DISTRIBUTED INFERENCE OVER REGRESSION AND CLASSIFICATION MODELS

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

Electrical Engineering Department
University of California, Los Angeles

ABSTRACT

We study the distributed inference task over regression and classification models where the likelihood function is strongly log-concave. We show that diffusion strategies allow the KL divergence between two likelihood functions to converge to zero at the rate $1/Ni$ on average and with high probability, where N is the number of nodes in the network and i is the number of iterations. We derive asymptotic expressions for the expected regularized KL divergence and show that the diffusion strategy can outperform both non-cooperative and conventional centralized strategies, since diffusion implementations can weigh a node's contribution in proportion to its noise level.

Index Terms— distributed regression, distributed classification, diffusion adaptation, Kullback-Leibler divergence, relative entropy.

1. INTRODUCTION AND RELATED WORK

In machine learning, many problems of interest can be cast as function fitting problems where the function is parameterized by some parameter vector, w^o [1–6]. For example, Gaussian density estimation problems require the learning of the probability distribution of the incoming data with the probability distribution parameterized by a mean vector and a non-singular covariance matrix. In regression problems, a function form is chosen and the best parameters are sought that fit the input variables to a target variable. In classification problems, the dependent variable is a class label (± 1 in the case of binary classification or detection) and therefore discrete.

A common method for learning the parameter vector w^o is maximum likelihood estimation (MLE). In this framework, the likelihood function of the dependent variable given the independent variables is determined and then maximized over w. In order to assess the performance of whichever algorithm is used to carry out the maximization, a deviation measure between the optimal parameter, w^o, and the estimated parameter, w, must be chosen. One useful way to measure performance is to evaluate the "distance" between the maximum attainable likelihood (attained when the optimal parameter w^o is used) and the likelihood attained by using some other parameter, w. The Kullback-Leibler (KL) divergence (or relative conditional entropy) [7, pp. 22–23] can be used for this purpose since it measures the discrepancy between two probability distributions, such as likelihood functions. Under some reasonable conditions on the likelihood function (such as strong log-concavity), it is possible to show that traditional non-cooperative stochastic gradient algorithms allow the KL divergence to asymptotically converge to zero at the rate $\Theta(1/i)$, where i is the number of iterations (or observed instances of the independent and dependent variable pairs). This means that of the order of $\Theta(1/\epsilon)$ samples are needed in order to achieve a KL divergence of the order $\Theta(\epsilon)$ on average.

In this work, we focus on the *distributed* learning problem where different nodes in an ad-hoc network wish to estimate w^o from their observed data *without* explicitly communicating the raw data. This constraint may be due to privacy concerns or communication concerns. We propose a distributed diffusion strategy that is able to learn the desired parameter asymptotically and we derive an asymptotic expression for the evolution of the KL divergence measure under some reasonable assumptions on the likelihood function. Among other results, we show that the distributed algorithm allows the nodes to achieve a KL divergence of the order of $\Theta(\epsilon)$ with only $\Theta(1/N\epsilon)$ samples on average, as opposed to the $\Theta(1/\epsilon)$ required for the non-cooperative scheme. We further show that the diffusion strategy's rate of convergence matches the rate of the centralized solution (see (19)), and even surpasses the centralized algorithm's performance when the noise variance varies across the nodes. We note that we are including the factor of N inside the big-Theta notation because we are interested in studying how the convergence rate depends on the number of nodes in the network; similar notation is used in [8, 9]. In a similar manner to our earlier work in [6, 10], we will derive an asymptotic expression for this rate later in the manuscript—see (20).

We may mention that the results of [11] show that the combine-then-adapt (CTA) diffusion strategy of [12–14] converges almost surely when the noise variances are uniform across the nodes. Similarly, it is shown in [10] under the same assumptions made in this paper that sufficient conditions for almost sure convergence of the adapt-then-combine (ATC) diffusion strategy of [6, 12, 13] is that the step-size sequence be not absolutely summable but square-summable. A similar result appears in [15]. This implies that the step-size sequence of the form $\mu(i) \triangleq \mu/i$, which we will utilize in this work, guarantees almost sure convergence. However, the analysis in [11] does not examine the convergence rate of the algorithms, which is relevant in the current context. Also, the work in [15] focuses on the case where the combination matrix is doubly-stochastic (or doubly-stochastic in the mean), which is, as we will see later, only optimal in the case where all nodes experience similar noise power [6, 10]. We consider the case in which the noise variances vary across the nodes and explain how the cooperation strategy can be optimized in order to achieve (and exceed) the performance of [11, 15] and the centralized solution.

