms while preserving the improved steady-state performance.

⋆ Y. Efe Erginbas is with Department of Electrical and Electronics Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey. The author performed the work while at EPFL, Switzerland. E-mail: efe.erginbas@ug.bilkent.edu.tr

† Stefan Vlaski and Ali H. Sayed are with School of Engineering, EPFL, CH-1015 Lausanne, Switzerland. E-mails: {stefan.vlaski, ali.sayed}@epfl.ch

2. GRAMIAN-BASED ADAPTIVE POLICY

We first describe the error recursions corresponding to the ATC diffusion formulation in (3) and start by defining the error vectors

$$\widetilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}\,, \qquad \widetilde{\boldsymbol{\psi}}_{k,i} = w^o - \boldsymbol{\psi}_{k,i} \tag{5}$$

for all $k = 1, \dots, N$ and $i \geq 0$. Following the approach taken in [6], we subtract both sides of equations (3a) and (3b) from w^o, substitute expression (2) for the perturbed gradient vector, and call upon the mean-value theorem to express the error recursion as:

$$\begin{cases} \widetilde{\boldsymbol{\psi}}_{k,i} = [I_M - \mu_k \boldsymbol{H}_{k,i-1}]\widetilde{\boldsymbol{w}}_{k,i-1} + \mu_k \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) & \text{(6a)} \\ \widetilde{\boldsymbol{w}}_{k,i} = \sum\limits_{\ell \in \mathcal{N}_k} \boldsymbol{a}_{\ell k,i} \widetilde{\boldsymbol{\psi}}_{\ell,i} & \text{(6b)} \end{cases}$$

where

$$\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla_w^2 J_k(w^o - t\widetilde{\boldsymbol{w}}_{k,i-1})dt \tag{7}$$

and $\nabla^2 J_k(\cdot)$ denotes the Hessian matrix of $J_k(\cdot)$. We also define the combination vectors and the combination matrix as follows:

$$\begin{aligned} \boldsymbol{a}_{k,i} &\triangleq \text{col}\{\boldsymbol{a}_{1k,i}, \dots, \boldsymbol{a}_{Nk,i}\} \in \mathbb{R}^N \text{ for } k = 1, \dots, N \\ \boldsymbol{A}_i &\triangleq [\boldsymbol{a}_{1,i}, \dots, \boldsymbol{a}_{N,i}] \in \mathbb{R}^{N \times N} \end{aligned} \tag{8}$$

2.1. Development of the Algorithm

We define the problem of selecting the combination weights as an optimization problem where our goal is to maximize the improvement obtained in each time step by estimating the best possible combination weighting for each node. For this purpose, we introduce a performance metric for the network and formulate the optimization problem using this metric. We construct the network Square-Deviation measure as follows:

$$\mathbf{SD}_{\text{av}}(i) \triangleq \frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2 = \frac{1}{N}\sum_{k=1}^N \|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{9}$$

Although it is similar to the widely-used MSD metric [6], SD measures the error norm at each iteration i in contrast to MSD that measures the expected norm of the steady-state error. For completeness, we also provide the standard definition for the average Mean-Square-Deviation metric as $\text{MSD}_{\text{av}} \triangleq \lim_{i \to \infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2/N$.

In order to account for error similarity between the nodes, we construct the Gramian matrix $\boldsymbol{Q}_i$ for the set of intermediate estimation error vectors $\{\widetilde{\boldsymbol{\psi}}_{k,i}\}$:

$$\boldsymbol{Q}_i \triangleq \widetilde{\boldsymbol{\Psi}}_i^{\mathsf{T}} \widetilde{\boldsymbol{\Psi}}_i \tag{10}$$

where $\widetilde{\boldsymbol{\Psi}}_i \triangleq \left[\widetilde{\boldsymbol{\psi}}_{1,i}, \dots, \widetilde{\boldsymbol{\psi}}_{N,i}\right]$. Using these definitions, we can express (6b) as a matrix vector product:

$$\widetilde{\boldsymbol{w}}_{k,i} = \widetilde{\boldsymbol{\Psi}}_i \boldsymbol{a}_{k,i} \tag{11}$$

