ON THE GENERALIZATION ABILITY OF DISTRIBUTED ONLINE LEARNERS

Zaid J. Towfic, Jianshu Chen, and Ali H. Sayed

Electrical Engineering Department
University of California, Los Angeles

ABSTRACT

We propose a fully-distributed stochastic-gradient strategy based on diffusion adaptation techniques. We show that, for strongly convex risk functions, the excess-risk at every node decays at the rate of $O(1/Ni)$, where N is the number of learners and i is the iteration index. In this way, the distributed diffusion strategy, which relies only on local interactions, is able to achieve the same convergence rate as centralized strategies that have access to all data from the nodes at every iteration. We also show that every learner is able to improve its excess-risk in comparison to the non-cooperative mode of operation where each learner would operate independently of the other learners.

Index Terms— diffusion adaptation, distributed optimization, risk function, convergence rate, mean-square-error.

1. INTRODUCTION

We study the distributed online prediction problem over a network of N learners. We assume the network is connected, meaning that any two arbitrary agents are either connected directly or by means of a path passing through other agents — see Fig. 1. Each learner k receives a sequence of data samples $\boldsymbol{x}_{k,i}$ $(i = 1, 2, \ldots)$ that arise from *the same* fixed distribution $\mathcal{X}$. In non-cooperative processing, the goal of each agent would be to learn the vector w^o that optimizes some *loss function* $Q_k(w, \boldsymbol{x}_{k,i})$ on average. For example, in order to learn the hyper-plane that best separates data from two classes $\boldsymbol{y}_{k,i} \in \{+1, -1\}$, a support-vector-machine (SVM) would optimize the expected value of the following loss function (with the expectation computed over the distribution of the data $\boldsymbol{x}_{k,i} \triangleq \{\boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i}\} \sim \mathcal{X}$ [1]:

$$Q_k^{\text{SVM}}(w,\boldsymbol{h}_{k,i},\boldsymbol{y}_{k,i}) \triangleq \frac{\rho}{2}\|w\|^2 + \max(0, 1-\boldsymbol{y}_{k,i}\boldsymbol{h}_{k,i}^{\mathsf{T}} w) \quad \text{[SVM loss]} \tag{1}$$

where ρ is a positive regularization constant. The expectation of the loss function over the distribution $\mathcal{X}$ is referred to as the *risk function* at node k [2, p.20]:

$$J_k(w) \triangleq \mathbb{E}_{\mathcal{X}}\{Q_k(w, \boldsymbol{x}_{k,i})\} \quad \text{[risk function]} \tag{2}$$

The risk measure is often interpreted as the generalization error achieved by the classifier. In cooperative processing, agents in a network work together to optimize the average global risk over all nodes in a distributed manner, where

$$J^{\text{glob}}(w) \triangleq \frac{1}{N}\sum_{k=1}^{N} J_k(w) \quad \text{[network risk function]} \tag{3}$$

This work was supported in part by NSF grants CCF-1011918 and CCF-0942936.

978-1-4673-1026-0/12/$31.00 ©2012 IEEE

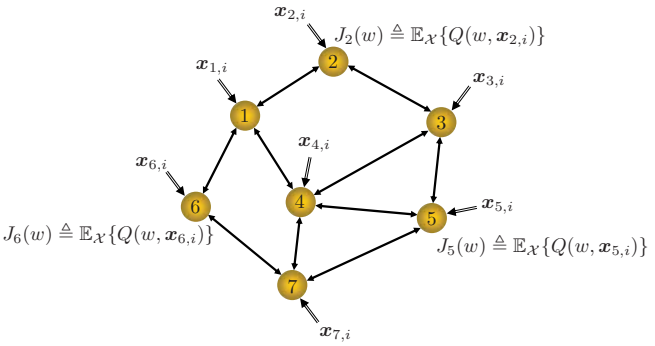

Fig. 1. A connected network in which any two nodes are connected either directly or via a path passing through other nodes.

We refer to the optimizer of (3) as w^o:

$$w^o \triangleq \arg\min_w J^{\text{glob}}(w) = \arg\min_w \sum_{k=1}^{N} J_k(w) \tag{4}$$

In many machine learning scenarios, the same loss function is used across all nodes in the network so that [3, 4]

$$J_k(w) = J(w) \quad \forall k \in \{1, 2, \ldots, N\} \tag{5}$$

In order to measure the performance of each node, we define the excess-risk (ER) at node k as:

$$\text{ER}_k(i) \triangleq \mathbb{E}_{\boldsymbol{w}}\{J(\boldsymbol{w}_{k,i}) - J(w^o)\} \tag{6}$$

where $\boldsymbol{w}_{k,i}$ is the estimator for w^o that is available at node k at time i. This estimator is a random quantity in view of the gradient noise that seeps into the algorithms described below and that are based on stochastic gradient approximation iterations.