Notation. Throughout the manuscript, random quantities are denoted in boldface. Matrices are denoted with capital letters while vectors and scalars are represented with small-case letters.

2. PROBLEM FORMULATION AND ASSUMPTIONS

We study the distributed regression/classification problem, which is commonly encountered in generalized linear models and in inference problems over graphical models. We consider a connected network with N nodes numbered $k = 1, \ldots, N$. Each node k observes successive realizations of data $\{\boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i}\}$ over time. The data depend

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

on some unknown parameter vector, w^o (e.g., the distribution of the data depends on w^o). The variable $\boldsymbol{h}_{k,i}$ is assumed to be temporally white and independent over space; it is also assumed to be independent of all other variables. The quantity $\boldsymbol{y}_{k,i}$ depends on $\boldsymbol{h}_{k,i}$.

The objective of the network is to determine the parameter vector w^o from the observed variables $\{\boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i}, i \geq 0\}_{k=1}^{N}$ across all nodes. In principle, each node k can pursue this task on its own and employ the maximum likelihood estimator to determine w^o. When the likelihood function, $p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)$, is log-concave, the maximization of the likelihood function is not demanding since this task can be solved by computationally inexpensive iterative algorithms such as gradient ascent. When the data samples arrive in real-time and not available for batch processing, as commonly occurs in real life networks such as peer-to-peer or social networks, *stochastic* gradient ascent algorithms provide a computationally efficient solution to maximize the *expected* log-likelihood function over the distribution of the data. We will see that stochastic algorithms seek to directly minimize the Kullback-Leibler divergence.

The Kullback-Leibler (KL) divergence, a common metric used in this context, measures the "distance" between two likelihood distributions [7, pp. 22–23]:

$$D_{\text{KL}}\big(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w^o) \parallel p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)\big) \triangleq \mathbb{E}_{\mathcal{D}_k}\left\{\log\left(\frac{p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w^o)}{p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)}\right)\right\} \quad (1)$$

where $p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w^o)$ indicates the maximum attainable likelihood over the parameter space while $p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)$ indicates the likelihood attained using the parameter set w. Moreover, $\mathcal{D}_k$ represents the distribution of the data $\{\boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i}\}$. The KL divergence is a measure of the dissimilarity between two probability distributions [4, p. 277]. While the KL divergence is not a true distance metric, since it is non-symmetric and does not satisfy the sub-additivity property, it is nevertheless non-negative and satisfies $D(p \parallel q) = 0$ if, and only if, $p = q$. The last property implies that demonstrating that the KL divergence between two likelihood functions converges to zero ensures that the two likelihood functions converge to each other almost everywhere. It is possible to rearrange the terms in (1) to observe that minimizing the KL divergence at node k over w is equivalent to minimizing the following cost function:

$$J_k(w) \triangleq \mathbb{E}_{\mathcal{D}_k}\{Q(w, \boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i})\} \quad (2)$$

where

$$Q(w, \boldsymbol{h}_{k,i}, \boldsymbol{y}_{k,i}) \triangleq -\log(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)) \quad (3)$$

We observe that when the likelihood function $p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i}; w)$ is log-concave, then the local cost function $J_k(w)$ is convex. Since data are being collected at N nodes spread over a network, then a reasonable objective would be to minimize the aggregate KL divergence over the network, or equivalently,

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \quad (4)$$

Actually, minimizing the global cost in (4) amounts to minimizing the KL divergence of the joint distribution $p(\{\boldsymbol{y}_{k,i}\}|\{\boldsymbol{h}_{k,i}\}; w)$ over the network. This objective matches what we wish to achieve in a fully distributed manner. There are at least two classes of fully distributed strategies that can be used to minimize global costs of the form (4): consensus strategies [16, 17] and diffusion strategies [12–14, 18]. Diffusion strategies are particularly attractive because they enable information to diffuse more thoroughly through networks and endow networks with adaptation and learning abilities. They have also been shown to outperform consensus strategies in terms of mean-square-error performance and convergence rate [19]. For these reasons, we continue our discussions by studying diffusion solutions for (4). Following arguments from [13], the following diffusion implementation can be motivated for the distributed minimization of (4).