In order to pursue an optimal combination policy $\boldsymbol{A}_i$ as a function of the estimation errors $\widetilde{\boldsymbol{\psi}}_{k,i}$, we define an objective function using the SD measure defined in (9) and substitute (10)-(11) to get

$$\mathbf{SD}_{\text{av}}(i) = \frac{1}{N}\sum_{k=1}^N \|\widetilde{\boldsymbol{\Psi}}_i \boldsymbol{a}_{k,i}\|^2 = \frac{1}{N}\sum_{k=1}^N \boldsymbol{a}_{k,i}^{\mathsf{T}} \boldsymbol{Q}_i \boldsymbol{a}_{k,i} \tag{12}$$

Consequently, the problem of finding the combination weights that minimize the given error measure can be expressed as a constrained minimization problem in the form:

$$\begin{aligned} \min_{\{\boldsymbol{a}_{\ell k,i}\}} \quad & \sum_{k=1}^N \boldsymbol{a}_{k,i}^{\mathsf{T}} \boldsymbol{Q}_i \boldsymbol{a}_{k,i} \\ \text{s.t.} \quad & \boldsymbol{a}_{\ell k,i} = 0, \text{ for all } \ell \notin \mathcal{N}_k \\ & \boldsymbol{a}_{k,i}^{\mathsf{T}} \mathbb{1} = 1, \text{ for } k = 1, \dots, N \end{aligned} \tag{13}$$

Since some entries of the vector $\boldsymbol{a}_{k,i}$ are constrained to be zero, we only need to solve this quadratic minimization problem for entries $\boldsymbol{a}_{\ell k,i}$ such that $\ell \in \mathcal{N}_k$. Therefore, we define the truncated combination vectors:

$$\boldsymbol{c}_{k,i} \triangleq \text{col}\{\boldsymbol{a}_{\ell k,i}\}_{\ell \in \mathcal{N}_k} \in \mathbb{R}^{n_k} \tag{14}$$

such that

$$\boldsymbol{a}_{k,i} = P_k \boldsymbol{c}_{k,i} \tag{15}$$

where $P_k \triangleq [\dots, \ell^{\text{th}} \text{ column of } I_N, \dots]$ for $\ell \in \mathcal{N}_k$ (i.e, P_k is the $N \times n_k$ matrix whose columns are standard (natural) basis vectors of indices corresponding to the neighbors of node k). Additionally, we define the local counterparts of the Gramian matrix and error matrix as follows:

$$\widetilde{\boldsymbol{\Psi}}_{k,i} \triangleq \widetilde{\boldsymbol{\Psi}}_i P_k \tag{16a}$$

$$\boldsymbol{Q}_{k,i} \triangleq \widetilde{\boldsymbol{\Psi}}_{k,i}^{\mathsf{T}} \widetilde{\boldsymbol{\Psi}}_{k,i} = P_k^{\mathsf{T}} \boldsymbol{Q}_i P_k \tag{16b}$$

Furthermore, in (13), neither the k^{th} term of the summation in the objective function nor the constraints for $\boldsymbol{a}_{k,i}$ depend on the selection of $\boldsymbol{a}_{\ell,i}$ for $\ell \neq k$. Therefore, this optimization problem can be decoupled into N independent sub-problems. Using the notation introduced in (15), we express the sub-problem related to node k as:

$$\begin{aligned} \min_{\boldsymbol{c}_{k,i}} \quad & \boldsymbol{c}_{k,i}^{\mathsf{T}} \boldsymbol{Q}_{k,i} \boldsymbol{c}_{k,i} \\ \text{s.t.} \quad & \boldsymbol{c}_{k,i}^{\mathsf{T}} \mathbb{1} = 1 \end{aligned} \tag{17}$$