One way to optimize (4) under (5) is for each node k to implement a stochastic gradient descent algorithm of the following form and independent of all other nodes:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_i \nabla_w Q(\boldsymbol{w}_{k,i-1}, \boldsymbol{x}_{k,i}) \quad \text{[no cooperation]} \tag{7}$$

where $\nabla_w Q(\cdot)$ denotes the gradient vector of the loss function. The gradient vector employed in (7) is an instantaneous approximation for the actual gradient vector, $\nabla J_w(\cdot)$. It was shown in [5] that for strongly convex risk functions $J(w)$, the non-cooperative scheme (7) achieves a convergence rate of the order of $O(1/i)$ under some conditions on the gradient noise and the step-size sequence μ_i. In

this way, in order to achieve an excess-risk accuracy of $O(\epsilon)$, the non-cooperative algorithm (7) would require $O(1/\epsilon)$ samples. It was further shown in [6] that no algorithm can improve upon this rate under the same conditions. This implies that if no cooperation is to take place between the nodes, then the best rate each learner would hope to achieve is on the order of $O(1/i)$.

On the other hand, assume the nodes transmit their samples to a central processor, which executes the following *centralized* algorithm:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu_i}{N}\sum_{k=1}^{N}\nabla_w Q(\boldsymbol{w}_{i-1}, \boldsymbol{x}_{k,i}) \quad \text{[centralized]} \qquad (8)$$

It can be shown that this implementation will have a convergence rate of the order of $O(1/Ni)$ for step-size sequences of the form $\mu_i = \mu/i$ and for some conditions on μ [7]. In other words, the centralized implementation (8) provides an $N-$fold increase in convergence rate relative to the non-cooperative solution (7). The main question we wish to answer in this work is whether it is possible to derive a a *fully* distributed algorithm that allows *every* node in the network to converge at the same rate as the centralized solution, i.e., $O(1/Ni)$, with only communication between neighboring nodes and for general ad-hoc networks. We extend the diffusion techniques of [8–11] and show that this objective is attainable.

To our knowledge, this is the first result regarding the use of *fully* distributed algorithms to achieve $O(1/Ni)$ convergence rate over *ad-hoc* networks. Previous works have generally required some special structure in the network, such as in [3] where the network obeys a master/worker architecture, and in [4] where communication must take place over a bounded-degree acyclic graph. The algorithms of [3,4] do not assume the risk function to be strongly convex and they achieve excess-risk convergence at the rate of $O(1/\sqrt{iN})$.

2. DISTRIBUTED FORMULATION AND ALGORITHM

Thus, consider a network of N learners. Each learner k receives a sample $\boldsymbol{x}_{k,i}$ at time i. The learners are not aware of the common underlying distribution $\mathcal{X}$ that generates the samples $\boldsymbol{x}_{k,i}$ for all nodes. Each learner wishes to optimize the risk function:

$$J(w) \triangleq \mathbb{E}_{\mathcal{X}}\{Q(w, \boldsymbol{x}_{k,i})\} \qquad (9)$$

Since the nodes are unaware of $\mathcal{X}$, and are therefore unable to evaluate the expectation in (9), each node k must optimize the risk based on its actual observations $\{\boldsymbol{x}_{k,i}\}$ in an online manner. Two useful techniques for distributed optimization are consensus strategies [12,13] and diffusion strategies [8–11]. It was argued in [14] that diffusion strategies outperform consensus strategies in terms of mean-square-error performance and convergence rate. In addition, diffusion networks are guaranteed to be stable if the individual nodes are individually stable; the same is not true for consensus networks, which can become unstable even when all nodes are stable. For this reason, we focus in this work on diffusion strategies and, in particular, extend results from [10, 11] to the context of online stochastic-gradient learners. Following the approach of [10], it is possible to derive the diffusion strategy listed in Alg. 1 for the distributed minimization of (9). The difference between Alg. 1 and the algorithm proposed in [10] is that we are now employing a diminishing step-size sequence μ_i as opposed to a constant step-size.

In order to analyze the performance guarantees of the algorithm, we assume in this article that the risk function $J(w)$ is strongly convex.

Algorithm 1 (Diffusion Adaptation)

Each node k begins with an estimate $w_{k,0}$ and step-size sequence μ_i. Each node k employs non-negative coefficients $\{a_{\ell k}\}$ such that

$$\sum_{\ell=1}^{N} a_{\ell k} = 1,\ a_{kk} > 0,\ a_{\ell k} = 0 \text{ if nodes } \ell \text{ and } k \text{ are not connected}$$

The coefficients $\{a_{\ell k}\}$ correspond to the entries of a left-stochastic combination matrix A; these coefficients are used to scale information arriving at node k from its neighbors. The neighborhood $\mathcal{N}_k$ for node k is defined as the set of nodes for which $a_{\ell k} \neq 0$. For each time instant $i \geq 1$, each node performs the following steps:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_i \widehat{\nabla}_w J(\boldsymbol{w}_{k,i-1}) & \text{[adaptation]} \quad (10) \\ \boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{\psi}_{\ell,i} & \text{[aggregation]} \quad (11) \end{cases}$$

where $\widehat{\nabla}_w J(\cdot)$ refers to a suitable instantaneous approximation for the true gradient vector $\nabla_w J(\cdot)$.