Algorithm 1 (Diffusion strategy)

```
Each node $k$ begins with an estimate $w_{k,0}$.
Let $\{a_{\ell k}\}$ denote nonnegative coefficients that satisfy:
    $\sum_{\ell=1}^{N} a_{\ell k} = 1, a_{kk} > 0, a_{\ell k} = 0$ if nodes $\ell$ and $k$ are disconnected   (5)
for $i = 1, 2, \ldots$ do
    $\psi_{k,i} = w_{k,i-1} - \mu(i)\widehat{\nabla}_w J_k(w_{k,i-1})$   [Adaptation]   (6)
    $w_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\psi_{\ell,i}$   [Aggregation]   (7)
end for
where $\widehat{\nabla}_w J_k(\cdot)$ is an instantaneous approximation for the true gradient
vector $\nabla_w J_k(\cdot)$, and $\mu(i) > 0$ is a step-size sequence.
```

In Alg. 1, each node performs two steps sequentially at each iteration i: first, the node uses (6) to move against a stochastic gradient of the cost function and, second, the node combines its updated estimate with that of its neighbors according to (7). The gradient vector in (6) is generally based on an instantaneous approximation for the true gradient using the current observed data $\{x_{k,i}, y_{k,i}\}$, such as:

$$\widehat{\nabla}_w J_k(w) = \nabla_w Q(w, x_{k,i}, y_{k,i}) \quad (8)$$

where $Q(\cdot, \cdot, \cdot)$ is defined in (3). For our subsequent analysis, we introduce the following reasonable assumption on $\widehat{\nabla}_w J_k(w)$ [13].

Assumption 1. *We model the perturbed gradient vector as:*

$$\widehat{\nabla}_w J_k(\boldsymbol{w}) = \nabla_w J_k(\boldsymbol{w}) + \boldsymbol{v}_{k,i}(\boldsymbol{w}) \quad (9)$$

where, conditioned on the past history of the estimators $\mathcal{H}_{i-1} \triangleq \{\boldsymbol{w}_{k,j} : k = 1, \ldots, N \text{ and } j \leq i-1\}$*, the gradient noise* $\boldsymbol{v}_{k,i}(\boldsymbol{w})$ *satisfies:*

$$\mathbb{E}\{\boldsymbol{v}_{k,i}(\boldsymbol{w})|\mathcal{H}_{i-1}\} = 0 \quad (10)$$

$$\mathbb{E}\{\|\boldsymbol{v}_{k,i}(\boldsymbol{w})\|^2|\mathcal{H}_{i-1}\} \leq \alpha\mathbb{E}\|w^o - \boldsymbol{w}\|^2 + \sigma_v^2 \quad (11)$$

for some $\alpha \geq 0$*,* $\sigma_v^2 \geq 0$*, and where* $\boldsymbol{w} \in \mathcal{H}_{i-1}$*.* ■

Due to the random nature of $\boldsymbol{v}_{k,i}(\cdot)$, the estimates $\boldsymbol{w}_{k,i}$ will be random and we will therefore use the boldface notation to refer to them. It is further assumed that the gradient noise $\boldsymbol{v}_{k,i}(\boldsymbol{w})$ is uncorrelated across the nodes, i.e.,

$$\mathbb{E}\{\boldsymbol{v}_{k,i}^{\mathsf{T}}(\boldsymbol{w}_{k,i})\boldsymbol{v}_{\ell,i}(\boldsymbol{w}_{\ell,i})\} = 0, \quad \forall\, k \neq \ell, \forall\, i \quad (12)$$

We also introduce the (steady-state) noise covariance matrix when the noise is evaluated at w^o:

$$R_{v,k} \triangleq \mathbb{E}\{\boldsymbol{v}_{k,i}(w^o)\boldsymbol{v}_{k,i}^{\mathsf{T}}(w^o)\}, \quad i \to \infty \quad (13)$$

In linear regression, for example, the dependent variable satisfies $\boldsymbol{y}_{k,i} \sim \mathcal{N}(\boldsymbol{h}_{k,i}^{\mathsf{T}} w^o, \sigma_k^2)$ and, therefore, different nodes will experience different amounts of noise during their learning task (see Example 2 in [13]). On the other hand, consider the example of logistic regression, a standard classification method.