We continue with writing the optimality conditions as a KKT system for this equality constrained quadratic optimization problem:

$$\begin{bmatrix} \boldsymbol{Q}_{k,i} & \mathbb{1}_{n_k} \\ \mathbb{1}_{n_k}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_{k,i} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0_{n_k} \\ 1 \end{bmatrix} \tag{18}$$

For general quadratic optimization problems, if the KKT system is not solvable, it means that the problem is unbounded below or infeasible [22, p. 522]. However, $\boldsymbol{Q}_{k,i}$ is certainly positive-semidefinite (since it is a Gramian matrix) and thus the minimum value of the objective is bounded below by zero. Additionally, we note that the feasible set given by the linear constraint is non-empty. Therefore, solving (18) will always yield an optimal $\boldsymbol{c}_{k,i}$ (there may be multiple optimal solutions if the KKT matrix is singular).

If $\boldsymbol{Q}_{k,i}$ turns out to be non-singular, it is straightforward to verify that the optimal solution can be found in the following closed-form:

$$\boldsymbol{c}_{k,i} = \frac{\boldsymbol{Q}_{k,i}^{-1} \mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^{\mathsf{T}} \boldsymbol{Q}_{k,i}^{-1} \mathbb{1}_{n_k}} \tag{19}$$

The resulting expressions enable us to compute the combination weights as a function of the matrix $\boldsymbol{Q}_{k,i}$ and hence $\widetilde{\boldsymbol{\psi}}_{k,i}$. However, these quantities are not known for the nodes, since the nodes only have access to $\boldsymbol{\psi}_{k,i}$, but not w^o. Consequently, the difficulty in employing the given combination policy, while optimal in the sense that it maximizes the reduction in squared deviation, lies in the fact that we are required to estimate the statistics of the error vectors.

In order to construct an estimate for the matrix $\boldsymbol{Q}_{k,i}$, we adopt a sample mean approach. We recall the definition of $\boldsymbol{Q}_{k,i}$ and express it in terms of the known quantities $\{\boldsymbol{\psi}_{k,i}\}$ as

$$\boldsymbol{Q}_{k,i} \triangleq \widetilde{\boldsymbol{\Psi}}_{k,i}^{\mathsf{T}} \widetilde{\boldsymbol{\Psi}}_{k,i} = (\boldsymbol{\Psi}_{k,i} - w^o \mathbb{1}_{n_k}^{\mathsf{T}})^{\mathsf{T}} (\boldsymbol{\Psi}_{k,i} - w^o \mathbb{1}_{n_k}^{\mathsf{T}}) \tag{20}$$

where $\boldsymbol{\Psi}_{k,i} \triangleq \left[\dots, \boldsymbol{\psi}_{\ell,i}, \dots\right]$ for $\ell \in \mathcal{N}_k$. However, estimating $\boldsymbol{Q}_{k,i}$ directly in this form is difficult because it requires knowledge of the optimal weight vector w^o. We follow the reasoning used in [19, 21], and approximate the optimal w^o by the expected estimate at iterate i, i.e., $w^o \approx \mathbb{E}\boldsymbol{\psi}_{k,i}$. Consequently, an estimate for $\boldsymbol{Q}_{k,i}$ will be

$$\widehat{\boldsymbol{Q}}_{k,i} \approx (\boldsymbol{\Psi}_{k,i} - \mathbb{E}\boldsymbol{\Psi}_{k,i})^{\mathsf{T}}(\boldsymbol{\Psi}_{k,i} - \mathbb{E}\boldsymbol{\Psi}_{k,i}) \tag{21}$$