Assumption 1. *The Hessian matrix of $J(w)$ is uniformly bounded, i.e.,*

$$\lambda_{\min} I \leq \nabla^2 J(w) \leq \lambda_{\max} I \qquad (12)$$

for some constants satisfying $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$. ■

When the risk function $J(w)$ is twice continuously differentiable, then Assumption 1 is equivalent to assuming that $J(w)$ is strongly convex with a Lipschitz-continuous gradient, as is commonly assumed in literature [3, 4, 15]:

$$\|\nabla_w J(x) - \nabla_w J(y)\| \leq \lambda_{\max}\|x - y\| \qquad (13)$$

One common choice for the approximate gradient $\widehat{\nabla}_w J(w)$ at node k and time i is the following instantaneous approximation in terms of the loss function $Q(\cdot)$:

$$\widehat{\nabla}_w J(\boldsymbol{w}_{k,i-1}) \triangleq \nabla_w Q(\boldsymbol{w}_{k,i-1}, \boldsymbol{x}_{k,i}) \qquad (14)$$

Assumption 2. *We model the perturbed gradient vector as:*

$$\widehat{\nabla}_w J(\boldsymbol{w}) = \nabla_w J(\boldsymbol{w}) + \boldsymbol{v}_{k,i}(\tilde{\boldsymbol{w}}) \qquad (15)$$

where $\tilde{\boldsymbol{w}} = w^o - \boldsymbol{w}$. Moreover, conditioned on the past history of the estimators $\{\boldsymbol{w}_{k,j}\}$ for $j \leq i-1$ and all k, the gradient noise $\boldsymbol{v}_{k,i}(\boldsymbol{w})$ satisfies:

$$\mathbb{E}\{\boldsymbol{v}_{k,i}(\tilde{\boldsymbol{w}})|\mathcal{H}_{i-1}\} = 0 \qquad (16)$$

$$\mathbb{E}\{\|\boldsymbol{v}_{k,i}(\tilde{\boldsymbol{w}})\|^2\} \leq \alpha\mathbb{E}\|\tilde{\boldsymbol{w}}\|^2 + \sigma_v^2 \qquad (17)$$

for some $\alpha \geq 0$, $\sigma_v^2 \geq 0$, and where $\mathcal{H}_{i-1} \triangleq \{\boldsymbol{w}_{k,j} \colon k = 1, \ldots, N \text{ and } j \leq i-1\}$. ■

We further assume that the noise is uncorrelated across all nodes when the noise is evaluated at the optimizer w^o so that

$$\mathbb{E}\{\boldsymbol{v}_{k,i}(0)^T \boldsymbol{v}_{\ell,i}(0)\} = 0 \quad \forall\, k \neq \ell, \forall\, i \qquad (18)$$

This assumption is reasonable when the data samples are collected independently at the nodes. Now consider the excess-risk at node k:

$$\mathbb{E}_{\boldsymbol{w}}\{J(\boldsymbol{w}_{k,i}) - J(w^o)\}$$

$$\stackrel{(a)}{=} \mathbb{E}_{\boldsymbol{w}} \left\{ -\int_0^1 \nabla J(w^o - t\,\tilde{\boldsymbol{w}}_{k,i})^{\mathsf{T}} dt\; \tilde{\boldsymbol{w}}_{k,i} \right\}$$

$$\stackrel{(b)}{=} \mathbb{E}_{\boldsymbol{w}} \left\{ -\int_0^1 \nabla J(w^o)^{\mathsf{T}} dt\; \tilde{\boldsymbol{w}}_{k,i} + \tilde{\boldsymbol{w}}_{k,i}^{\mathsf{T}} \left[\int_0^1 t \int_0^1 \nabla^2 J(w^o - s\,t\,\tilde{\boldsymbol{w}}_{k,i}) ds dt \right] \tilde{\boldsymbol{w}}_{k,i} \right\}$$

$$\stackrel{(c)}{=} \mathbb{E}_{\boldsymbol{w}} \left\{ \tilde{\boldsymbol{w}}_{k,i}^{\mathsf{T}} \left[\int_0^1 t \int_0^1 \nabla^2 J(w^o - s\,t\,\tilde{\boldsymbol{w}}_{k,i}) ds dt \right] \tilde{\boldsymbol{w}}_{k,i} \right\}$$

$$\triangleq \mathbb{E}_{\boldsymbol{w}} \{ \|\tilde{\boldsymbol{w}}_{k,i}\|^2_{\boldsymbol{S}_{k,i}} \} \tag{19}$$

$$\stackrel{(d)}{\leq} \frac{\lambda_{\max}}{2} \mathbb{E}_{\boldsymbol{w}} \|\tilde{\boldsymbol{w}}_{k,i}\|^2 \tag{20}$$

where $\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}$, $\mathbb{E}_{\boldsymbol{w}}\{\cdot\}$ denotes expectation over $\boldsymbol{w}$, steps (a) and (b) are a consequence of the following property from [7, p.24]:

$$f(a+b) = f(a) + \int_0^1 \nabla f(a + t \cdot b)^{\mathsf{T}}\; dt \cdot b \tag{21}$$

And step (c) is a consequence of the fact that w^o optimizes $J(w)$ so that $\nabla_w J(w^o) = 0$. Step (d) is due to (12) in Assumption 1. Finally, the weighting matrix in (19) is defined as:

$$\boldsymbol{S}_{k,i} \triangleq \left[\int_0^1 t \int_0^1 \nabla^2 J(w^o - s\,t\,\tilde{\boldsymbol{w}}_{k,i}) ds dt \right] \tag{22}$$

Expression (19) is a useful result since it relates excess-risk analysis to mean-square analysis. The result states that the excess-risk at node k at time i is *equal* to a weighted mean-square-error with weight matrix (22). Consequently, if the distributed algorithm can be guaranteed to converge in the mean-square sense, then (20) would imply that the algorithm also converges in excess-risk. For this reason, in the next section, we proceed to show that the distributed diffusion strategy (10)-(11) converges in the mean-square sense. Subsequently, we show that the algorithm achieves the desired $O(1/Ni)$ convergence rate.