Example 1 (Logistic regression)**.** *In logistic regression, the dependent variable $\boldsymbol{y}_{k,i}$ is binary, i.e., it assumes values from the set $\{+1,-1\}$, while the independent variable $\boldsymbol{h}_{k,i} \in \mathbb{R}^{M\times 1}$. The log-odds is assumed to obey a linear model [4, p. 117]:*

$$\log\left(\frac{p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w)}{1-p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w)}\right) = \boldsymbol{y}_{k,i}\boldsymbol{h}_{k,i}^{\mathsf{T}}w \qquad (14)$$

Solving for $p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w)$, we have that the likelihood is given by

$$p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w) = \frac{1}{1+e^{-\boldsymbol{y}_{k,i}\boldsymbol{h}_{k,i}^{\mathsf{T}}w}}$$

and the loss function $Q(w,\boldsymbol{h}_{k,i},\boldsymbol{y}_{k,i})$ in (3) *is found to be*

$$Q(w,\boldsymbol{h}_{k,i},\boldsymbol{y}_{k,i}) = \log\left(1+e^{-\boldsymbol{y}_{k,i}\boldsymbol{h}_{k,i}^{\mathsf{T}}w}\right) \qquad (15)$$

■

In both cases (linear and logistic regression), all nodes will share the same optimizer of their cost functions, which is the case of interest in this exposition. In logistic regression, all nodes share the same cost function $J(w)$ but the gradient oracle that provides $\widehat{\nabla}_k J(w)$ may offer different quality estimates to different nodes. In this case, we will see that cooperation will help nodes overcome this heterogeneity and will cause the quality of the estimate of *all* nodes to improve in comparison to naïve centralized processing (see (19) and Fig. 2 further ahead). In order to facilitate the analysis, we introduce the following assumption regarding the functions $J_k(w)$.

Assumption 2. *The risk function $J_k(w)$ is twice continuously differentiable and the Hessian matrix of $J_k(w)$ is uniformly bounded:*

$$\lambda_{\min} I \le \nabla_w^2 J_k(w) \le \lambda_{\max} I \qquad (16)$$

where $0 < \lambda_{\min} \le \lambda_{\max} < \infty$. ■

Specifically, Assumption 2 states that the cost function $J_k(w)$ is strongly convex with a bounded Hessian (the eigenvalues of $\nabla_w^2 J_k(w)$ are no less than $\lambda_{\min}$ and no greater than $\lambda_{\max}$). We can easily transform logistic regression presented in Example 1 to be strongly convex by adding a regularization term of the form $\frac{\rho}{2}\|w\|_2^2$ to the negative log-likelihood function [20, 21], or equivalently, to minimize the *regularized* KL divergence:

$$D_{\text{RKL}}\big(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w^o) \parallel p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w)\big) \triangleq$$
$$\frac{\rho}{2}\left(\|w\|_2^2 - \|w^o\|_2^2\right) + \mathbb{E}_{\mathcal{D}_k}\left\{\log\left(\frac{p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w^o)}{p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w)}\right)\right\} \qquad (17)$$

We will compare the performance of the diffusion strategy in Alg. 1 with that of the following non-cooperative algorithm:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu(i)\widehat{\nabla}_w J_k(\boldsymbol{w}_{k,i-1}) \quad \text{[non-cooperative]} \qquad (18)$$

It can be shown that the non-cooperative algorithm (18) allows the KL divergence to converge to zero at the rate $\Theta(1/i)$ [22–24]. On the other hand, when the nodes are allowed to transmit their data to a central node at every iteration, the following full gradient algorithm may be executed at the central node:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu(i)}{N}\sum_{k=1}^{N}\widehat{\nabla}_w J_k(\boldsymbol{w}_{i-1}) \qquad \text{[centralized]} \qquad (19)$$

It is possible to show that the centralized algorithm (19) can converge at the rate $\Theta(1/Ni)$ [6,10,25]. This conclusion suggests that there is an N-fold improvement that can be attained by cooperation. In order to show similar results for the distributed algorithm listed in Alg. 1, we require the following assumption on the network topology.

Assumption 3 (Connected network)**.** *There exists a path from every node to every other node in the network.* ■

This assumption, together with (5), guarantees that the matrix A is primitive [18]. In the next section, we study the convergence rate (to zero) of the regularized KL divergence defined in (17) when the diffusion strategy is employed, and show that the diffusion strategy can achieve the performance of the centralized solution (19) and even surpass it when the $N \times N$ combination matrix $A = [a_{\ell k}]$ is chosen properly. We note that condition (5) ensures that A is a left-stochastic matrix, i.e., $A^T\mathbb{1} = \mathbb{1}$, where $\mathbb{1}$ is the all-ones vector.