Using the approximation that $\boldsymbol{Q}_{k,i}$ is locally stationary, we propose using an exponential moving average scheme to construct $\widehat{\boldsymbol{Q}}_{k,i}$ as:

$$\begin{aligned}\widehat{\boldsymbol{Q}}_{k,i} &= (1-\alpha_1)\widehat{\boldsymbol{Q}}_{k,i-1} + \alpha_1(\boldsymbol{\Psi}_{k,i} - \bar{\boldsymbol{\Psi}}_{k,i-1})^{\mathsf{T}}(\boldsymbol{\Psi}_{k,i} - \bar{\boldsymbol{\Psi}}_{k,i-1})\\ \bar{\boldsymbol{\Psi}}_{k,i} &= (1-\alpha_2)\bar{\boldsymbol{\Psi}}_{k,i-1} + \alpha_2\boldsymbol{\Psi}_{k,i}\end{aligned} \tag{22}$$

for some constants $0 < \alpha_1, \alpha_2 \ll 1$. Using this estimation scheme, we can complete the construction of the algorithm as follows:

```
Algorithm 1: Gramian-Based Adaptive Diffusion
Parameters and initialization: $0 < \alpha_1, \alpha_2 \ll 1$, $\mu_k > 0$,
  $\widehat{\boldsymbol{Q}}_{k,-1} = I_{n_k}$, $\bar{\boldsymbol{\Psi}}_{k,-1} = 0_{M\times n_k}$, $\boldsymbol{w}_{k,-1} = 0_M$.
for each time $i \ge 0$ do
    for each node $k$ do
        $\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1})$
    end
    for each node $k$ do
        $\boldsymbol{\Psi}_{k,i} = \big[\dots, \boldsymbol{\psi}_{\ell,i}, \dots\big]$ , for $\ell \in \mathcal{N}_k$
        $\boldsymbol{G}_{k,i} = (\boldsymbol{\Psi}_{k,i} - \bar{\boldsymbol{\Psi}}_{k,i-1})^{\mathsf{T}}(\boldsymbol{\Psi}_{k,i} - \bar{\boldsymbol{\Psi}}_{k,i-1})$
        $\widehat{\boldsymbol{Q}}_{k,i} = (1-\alpha_1)\widehat{\boldsymbol{Q}}_{k,i-1} + \alpha_1\boldsymbol{G}_{k,i}$
        $\bar{\boldsymbol{\Psi}}_{k,i} = (1-\alpha_2)\bar{\boldsymbol{\Psi}}_{k,i-1} + \alpha_2\boldsymbol{\Psi}_{k,i}$
        solve $\begin{bmatrix}\widehat{\boldsymbol{Q}}_{k,i} & \mathbb{1}_{n_k}\\ \mathbb{1}_{n_k}^{\mathsf{T}} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{c}_{k,i}\\ \lambda\end{bmatrix} = \begin{bmatrix}0_{n_k}\\ 1\end{bmatrix}$ , for $\boldsymbol{c}_{k,i}$
        $\boldsymbol{w}_{k,i} = \boldsymbol{\Psi}_{k,i}\boldsymbol{c}_{k,i}$
    end
end
```

3. STEADY-STATE MEAN-SQUARE PERFORMANCE

To examine the performance of the algorithm, it is necessary to introduce some simplifying assumptions; otherwise, the analysis becomes intractable due to the multiple adaptation layers. Some of these assumptions are common in the literature on decentralized adaptive algorithms. They essentially essentially require that the construction of the approximate gradient vector should not introduce bias and that its error variance should decrease as the quality of the iterate improves [6].

Assumption 1 (Conditions on Gradient Noise). *The gradient noise processes are temporally and spatially independent. Additionally, the first and second-order moments of the gradient noise processes satisfy the following conditions:*

$$\mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\right] = 0 \tag{23a}$$

$$\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2\right] \le \beta^2\mathbb{E}\left[\|w^o - \boldsymbol{w}_{k,i-1}\|^2\right] + \sigma_{s,k}^2 \tag{23b}$$

for some constants $\beta^2 \ge 0$, $\sigma_{s,k}^2 \ge 0$. □

For static combination matrices that satisfy (4), it has been shown that $\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ can be made arbitrarily small in steady-state if sufficiently small step-sizes are used [6, 7]. Consequently, it has been argued that $\boldsymbol{H}_{k,i-1}$ can be approximated by $H_k^o \triangleq \nabla_w^2 J_k(w^o)$ for small step-sizes. In light of this observation, we employ the assumption that similar arguments will hold for our dynamic combination policy. In other words, we argue that the iterates can get sufficiently close to w^o such that the curvature and noise structures can be well-approximated with their corresponding values at w^o.