3. MEAN-SQUARE-CONVERGENCE

In this section, we show that the diffusion strategy (10)-(11) converges in excess-risk under conditions on the step-size sequence.

Theorem 1 (Asymptotic MSE Bound)**.** *Let Assumptions 1-2 hold and let the step-size sequence satisfy*

$$\sum_{i=1}^{\infty} \mu_i = \infty, \quad \lim_{i\to\infty} \mu_i = 0, \tag{23}$$

then

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \to 0. \tag{24}$$

Furthermore, let the step-size sequence be chosen as $\mu_i \triangleq \mu/i$ where $\mu > (2\lambda_{\min})^{-1}$, then the MSE at each node k satisfies

$$\lim_{i\to\infty} \frac{\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2}{i^{-1}} \leq \frac{\mu^2 \sigma_v^2}{2\mu\lambda_{\min} - 1} \tag{25}$$

for all $k = 1, \ldots, N$.

Sketch of proof. Extending the arguments of [10], and noting that the step-size sequence is time-varying as opposed to constant, it is possible to derive the following scalar recursion:

$$\|\mathcal{W}_i\|_\infty \leq \beta_i \|\mathcal{W}_{i-1}\|_\infty + \sigma_v^2 \mu_i^2 \tag{26}$$

where

$$\mathcal{W}_i \triangleq \left[\mathbb{E}\|\tilde{\boldsymbol{w}}_{1,i}\|^2, \ldots, \mathbb{E}\|\tilde{\boldsymbol{w}}_{N,i}\|^2 \right]^{\mathsf{T}} \tag{27}$$

$$\beta_i \triangleq 1 - 2\mu_i \lambda_{\min} + \mu_i^2 (\lambda_{\max}^2 + \alpha) \tag{28}$$

We can now use Lemma 3 from [7, p.45] to establish the asymptotic convergence result (24). Moreover, using the same technique as the one used in the proof of Lemma 4 from [7, p.45], it is possible to establish the rate of convergence (25). ■

Observe that (24) implies that each node converges in the mean-square-error sense. Combining this result with (20), we also conclude that each node converges in excess-risk. However, the bound (25) is still not sufficient to reveal the gain that results from cooperation; the bound only shows that the nodes are converging at the rate of $O(1/i)$, which is still the same rate as the non-cooperative scheme (7). The main cause for this weaker result is that the bound in (25) does not depend on (or exploit) the combination matrix A. In order to quantify the benefit of cooperation, we continue our discussion by assuming that the matrix A is left-stochastic *and* primitive [16, p.730]; every connected network with at least one self-loop ($a_{kk} > 0$) automatically leads to a primitive A [8].

4. EXCESS-RISK APPROXIMATION

From the previous section, we know that the diffusion algorithm guarantees a rate of *at least* $O(1/i)$ when $\mu > (2\lambda_{\min})^{-1}$ and A is left-stochastic. We show that the diffusion algorithm can approach the rate $O(1/Ni)$ achieved by the centralized algorithm (8).

4.1. Approximate Steady State Performance

The arguments presented here extend the results of [10] to the case in which the step-size sequence is time-variant as opposed to constant. We begin by introducing the network error vector $\tilde{\boldsymbol{w}}_i \triangleq \mathrm{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\}$ and the quantities:

$$\mathcal{A} \triangleq A \otimes I_M, \quad \boldsymbol{g}_i(\tilde{\boldsymbol{w}}) \triangleq \mathrm{col}\left\{\boldsymbol{v}_{1,i}(\tilde{\boldsymbol{w}}), \ldots, \boldsymbol{v}_{N,i}(\tilde{\boldsymbol{w}})\right\} \tag{29}$$

where $\otimes$ denotes the Kronecker product [17, p.243]. We also let

$$\mathcal{R}_v \triangleq \mathbb{E}\{\boldsymbol{g}_i(0)\boldsymbol{g}_i(0)^{\mathsf{T}}\} = I \otimes R_v \tag{30}$$

where the last equality is due to our assumption that the noise is uncorrelated across the nodes and the matrix R_v is defined as:

$$R_v \triangleq \mathbb{E}\{\boldsymbol{v}_{k,i}(0)\boldsymbol{v}_{k,i}(0)^{\mathsf{T}}\}, \quad k \in \{1, \ldots, N\} \tag{31}$$