3. ASYMPTOTIC BEHAVIOR OF THE KL DIVERGENCE

In this section, we present our main result. We let the diffusion strategy (6)–(7) utilize a step-size sequence of the form $\mu(i) \triangleq \mu/i$ where the initial step-size μ is positive. Using Assumptions 1-2, we can now establish the following result.

Theorem 1 (Asymptotic behavior of KL divergence)**.** *The regularized expected KL divergence obeys*

$$\mathbb{E}_{\boldsymbol{w}_{k,i-1}}\{D_{\mathit{RKL}}\big(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w^o) \parallel p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};\boldsymbol{w}_{k,i-1})\big)\}$$
$$\sim \frac{1}{2}p^{\mathsf{T}}L_i p + \Theta(i^{-2\lambda_{\min}\mu}) \qquad (20)$$

as $i \to \infty$, where L_i is an $N \times N$ diagonal matrix with

$$\{L_i\}_{k,k} = \sum_{m=1}^{M}\lambda_m\mu^2\alpha_m(i)(\Phi^{\mathsf{T}}R_{v,k}\Phi)_{mm} \qquad (21)$$

and $\alpha_m(i)$ is defined as

$$\alpha_m(i) \triangleq \begin{cases} \frac{i^{-1}}{2\lambda_m\mu-1}, & 2\lambda_m\mu > 1 \\ \frac{\log(i)}{i}, & 2\lambda_m\mu = 1 \\ \frac{{}_3F_2(1,1,1;2-\lambda_m\mu,2-\lambda_m\mu;1)}{\Gamma(2-\lambda_m\mu)^2}\cdot i^{-2\lambda_m\mu}, & 2\lambda_m\mu < 1 \end{cases} \qquad (22)$$

where $\Gamma(\cdot)$ and ${}_3F_2(a_1,a_2,a_3;b_1,b_2;z)$ are the Gamma and generalized hypergeometric functions [26, pp. 892,1010], respectively, p is the right eigenvector of the combination matrix A associated with eigenvalue 1*; and is normalized so that $\mathbb{1}^{\mathsf{T}}p = 1$, $\nabla^2 J(w^o)$ has the eigenvalue decomposition $\nabla^2 J(w^o) = \Phi\Lambda\Phi^{\mathsf{T}}$ where $\Lambda = \text{diag}\{\lambda_1,\ldots,\lambda_M\}$ is positive-definite and Φ orthogonal, and $R_{v,k}$ is defined in* (13)*. Moreover, $\lambda_{\min}$ is defined as the smallest eigenvalue of $\nabla^2 J(w^o)$: $\lambda_{\min} \triangleq \min\{\lambda_1,\ldots,\lambda_M\}$.*

Proof. Omitted due to space limitation. ■

In Theorem 1, the notation $f(i) = \Theta(g(i))$ means that there exists a pair of constants c_1, c_2 and positive integer j such that $c_1 g(i) \le f(i) \le c_2 g(i)$ for $i \ge j$; in other words, $f(i)$ asymptotically decays/grows at the same rate as $g(i)$ (up to a constant). The key point that we wish to convey from Theorem 1 is that the fastest rate of convergence is achieved when $2\lambda_{\min}\mu > 1$. The asymptotic regularized KL divergence then becomes $(2i)^{-1}p^{\mathsf{T}}Lp$, where L is the diagonal $N \times N$ matrix with entries

$$\{L\}_{k,k} = \sum_{m=1}^{M}\frac{\lambda_m\mu^2}{2\lambda_m\mu-1}(\Phi^{\mathsf{T}}R_{v,k}\Phi)_{mm} \qquad (23)$$

The convergence rate's dependence on N is still encoded in the right-eigenvector p of the combination matrix A. In order to optimize over p, we notice that the following optimization problem is convex:

$$\min_{p} \quad p^{\mathsf{T}}Lp$$
$$\text{subject to:} \quad p_k > 0, \;\; \mathbb{1}^{\mathsf{T}}p = 1$$

where the $\{p_k\}$ denote the individual entries of p. The solution is:

$$p^o = \frac{L^{-1}\mathbb{1}}{\mathbb{1}^\mathsf{T} L^{-1}\mathbb{1}} \tag{24}$$