Furthermore, as done in the analysis of other adaptive policies in the literature [19, 21], we will assume that the estimation process is wide-sense stationary and samples are temporally uncorrelated in steady-state. This follows from the observation that the estimates $\boldsymbol{w}_{k,i}$ will approach w^o in expectation and the deviations will be mostly caused by independent noise factors.

Assumption 2 (Steady-State). *Using small enough step-sizes, it is assumed that the following stationarity conditions are approximately valid:*

$$\lim_{i\to\infty} \boldsymbol{H}_{k,i-1} \approx H_k^o \tag{24a}$$

$$\lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\boldsymbol{s}_{k,i}^{\mathsf{T}}(\boldsymbol{w}_{k,i-1})\right] \approx R_{s,k} \tag{24b}$$

$$\lim_{i\to\infty} \mathbb{E}\boldsymbol{\Psi}_i \approx \mathbb{E}\boldsymbol{\Psi} \tag{24c}$$

$$\lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{\Psi}_i^{\mathsf{T}}\boldsymbol{\Psi}_{i-1}\right] \approx \mathbb{E}\boldsymbol{\Psi}^{\mathsf{T}}\mathbb{E}\boldsymbol{\Psi} \tag{24d}$$

where $H_k^o \triangleq \nabla_w^2 J_k(w^o)$, $R_{s,k} \triangleq \mathbb{E}\left[\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}_{k,i}^{\mathsf{T}}(w^o)\right]$ and $\mathbb{E}\boldsymbol{\Psi}$ is an unknown deterministic matrix. □

Lastly, we will require the independence of each combination matrix from the last estimation error. For small enough α_1 and α_2 values, the combination matrix $\boldsymbol{A}_i$ at step i is a function of many past noise samples, while the estimation error $\widetilde{\boldsymbol{w}}_{i-1}$ at step i mostly depends on more recent noise samples. Therefore, it is reasonable to employ the assumption that two quantities are independent.

Assumption 3 (Independent Combination Matrix). *For small α_1 and α_2 values, $\boldsymbol{A}_i$ and $\widetilde{\boldsymbol{w}}_{k,i-1}$ are independent in steady-state.*

3.1. Approximate Steady-State Combination Policy

Since the expression for the combination policy and the distribution of $\boldsymbol{\Psi}_i$ matrices are intertwined with each other, it is not straightforward to write down a steady-state expression for $\mathbb{E}\boldsymbol{A}_i$. However, we can introduce the assumption that $\mathbb{E}\boldsymbol{A}_i$ will converge to some constant matrix, following [21]. This assumption originates from the observation that $\boldsymbol{A}_i$ matrices change slowly over iterations (for small α_1 and α_2 values) and therefore capture information mostly about the error statistics of the nodes. Furthermore, we assume that this constant matrix is such that it minimizes the expected error at the nodes and hence it is equal to the combination matrix generated by (19) using $\lim_{i\to\infty}\mathbb{E}\widehat{\boldsymbol{Q}}_{k,i}$ for $\boldsymbol{Q}_{k,i}$. Essentially, we assume that the expectation applied to both sides of (19) can be approximated as follows in the steady-state limit:

$$\mathbb{E}\boldsymbol{c}_{k,i} = \mathbb{E}\left[\frac{\boldsymbol{Q}_{k,i}^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^{\mathsf{T}}\boldsymbol{Q}_{k,i}^{-1}\mathbb{1}_{n_k}}\right] \approx \frac{\mathbb{E}\left[\boldsymbol{Q}_{k,i}\right]^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^{\mathsf{T}}\mathbb{E}\left[\boldsymbol{Q}_{k,i}\right]^{-1}\mathbb{1}_{n_k}} \tag{25}$$