The covariance matrix R_v does not depend on the node at which it is being evaluated since all nodes are assumed to sample from the same distribution $\mathcal{X}$ and have the same loss function $Q(\cdot)$. Some algebra similar to [10, 11] would show that for large enough i:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_S^2 \approx \frac{\mu_i^2}{2} \mathrm{vec}\left(\mathcal{A}^{\mathsf{T}} \mathcal{R}_v \mathcal{A}\right)^{\mathsf{T}} (I - \mathcal{F}_i)^{-1} \mathrm{vec}(S) \tag{32}$$

where $\|x\|_S^2 \triangleq x^{\mathsf{T}} S x$ and

$$\mathcal{F}_i \triangleq \mathcal{B}_i^{\mathsf{T}} \otimes \mathcal{B}_i^{\mathsf{T}} \tag{33}$$

$$\mathcal{B}_i \triangleq A^{\mathsf{T}} \otimes \left(I_M - \mu_i \nabla_w^2 J(w^o)\right) \tag{34}$$

$$S \triangleq E_{kk} \otimes S_k \tag{35}$$

$$S_k \triangleq \frac{1}{2}\nabla_w^2 J(w^o) \tag{36}$$

$$E_{kk} \triangleq e_k e_k^{\mathsf{T}} \tag{37}$$

where e_k is the k-th standard basis vector in $\mathbb{R}^N$. The weight matrix S_k corresponds to the asymptotic approximation of (22) when $\tilde{\boldsymbol{w}}_{k,i}$ is small (in the mean-square sense), which occurs for large i:

$$\boldsymbol{S}_{k,i} \approx S_k \triangleq \frac{1}{2}\nabla_w^2 J(w^o) \qquad \text{[large } i\text{]} \tag{38}$$

4.2. Distributed Performance Gain for Diffusion

In order to assess the learning rate of the algorithm, we extend the arguments of [11] and examine more closely the structure of the matrices appearing in (32). Since the Hessian at w^o is symmetric, we represent its eigenvalue decomposition as:

$$\nabla_w^2 J(w^o) = \Phi \Lambda \Phi^{\mathsf{T}} \tag{39}$$

where Φ is orthogonal and Λ is diagonal with positive entries. On the other hand, since the combination matrix A is not necessarily diagonalizable, we represent its Jordan decomposition as:

$$A = TDT^{-1} \tag{40}$$

where D is a block diagonal matrix with Jordan blocks. We can now rewrite $\mathcal{B}_i$ as

$$\mathcal{B}_i = (T^{-\mathsf{T}} \otimes \Phi)[D^{\mathsf{T}} \otimes (I - \mu_i \Lambda)](T^{\mathsf{T}} \otimes \Phi^{\mathsf{T}}) \tag{41}$$

and $\mathcal{F}_i$ as

$$\mathcal{F}_i = \mathcal{G}\left[(D \otimes (I_M - \mu_i\Lambda)) \otimes (D \otimes (I_M - \mu_i \Lambda))\right]\mathcal{G}^{-1}$$

where

$$\mathcal{G} \triangleq (T \otimes \Phi) \otimes (T \otimes \Phi) \tag{42}$$

Then, from (19) and (32):

$$\mathrm{ER}_k(i) \approx \frac{\mu_i^2}{2}\mathrm{vec}(D^{\mathsf{T}}T^{\mathsf{T}}TD \otimes \Phi^{\mathsf{T}}R_v\Phi)^{\mathsf{T}}\Omega_i^{-1}\mathrm{vec}(T^{-1}E_{kk}T^{-\mathsf{T}} \otimes \Lambda) \tag{43}$$

where

$$\Omega_i \triangleq I_{M^2N^2} - (D \otimes (I_M - \mu_i\Lambda)) \otimes (D \otimes (I_M - \mu_i\Lambda)) \tag{44}$$

We can now establish the following result.

Theorem 2. *Given a left-stochastic primitive matrix A and a step-size sequence $\mu_i \triangleq \mu/i$ where $\mu > (2\lambda_{\min})^{-1}$, the excess-risk at node k and large enough time i is approximated by*

$$\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4i}\|r\|_2^2 \tag{45}$$

where r is the right-eigenvector of the matrix A associated with eigenvalue 1 and satisfies $\mathbb{1}^{\mathsf{T}}r = 1$. Furthermore, the excess-risk is minimized when the matrix A is doubly-stochastic and primitive; in which case the excess-risk is approximated by

$$\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4Ni} \tag{46}$$

Table 1. Asymptotic performance for the non-coopertive, diffusion, and centralized strategies.

	Non-Cooperative (7)	Centralized (8)	Diffusion (10)-(11)
$\mathrm{ER}_k(i)$	$\frac{\mu \mathsf{Tr}(R_v)}{4i}$	$\frac{\mu \mathsf{Tr}(R_v)}{4Ni}$	$\frac{\mu \mathsf{Tr}(R_v)}{4Ni}$

Proof. See Appendix A. ■

One particular choice of a doubly-stochastic combination matrix A can be constructed from the following Metropolis rule [9]:

$$a_{\ell k} = \begin{cases} \min\left(\frac{1}{|\mathcal{N}_\ell|}, \frac{1}{|\mathcal{N}_k|}\right), & \ell \in \mathcal{N}_k, \ell \neq k \\ 1 - \sum_{j \in \mathcal{N}_k \setminus \{k\}} a_{jk}, & \ell = k \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

This result can be used to derive the non-cooperative asymptotic performance by letting $N = 1$ in (46). The performance of all three strategies are listed in Table 1. Observe that the asymptotic performance of the diffusion algorithm matches that of the centralized algorithm.

The above result implies that *every* node in the network will improve its estimate (in comparison to the non-cooperative solution (7)) by a factor of N. In addition, this result also implies that each node will be converging in excess-risk at the same rate as the centralized algorithm (8). Therefore, if an excess-risk on the order of $O(\epsilon)$ is desired, then only $O(1/N\epsilon)$ samples per node are necessary, as opposed to $O(1/\epsilon)$ samples for the non-cooperative solution in (7). Finally, we note that the excess-risk approximation in (46) does not explicitly depend on the Hessian of the risk function evaluated at w^o, unlike the bounds derived in (20) and (25).