It is straightforward to verify that $\mathbb{1}^\mathsf{T} p^o = 1$ and that since the diagonal elements of L^{-1} are positive, then $p^o > 0$. To examine the effectiveness of this choice for p, consider the case where $2\lambda_{\min}\mu \gg 1$, in which case the matrix L becomes

$$L \approx \frac{\mu}{2}\text{diag}\{\text{Tr}(R_{v,1}),\ldots,\text{Tr}(R_{v,N})\} \quad [2\lambda_{\min}\mu \gg 1] \tag{25}$$

so (17) is asymptotically approximated by (on average),

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{w}_{k,i-1}}\{D_{\text{RKL}}\big(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w^o) \parallel p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};\boldsymbol{w}_{k,i-1})\big)\}\\ &\qquad\approx \frac{\mu}{4i}\cdot\frac{1}{\sum_{k=1}^N \text{Tr}(R_{v,k})^{-1}}, \qquad [2\lambda_{\min}\mu \gg 1]\end{aligned} \tag{26}$$

Since the harmonic mean is bounded by the arithmetic mean, we have:

$$\frac{\mu}{4i}\cdot\frac{1}{\sum_{k=1}^N \text{Tr}(R_{v,k})^{-1}} \le \frac{\mu}{4i}\cdot\frac{1}{N^2}\sum_{k=1}^N \text{Tr}(R_{v,k}) \tag{27}$$

and the inequality is strict when the noise variances are not uniform across the nodes. Actually, it can be shown using Assumption 1 that the right-hand-side of (27) is the performance attained by the centralized algorithm (19). This implies that the diffusion algorithm will asymptotically achieve the same performance as (19) when the noise variances across the nodes are the same. In addition, when the noise variances are *not* uniform, then the diffusion algorithm actually has *better* performance than (19). This is unsurprising since (19) weighs the gradients from the nodes uniformly. The centralized algorithm can be made to achieve better performance if it is modified to weigh the gradients of the different nodes according to their noise level.

The only remaining task is to construct a combination matrix A in a distributed manner, so that it satisfies $Ap^o = p^o$. This can be accomplished by using the Hastings weights [25, 27] to generate a left-stochastic combination matrix A:

$$a_{\ell k} = \begin{cases}\frac{1}{p_k^o}\min\left(\frac{p_\ell^o}{|\mathcal{N}_\ell|},\frac{p_k^o}{|\mathcal{N}_k|}\right), & \ell \in \mathcal{N}_k, \ell \neq k\\ 1-\sum_{j\in\mathcal{N}_k\setminus\{k\}} a_{jk}, & \ell = k\\ 0, & \text{otherwise}\end{cases} \tag{28}$$

where p_k^o indicates the k-th element of the vector p^o and $\mathcal{N}_k$ indicates the neighborhood of node k including the node itself: $\mathcal{N}_k = \{\ell : a_{\ell k} \neq 0\}$. Notice that the weights (28) can be computed in a decentralized manner using only information available from each node's neighborhood. It is finally possible to utilize Markov's inequality [28, p. 151] to transform the result from Theorem 1 to a high probability statement regarding the KL divergence itself, not just its expectation. That is, with probability at least $1-\delta$, we have $D_{\text{RKL}}\big(p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};w^o) \parallel p(\boldsymbol{y}_{k,i}|\boldsymbol{h}_{k,i};\boldsymbol{w}_{k,i-1})\big) \le \frac{\mu^2}{2\delta}p^\mathsf{T} L_i p$.

4. SIMULATION

In order to illustrate our results, we simulate the learning ability of a distributed network of classifiers based on Example 1. We generate an adhoc network of $N = 8$ nodes where each node samples $M = 2$ dimensional feature vectors from a Gaussian mixture with two components and probability density function $\boldsymbol{h}_{k,i} \sim \frac{1}{2}\mathcal{N}(2\mathbb{1}, I_M) + \frac{1}{2}\mathcal{N}(-2\mathbb{1}, I_M)$, where $\mathcal{N}(\gamma,\Sigma)$ denotes the multivariate Gaussian density function with mean vector γ and covariance matrix Σ. The labels $\boldsymbol{y}_{k,i}$ are generated according to (15) where w is chosen arbitrary. Each node performs the diffusion algorithm listed in Alg. 1 by computing the gradient according to the instantaneous approximation of the strongly-convex regularized KL divergence (17) with added Gaussian random noise. The noise profile across the nodes is illustrated in Fig. 1. Notice that two nodes (nodes 1 and 8) are experiencing a large amount of noise in comparison to the other nodes in the network—this is to illustrate the advantage over naïve averaging of the gradients that is performed in (19). The combination weights for the diffusion algorithm are chosen according to the Hastings rule (28) and the p^o is found by (24) where L is approximated by (25). The regularization constant ρ is chosen to be 5 in the simulation since it will determine $\lambda_{\min}$ and in order to ensure that our approximation in (26) is valid, we choose μ to be 10. The optimizer is found by optimizing the sum of the regularized KL divergences and using empirical average for the expectation.