Assumption 4. *Expected value of the combination matrix $\boldsymbol{A}_i$ converges to A_∞ in steady-state regime and we can employ following approximations:*

$$\lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{A}_i\right] \approx A_\infty \tag{26a}$$

$$\lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{\mathcal{A}}_i \otimes \boldsymbol{\mathcal{A}}_i\right] \approx \mathcal{A}_\infty \otimes \mathcal{A}_\infty \tag{26b}$$

where $\boldsymbol{\mathcal{A}}_i = \boldsymbol{A}_i \otimes I_M$, $\mathcal{A}_\infty = A_\infty \otimes I_M$ and columns of A_∞ are

$$a_{k,\infty} = \frac{P_k\widehat{Q}_{k,\infty}^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^{\mathsf{T}}\widehat{Q}_{k,\infty}^{-1}\mathbb{1}_{n_k}} \tag{27}$$

where $\widehat{Q}_{k,\infty} \triangleq P_k^T\widehat{Q}_\infty P_k$ and $\widehat{Q}_\infty \triangleq \lim_{i\to\infty}\mathbb{E}\widehat{\boldsymbol{Q}}_i$. □

Following this assumption, we write an expression for $\mathbb{E}\widehat{\boldsymbol{Q}}_i$ in steady-state, so that we can approximate A_∞.

Theorem 1. *Under Assumptions 1-2, the steady state expectation for $\widehat{\boldsymbol{Q}}_i$ can be approximated as a diagonal matrix of the form:*

$$\lim_{i\to\infty} \mathbb{E}\widehat{\boldsymbol{Q}}_i \approx \frac{1}{2} diag\{\mu_1^2\sigma_{s,1}^2, \ldots, \mu_N^2\sigma_{s,N}^2\} \tag{28}$$

where $\sigma_{s,k}^2 \triangleq \mathbb{E}\|\boldsymbol{s}_{k,i}(w^o)\|^2$.

Proof. The proof is omitted due to space limitations. □

When we substitute (28) into (27), each entry of the matrix A_∞ is found to be approximated by

$$a_{\ell k,\infty} = \begin{cases} \dfrac{\theta_\ell}{\sum_{m\in\mathcal{N}_k}\theta_m} & \text{, if } \ell \in \mathcal{N}_k \\ 0 & \text{, otherwise} \end{cases} \tag{29}$$

where $\theta_\ell \triangleq 1/(\mu_\ell^2\sigma_{s,\ell}^2)$. As this result shows, the steady-state combination weights generated by the proposed algorithm matches in expectation with the Relative-Variance rule [6, 19]. Of course, this result only holds under the simplifying Assumption 4. Nevertheless, as we illustrate numerically in Sec. 5, the resulting approximation error is small in practice. Therefore, we can conclude that our proposed algorithm can match the enhanced steady-state performance of previously proposed algorithms.

3.2. Steady-State Mean-Square Performance

Our next goal is to approximate the steady-state network MSD. We follow the analysis conducted in [6, 7] and adapt it according to our case of study. We also use the low-rank approximation method described in [6] so that network MSD can be approximated in terms of the data/noise statistics, the Perron eigenvector of the combination policy, and the step sizes.

Theorem 2 (Low-Rank Approximation for the Network MSD). *Under Assumptions 1-4, the steady-state network MSD obtained by the proposed algorithm is approximately equal to*

$$\text{MSD}_{\text{av}} \approx \frac{1}{2}\text{Tr}\left[\left(\sum_{k=1}^{N}\mu_k p_k H_k^o\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2 p_k^2 R_{s,k}\right)\right] \tag{30}$$

where $H_k^o = \nabla_w^2 J_k(w^o)$, $R_{s,k} = \mathbb{E}\left[\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}_{k,i}^\mathsf{T}(w^o)\right]$ *and*

$$p_k = \frac{\theta_k \sum_{m\in\mathcal{N}_k}\theta_m}{\sum_{\ell=1}^{N}\left(\theta_\ell \sum_{m\in\mathcal{N}_\ell}\theta_m\right)} \tag{31}$$

using the notation $\theta_k = 1/(\mu_k^2\sigma_{s,k}^2)$ *, for* $k = 1, \ldots, N$.