5. SIMULATION

In order to demonstrate our results, we simulate the diffusion algorithm on a moderate-size regularized logistic regression problem. The loss function is defined as:

$$Q(w, \boldsymbol{h}, \boldsymbol{y}) = \frac{\rho}{2}\|w\|^2 + \log\left(1 + e^{-\boldsymbol{y}\boldsymbol{h}^{\mathsf{T}}w}\right) \tag{48}$$

This loss function can be seen as a twice-differentiable approximation to the hinge-loss that is used in the SVM formulation (1). The data used in the simulations originate from the "alpha" and "beta" datasets [18]. The datasets contain $P = 500,000$ instances each, where $y \in \{-1, +1\}$, $h \in \mathbb{R}^{M \times 1}$, and $M = 500$. The data are randomly permuted and divided evenly across $N = 8$ nodes on a randomly generated topology at every experiment. The Metropolis combination rule (47) is used for the combination weights since it leads to a doubly-stochastic A. The regularization constant ρ is taken to be $\rho = 5$ and the step-size sequence μ_i is taken to be $\mu_i \triangleq 0.9/i$. The performance metric is the excess-risk (6). Since the distribution from which the data $\{\boldsymbol{h}_i, \boldsymbol{y}_i\}$ arise is not available, we use the empirical mean to estimate the expectation over the data. That is,

$$J(w) \approx \frac{1}{P}\sum_{p=1}^{P} Q(w, h_p, y_p) \tag{49}$$

Furthermore, the optimizer w^o that is required in order to evaluate the excess-risk is taken to be

$$w^o \approx \arg\min_w \frac{1}{P}\sum_{p=1}^{P} Q(w, h_p, y_p) \tag{50}$$

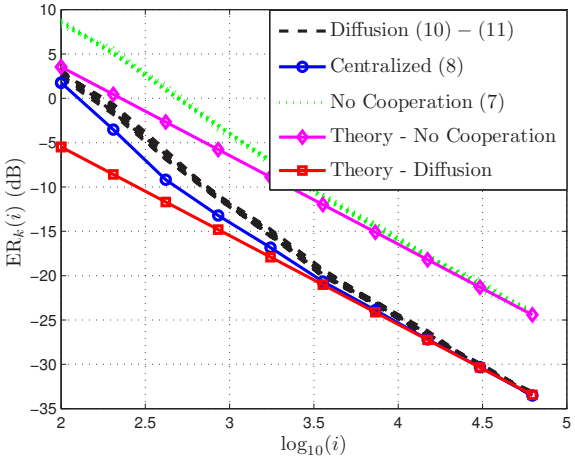

Fig. 2. Excess-risk attained by nodes that utilize the non-cooperative algorithm (7), the centralized algorithm (8), and the diffusion algorithm (10)-(11). The theoretical curves are from Table 1. Data for this simulation originate from the "alpha" dataset [18].

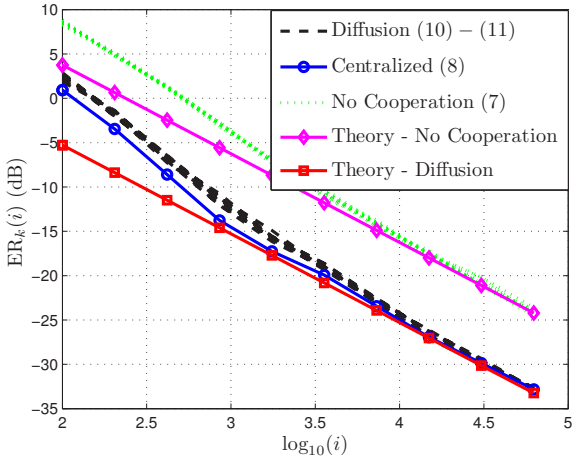

Fig. 3. Excess-risk attained by nodes that utilize the non-cooperative algorithm (7), the centralized algorithm (8), and the diffusion algorithm (10)-(11). The theoretical curves are from Table 1. Data for this simulation originate from the "beta" dataset [18].

In Fig. 2, we plot the excess-risk attained by the diffusion algorithm (10)-(11), the non-cooperative solution (7), and the centralized algorithm (8) for the "alpha" dataset. The performance of the algorithms is plotted in Fig. 3 for the "beta" dataset. The curves are averaged over 100 experiments. Observe that the diffusion algorithm and the centralized algorithm achieve the expected 9dB improvement in excess-risk when compared with the non-cooperative solution (7). The simulation curves are within 1dB of the theoretical excess-risk (46). In addition, notice that *every node* that utilizes the diffusion algorithm asymptotically achieves the same rate as the centralized algorithm. For this reason, it is beneficial for the nodes to cooperate in order to improve their performance.

6. CONCLUSION

We demonstrated the convergence in mean-square-error and excess-risk of the diffusion optimization algorithm (10)–(11) under reasonable conditions on the gradient noise and step-size sequence when all nodes from the network optimize the same loss function over data sampled from some common distribution $\mathcal{X}$. In addition, we established that the diffusion algorithm improves the asymptotic convergence rate of every node by a factor of N, where N is the number of nodes in the network. This convergence rate matches that of the centralized algorithm (8).