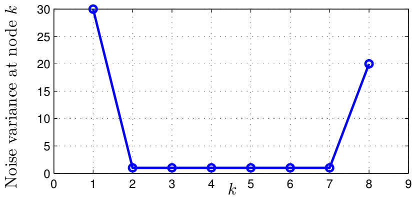

Fig. 1. Plot of noise variances across the 8 nodes in the simulation.

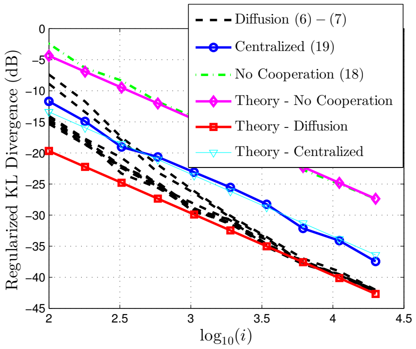

Fig. 2. Regularized KL divergence attained by nodes that utilize the non-cooperative algorithm (18), centralized algorithm (19), and diffusion algorithm (6)-(7). Theoretical curves are from (26)-(27). Curves are averaged over 100 experiments.

Figure 2 shows the resulting curves. We plot the performance of the non-cooperative algorithm along with its theoretical performance (both averaged over the nodes). In addition, we show the performance of the *centralized* algorithm (19) along with its theoretical performance obtained from the right-hand-side of (27). We see that the diffusion algorithm outperforms the non-cooperative algorithm and the centralized algorithm listed in (19). We further observe that the two nodes with the large noise variance begin with a higher regularized KL divergence but the diffusion of information throughout the network allows them to converge at the same rate as the other nodes in the network (and still faster than the rate that the nodes with low noise variance achieved by themselves). Had the diffusion algorithm utilized doubly-stochastic weights such as in [6, 11], its performance would only asymptotically match that of the centralized algorithm (19) and the right-hand-side of (27).

We conclude that the use of the optimized weights determined by (28), (24), and (25) yields an improvement in convergence rate over naïve averaging and that the diffusion strategy will asymptotically allow *all* nodes in the adhoc network to converge at this fast rate strictly through local interactions.

5. REFERENCES

[1] C. Cortes and V. N. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[2] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, Apr. 2000.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, Jan. 1977.

[4] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, MA, 4th edition, 2008.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, NY, 2nd edition, 1995.

[6] Z. J. Towfic, J. Chen, and A. H. Sayed, “On the generalization ability of distributed online learners,” in *Proc. IEEE International Workshop on Machine Learning for Signal Processing* (MLSP), Santander, Spain, Sep. 2012, pp. 1–6.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, NJ, 1991.

[8] 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.

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

[10] Z. J. Towfic, J. Chen, and A. H. Sayed, “Excess-risk analysis of distributed stochastic learners,” *submitted for publication.* Also available as *arXiv:1302.1157*, Feb. 2013.

[11] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[12] 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.

[13] 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.

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

[15] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz, “Performance analysis of a distributed Robbins-Monro algorithm for sensor networks,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 1030–1034.

[16] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[17] 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.

[18] A. H. Sayed, “Diffusion adaptation over networks,” in *E-Reference Signal Processing,* R. Chellapa and S. Theodoridis, editors, Elsevier, 2013. Also available as *arXiv:1205.4220v1*, May 2012.

[19] S.-Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, May 2012.

[20] J. Zhu and E. P. Xing, “Maximum entropy discrimination markov networks,” *The Journal of Machine Learning Research*, vol. 10, pp. 2531–2569, Nov. 2009.

[21] G. Lebanon and J. Lafferty, “Boosting and maximum likelihood for exponential models,” in *Proc. Neural Information Processing Systems* (NIPS), Vancouver, British Columbia, Canada, Dec. 2001, pp. 447–454.

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

[23] 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.

[24] 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.

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

[26] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, New York, 2007.

[27] W. K. Hastings, “Monte Carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

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