Proof. The proof is omitted due to space limitations. □

4. SIMPLIFIED VERSION OF THE ALGORITHM

In the steady-state analysis of the originally proposed algorithm, we have observed that the expected value of the matrix $\boldsymbol{Q}_i$ converges to a diagonal matrix as i goes to infinity. Since computation of only diagonal elements would also result in the same steady-state MSD with our original approach, we consider a diagonal approximation for $\boldsymbol{Q}_i$ starting from the initial iterate. Therefore, one approximation scheme for $\boldsymbol{Q}_i$ can be expressed as

$$\begin{aligned} \boldsymbol{q}_{k,i} &= (1-\alpha_1)\boldsymbol{q}_{k,i-1} + \alpha_1\|\boldsymbol{\psi}_{k,i} - \bar{\boldsymbol{\psi}}_{k,i-1}\|^2 \\ \bar{\boldsymbol{\psi}}_{k,i} &= (1-\alpha_2)\bar{\boldsymbol{\psi}}_{k,i-1} + \alpha_2\boldsymbol{\psi}_{k,i} \end{aligned} \tag{32}$$

where $\boldsymbol{q}_{k,i}$ is the k^{th} diagonal entry of the approximation $\widehat{\boldsymbol{Q}}_i$. Using (19), each entry of the matrix $\boldsymbol{A}_i$ becomes equal to

$$\boldsymbol{a}_{\ell k,i} = \begin{cases} \dfrac{\boldsymbol{q}_{\ell,i}^{-1}}{\sum_{m\in\mathcal{N}_k}\boldsymbol{q}_{m,i}^{-1}} & \text{if } \ell \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

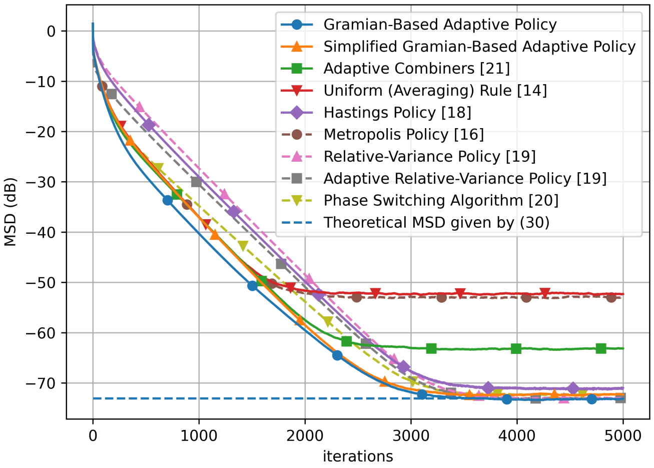

Fig. 1: Learning curves obtained with different combination policies

5. SIMULATIONS

We provide an empirical performance analysis of various combination policies with ATC algorithm on a decentralized logistic regression problem. The objective is to generate estimates for optimal w^o that minimizes all of the local cost functions:

$$J_k(w) = \mathbb{E}\left\{\ln(1 + e^{-\boldsymbol{\gamma}_k\boldsymbol{h}_k^\mathsf{T}w})\right\} + \frac{\rho_k}{2}\|w\|^2 \tag{34}$$

where $(\boldsymbol{h}_k, \boldsymbol{\gamma}_k)$ represent the streaming data received at node k. We consider the stochastic construction given by instantaneous data samples $(\boldsymbol{h}_{k,i}, \boldsymbol{\gamma}_{k,i})$. At each iteration i, we first generate labels $\boldsymbol{\gamma}_{k,i} \in \{\pm 1\}$ independently and then generate the corresponding feature vector $\boldsymbol{h}_{k,i} \sim \mathcal{N}(\boldsymbol{\gamma}_{k,i}\mu_{h,k}\mathbb{1}_M, \sigma_{h,k}^2 I_M) \in \mathbb{R}^M$. In all experiments, we set $M = 10$, $N = 20$, $\mu_k = 0.005$ and select ρ_k approximately equal to 0.5 for all nodes, where the minor adjustments serve the purpose of ensuring a common minimizer among all local cost functions. Scaling factors $\mu_{h,k}$ and $\sigma_{h,k}^2$ are selected uniformly in the interval $(0.6, 1.4)$ and log-uniformly in the interval $(10^{-2}, 1)$, respectively. All other algorithm related parameters are selected such that the observed average network MSD is minimized. For both versions of the proposed algorithm, the parameters are set to $\alpha_1 = 0.01$ and $\alpha_2 = 0.03$.