7. REFERENCES

[1] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th international conference on Machine learning* (ICML), Corvallis, OR, Jun. 2007, pp. 807–814.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, NY, 2000.

[3] A. Agarwal and J. Duchi, "Distributed delayed stochastic optimization," in *Proc. Neural Information Processing Systems* (NIPS), Granada, Spain, Dec. 2011, pp. 873–881.

[4] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction," in *Proc. International Conference on Machine Learning* (ICML), Bellevue, WA, Jun. 2011, pp. 713–720.

[5] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[6] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, "Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization," *Arxiv preprint arXiv:1009.0571*, Nov. 2011.

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

[8] A. H. Sayed, "Diffusion adaptation over networks," available as *arXiv:1205.4220v1*, May 2012.

[9] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[10] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," to appear in *IEEE Transactions on Signal Processing,* 2012. Also available as *arXiv:1111.0034*, Oct. 2011.

[11] X. Zhao and A. H. Sayed, "Performance limits of distributed estimation over LMS adaptive networks," to appear in *IEEE Transactions on Signal Processing,* 2012. Also available as *arXiv:11206.3728v1*, Jun. 2012.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation*, Prentice Hall, NJ, 1989.

[13] 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.

[14] S. Y. Tu and A. H. Sayed, "Diffusion networks outperform consensus networks," to appear in *Proc. IEEE Statistical Signal Processing Workshop* (SSP), Ann Arbor, MI, August 2012. Also available as *arXiv:1205.3993v1*, May 2012.

[15] F. Bach and E. Moulines, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Proc. Neural Information Processing Systems* (NIPS), Granada, Spain, Dec. 2011, pp. 451–459.

[16] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, NY, 4-th edition, 2002.

[17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, NY, 1991.

[18] Pascal Large Scale Learning Challenge, "Alpha and Beta datasets," http://largescale.ml.tu-berlin.de, Jan. 2008.

A. PROOF OF THEOREM 2

Since the stability of $\mathcal{F}_i$ is guaranteed for sufficiently small step-sizes (see App. C in [10]), we can express the inverse of Ω_i in terms of its power-series expansion:

$$\Omega_i^{-1} = \sum_{j=0}^{\infty} (D \otimes (I_M - \mu_i \Lambda))^j \otimes (D \otimes (I_M - \mu_i \Lambda))^j \quad (51)$$

Substituting (51) into (43) and simplifying yields

$$\mathrm{ER}_k(i) \approx \frac{\mu_i^2}{2} \sum_{j=0}^{\infty} \mathsf{Tr}\Big(D^{\mathsf{T}} T^{\mathsf{T}} T D D^j T^{-1} E_{kk} T^{-\mathsf{T}} D^{j\mathsf{T}} \otimes \Phi^{\mathsf{T}} R_v \Phi (I_M - \mu_i \Lambda)^j \Lambda (I_M - \mu_i \Lambda)^j \Big) \quad (52)$$

The matrix $(I_M - \mu_i \Lambda)^j \Lambda (I_M - \mu_i \Lambda)^j$ is diagonal, and we get

$$\mathsf{Tr}(\Phi^{\mathsf{T}} R_v \Phi (I_M - \mu_i \Lambda)^j \Lambda (I_M - \mu_i \Lambda)^j) = \sum_{m=1}^{M} (\Phi^{\mathsf{T}} R_v \Phi)_{mm} \lambda_m (1 - \mu_i \lambda_m)^{2j}$$

where λ_m is the m-th diagonal entry of Λ, and the notation $(B)_{ij}$ indicates the (i,j)-th element of matrix B. Additionally, since $\mathsf{Tr}(X \otimes Y) = \mathsf{Tr}(X)\mathsf{Tr}(Y)$, we have

$$\mathrm{ER}_k(i) \approx \frac{\mu_i^2}{2} \sum_{m=1}^{M} \lambda_m \cdot (\Phi^{\mathsf{T}} R_v \Phi)_{mm} \sum_{j=0}^{\infty} (1 - \mu_i \lambda_m)^{2j} \times \mathsf{Tr}\left(D^{\mathsf{T}} T^{\mathsf{T}} T D D^j T^{-1} E_{kk} T^{-\mathsf{T}} D^{j\mathsf{T}} \right) \quad (53)$$

Using the property $\mathsf{Tr}(A^{\mathsf{T}} B C D^{\mathsf{T}}) = \mathrm{vec}(A)^{\mathsf{T}} (D \otimes B) \mathrm{vec}(C)$ [17, p.252], where the operation $\mathrm{vec}(\cdot)$ stacks the columns of its matrix argument on top of each other, we get

$$\mathrm{ER}_k(i) \approx \frac{\mu_i^2}{2} \sum_{m=1}^{M} \lambda_m \cdot (\Phi^{\mathsf{T}} R_v \Phi)_{mm} \mathrm{vec}\left(D^{\mathsf{T}} T^{\mathsf{T}} T D \right)^{\mathsf{T}} \times (I_M - (1 - \mu_i \lambda_m)^2 D \otimes D)^{-1} \mathrm{vec}\left(T^{-1} E_{kk} T^{-\mathsf{T}} \right)$$

Now, for sufficiently large i, we have that $\mu_i = \mu/i$ will become sufficiently small and we approximate $(1 - \mu_i \lambda_m)^2 \approx 1 - 2\mu_i \lambda_m$. Therefore,

$$\mathrm{ER}_k(i) \approx \frac{\mu_i^2}{2} \sum_{m=1}^{M} \lambda_m \cdot (\Phi^{\mathsf{T}} R_v \Phi)_{mm} \mathrm{vec}\left(D^{\mathsf{T}} T^{\mathsf{T}} T D \right)^{\mathsf{T}} \times (I_{N^2} - (1 - 2\mu_i \lambda_m) D \otimes D)^{-1} \mathrm{vec}\left(T^{-1} E_{kk} T^{-\mathsf{T}} \right) \quad (54)$$

Using the fact that A is a left-stochastic and primitive matrix, we conclude from the Perron-Frobenius theorem [16, p.730] that D has the form

$$D = \begin{bmatrix} 1 & 0_{N-1}^{\mathsf{T}} \\ 0_{N-1} & D_{N-1} \end{bmatrix} \quad (55)$$

where 0_{N-1} is the zero vector of length $N-1$, D_{N-1} is an $N-1 \times N-1$ matrix formed of stable Jordan blocks. It follows that

$$\mu_i (I_{N^2} - (1 - 2\mu_i \lambda_m) D \otimes D)^{-1} = \quad (56)$$

$$\begin{bmatrix} \frac{1}{2\lambda_m} & 0 & 0 & 0 \\ 0 & P_{i,m} & 0 & 0 \\ 0 & 0 & P_{i,m} & 0 \\ 0 & 0 & 0 & Q_{i,m} \end{bmatrix} \quad (57)$$

where

$$P_{i,m} \triangleq \mu_i (I_N - (1 - 2\mu_i \lambda_m) D_{N-1})^{-1} \quad (58)$$

$$Q_{i,m} \triangleq \mu_i (I_{N^2} - (1 - 2\mu_i \lambda_m) D_{N-1} \otimes D_{N-1})^{-1} \quad (59)$$

Observe that both matrices $P_{i,m}$ and $Q_{i,m}$ are approximately proportional to μ_i when μ_i is sufficiently small:

$$P_{i,m} \approx \mu_i (I_N - D_{N-1})^{-1} \quad (60)$$

$$Q_{i,m} \approx \mu_i (I_{N^2} - D_{N-1} \otimes D_{N-1})^{-1} \quad (61)$$

For this reason, we approximate (57) by

$$\mu_i (I_{N^2} - (1 - 2\mu_i \lambda_m) D \otimes D)^{-1} \approx \frac{1}{2\lambda_m} E_{11} \otimes E_{11} \quad (62)$$

We can now rewrite expression (54) for the excess-risk at node k as:

$$\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4i} \mathrm{vec}(D^{\mathsf{T}} T^{\mathsf{T}} T D)^{\mathsf{T}} (E_{11} \otimes E_{11}) \mathrm{vec}(T^{-1} E_{kk} T^{-\mathsf{T}})$$

Now, noting that $\mathrm{vec}(A^{\mathsf{T}} A)^{\mathsf{T}} = \mathrm{vec}(I)^{\mathsf{T}} (A \otimes A)$, we have

$$\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4i} \mathrm{vec}(I)^{\mathsf{T}} (T D E_{11} \otimes T D E_{11}) \mathrm{vec}(T^{-1} E_{kk} T^{-\mathsf{T}})$$

Due to the structure of D in (55), we have that $D E_{11} = E_{11}$ and the expression for $\mathrm{ER}_k(i)$ simplifies to:

$$\boxed{\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4i} \mathsf{Tr}(T E_{11} T^{-1} E_{kk} T^{-\mathsf{T}} E_{11} T^{\mathsf{T}})} \quad (63)$$

We now observe that $T E_{11} T^{-1}$ is a rank-1 matrix that is spanned by the left- and right-eigenvectors of A corresponding to the eigenvalue 1. The left eigenvector is $\mathbb{1}_N$ since A is left-stochastic. Denote the right eigenvector by r and normalize the sum of its entries to unity; i.e., $r^{\mathsf{T}} \mathbb{1}_N = 1$ and $Ar = r$. Then, $T E_{11} T^{-1} = r \mathbb{1}_N^{\mathsf{T}}$. Substituting into (63) we get

$$\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4i} \mathsf{Tr}(r \mathbb{1}_N^{\mathsf{T}} E_{kk} \mathbb{1}_N r^{\mathsf{T}}) = \frac{\mu \mathsf{Tr}(R_v)}{4i} \|r\|_2^2 \quad (64)$$

Motivated by [11], consider the optimization problem that minimizes expression (64) over vectors r:

$$\min \|r\|_2^2$$
$$\text{subject to: } r^{\mathsf{T}} \mathbb{1} = 1$$

This optimization problem has the closed-form solution:

$$\boxed{r^o = \frac{1}{N} \mathbb{1}_N} \quad (65)$$

Therefore, we conclude that the excess-risk expression for doubly-stochastic combination matrices can be approximated by

$$\boxed{\mathrm{ER}_k(i) \approx \frac{\mu \mathsf{Tr}(R_v)}{4Ni}} \quad (66)$$

which is our desired result.