The plots in Fig.1 depict the expected value of the network SD, which is approximated by averaging the results obtained from 400 independent experiments that use the same data statistics. Furthermore, we numerically approximate H_k^o and $R_{s,k}$ matrices using the expressions given in [6, p. 321], so that we can calculate the theoretical MSD given by equation (30). We observe that the MSD obtained using either version of the Gramian-Based Adaptive Diffusion strategy agrees with the theoretical value and it also coincides with the results obtained by the Relative-Variance rule as discussed in Sect. 3.1. Proposed strategies maintain the best steady-state performance achieved by other combination rules. Moreover, both versions of the proposed algorithm outperform all others during the transient-phase, satisfying our primary objective in their development.

6. CONCLUSION

We proposed an adaptive combination rule for distributed estimation over diffusion networks. We achieved this formulation by defining an optimization problem where the goal is to obtain the least estimation error possible in each time step. We analyzed the performance of the proposed algorithm and concluded that it can maintain the enhanced MSD performance provided by previous algorithms in the literature, while improving the transient performance.

7. REFERENCES

[1] B. T. Polyak, *Introduction to Optimization*, Optimization Software, Publications Division, New York, 1987.

[2] D. P. Bertsekas, “A new class of incremental gradient methods for least squares problems,” *SIAM Journal on Optimization*, vol. 7, pp. 913–926, Apr. 1996.

[3] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[4] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, Feb. 2010.

[5] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835 – 1854, Oct. 2013.

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

[7] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[8] Z. J. Towfic and A. H. Sayed, “Stability and performance limits of adaptive primal-dual networks,” *IEEE Transactions on Signal Processing*, vol. 63, pp. 2888–2903, June 2015.

[9] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, “A saddle point algorithm for networked online convex optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, June 2015.

[10] W. Shi, Q. Ling, G. Wu, and W. Yin, “A proximal gradient algorithm for decentralized composite optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, Nov. 2015.

[11] D. Jakovetić, J. M. F. Moura, and J. Xavier, “Linear convergence rate of a class of distributed augmented lagrangian algorithms,” *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, Apr. 2015.

[12] R. Xin, S. Kar, and U. A. Khan, “Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 102–113, May 2020.

[13] R. Nassif, S. Vlaski, and A. H. Sayed, “Distributed inference over networks under subspace constraints,” *IEEE ICASSP*, pp. 5232–5236, May 2019.

[14] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” *IEEE Conference on Decision and Control*, pp. 2996–3000, Dec. 2005.

[15] D. S. Scherber and H. C. Papadopoulos, “Locally constructed algorithms for distributed computations in ad-hoc networks,” *International Symposium on Information Processing in Sensor Networks*, pp. 11–19, Apr. 2004.

[16] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *IEEE International Conference on Decision and Control*, vol. 5, pp. 4997–5002, Dec. 2003.

[17] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” *International Symposium on Information Processing in Sensor Networks*, pp. 63–70, Apr. 2005.

[18] X. Zhao and A. H. Sayed, “Performance limits for distributed estimation over LMS adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[19] X. Zhao, S-Y Tu, and A. H. Sayed, “Diffusion adaptation over networks under imperfect information exchange and non-stationary data,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.

[20] J. Fernandez-Bes, J. Arenas-Garcia, and A. H. Sayed, “Adjustment of combination weights over adaptive diffusion networks,” *IEEE ICASSP*, pp. 6409–6413, May 2014.

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

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