DIFFUSION STOCHASTIC OPTIMIZATION WITH NON-SMOOTH REGULARIZERS

Stefan Vlaski, Lieven Vandenberghe and Ali H. Sayed

Department of Electrical Engineering, University of California, Los Angeles

ABSTRACT

We develop an effective distributed strategy for seeking the Pareto solution of an aggregate cost consisting of regularized risks. The focus is on stochastic optimization problems where each risk function is expressed as the expectation of some loss function and the probability distribution of the data is unknown. We assume each risk function is regularized and allow the regularizer to be non-smooth. Under conditions that are weaker than assumed earlier in the literature and, hence, applicable to a broader class of adaptation and learning problems, we show how the regularizers can be smoothed and how the Pareto solution can be sought by appealing to a multi-agent diffusion strategy. The formulation is general enough and includes, for example, a multi-agent proximal strategy as a special case.

Index Terms— Distributed optimization, diffusion strategy, smoothing, proximal operator, non-smooth regularizer, proximal diffusion, regularized diffusion.

1. INTRODUCTION AND RELATED WORK

We consider a strongly-connected network consisting of N agents. For any two agents k and ℓ, we attach a pair of nonnegative coefficients $\{a_{\ell k}, a_{k\ell}\}$ to the edge linking them. The scalar $a_{\ell k}$ is used to scale data moving from agent ℓ to k; likewise, for $a_{k\ell}$. Strong-connectivity means that it is always possible to find a path, in either direction, with nonzero scaling weights linking any two agents. In addition, at least one agent k in the network possesses a self-loop with $a_{kk} > 0$. Let $\mathcal{N}_k$ denote the set of neighbors of agent k. The coefficients $\{a_{\ell k}\}$ are convex combination weights that satisfy

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

If we introduce the combination matrix $A = [a_{\ell k}]$, it then follows from (1) and the strong-connectivity property that A is a left-stochastic primitive matrix. In view of the Perron-Frobenius Theorem [1–3], this ensures that A has a single eigenvalue at one while all other eigenvalues are inside the unit circle. If we let p denote the right-eigenvector of A that is associated with the eigenvalue at one, and if we normalize the entries of p to add up to one, then it also holds that all entries of p, denoted by $\{p_k\}$, are strictly positive, i.e.,

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{2}$$

We associate with each agent k a risk function $J_k(w) : \mathbb{R}^M \to \mathbb{R}$, assumed differentiable. In most adaptation and learning problems, risk functions are expressed as the expectation of loss functions. Hence, we assume that $J_k(w) = \mathbb{E} Q_k(w; \boldsymbol{x})$ for some loss function $Q_k(\cdot)$ and where $\boldsymbol{x}$ denotes random data. The expectation is computed over the distribution of this data (note that, in our notation, we use boldface letters for random quantities and normal letters for deterministic quantities). We also associate with agent k a regularization term, $R_k(w) : \mathbb{R}^M \to \mathbb{R}$, which is a known deterministic function although possibly non-differentiable. Regularization factors of this form can, for example, help induce sparsity properties (such as using ℓ_1 or elastic-net regularizers) [4–6].

The objective we are interested in is to devise a fully distributed strategy to seek the minimizer of the following weighted aggregate cost, denoted by w^o:

$$w^o \triangleq \arg\min_w \sum_{k=1}^{N} p_k \left\{ J_k(w) + R_k(w) \right\} \tag{3}$$

The weights $\{p_k\}$ indicate that the resulting minimizer w^o can be interpreted as a Pareto solution for the collection of regularized risks $\{J_k(w) + R_k(w)\}$ [3,7]. We are particularly interested in determining this Pareto solution in the *stochastic* setting when the distribution of the data $\boldsymbol{x}$ is generally unknown. This means that the risks $J_k(w)$, or their gradient vectors, are also unknown. As such, *approximate* gradient vectors will need to be employed. A common construction in stochastic approximation theory is to employ the following choice at each iteration i:

$$\widehat{\nabla_w J_k}(w) \;=\; \nabla_w Q_k(w; \boldsymbol{x}_i) \tag{4}$$

where $\boldsymbol{x}_i$ represents the data that is available (observed) at time i. The difference between the true gradient vector and its approximation is called *gradient noise*. This noise will seep into the operation of the distributed algorithm and one main challenge is to show that, despite its presence, the proposed solution is able to approach w^o asymptotically. A second challenge we face in constructing an effective distributed solution is the non-smoothness (non-differentiability) of the regularizers. Motivated by a technique proposed in [8] in the context of *single* agent optimization, we will address this difficulty in the multi-agent case by introducing a smoothed version of the regularizers and showing that the solution w^o can still be recovered under this substitution as the size of the smoothing parameter is reduced. We adopt a general formulation that will be shown to include a distributed proximal solution as a special case.

There are several useful works in the literature that study optimization problems with non-smooth regularizers. For example, the work [9] relies on the use of *sub-gradient* iterations but requires that the sub-gradients of the regularized risks, $J_k(w) + R_k(w)$, should be uniformly bounded. Unfortunately, this condition is not satisfied in many important cases of interest, for example, even when $J_k(w)$ is simply quadratic in w or when the $R_k(w)$ are indicator functions used to encode constraints. Variations for specific choices of $J_k(\cdot)$ are examined in [10–13] where only the subgradients of

This work was supported in part by NSF grants CIF-1524250 and ECCS-1407712 and DARPA project N66001-14-2-4029. Emails: {svlaski,vandenbe,sayed}@ucla.edu

$R_k(\cdot)$ are required to be bounded. For the case when the $R_k(w)$ are chosen as indicator functions in constrained problem formulations, a distributed diffusion strategy based on the use of suitable penalty functions is proposed in [14].

Some more recent studies pursue distributed solutions by relying on the use of proximal iterations (as opposed to sub-gradient iterations); an accessible survey on the proximal operator and its properties appears in [15]. For example, for purely deterministic costs, distributed proximal strategies are developed in [16, 17]. In the *single-agent* case, the behavior of the forward-backward algorithm under stochastic perturbations is investigated in [18]. Distributed variations for mean-square error costs with bounded regularizer subgradients are proposed in [19, 20] for single-task problems and in [21] for multi-task environments. A strategy for general stochastic costs with *small*, Lipschitz continuous regularizers is studied in [22].

Most of these prior works involve requirements that limit their application to particular scenarios, whether in terms of requiring bounded sub-gradients, or focusing on quadratic costs, or requiring small regularizers. The purpose of this work is to propose a general distributed strategy and a line of analysis that is applicable to a wide class of stochastic costs under non-differentiable regularizers. The first step in the solution involves replacing each $R_k(w)$ by a differentiable approximation, $R_k^\delta(w)$, parametrized by $\delta > 0$, such that

$$R_k^\delta(w) \leq R_k(w) \quad \text{and} \quad \lim_{\delta \to 0} R_k^\delta(w) = R_k(w). \tag{5}$$

The accuracy of the approximation is controlled through δ. Subsequently, we approximate problem (3) by

$$w_\delta^o \triangleq \arg\min_w \sum_{k=1}^N p_k \left\{ J_k(w) + R_k^\delta(w) \right\} \tag{6}$$

In the next sections we explain how to construct the smooth approximation, $R_k^\delta(w)$, by appealing to conjugate functions and will show that the distance $\|w^o - w_\delta^o\|$ can be made arbitrarily small as $\delta \to 0$. We then present an algorithm to solve for the minimizer of (6) in a distributed manner. The analysis will rely on the following common assumptions [3, 23, 24].

Assumption 1 (Lipschitz gradients)**.** *For each k, the gradient $\nabla_w J_k(\cdot)$ is Lipschitz, namely, for any $x, y \in \mathbb{R}^M$:*

$$\|\nabla_w J_k(x) - \nabla_w J_k(y)\| \leq \lambda_U \|x - y\| \tag{7}$$

Assumption 2 (Strong Convexity)**.** *The weighted aggregate of the differentiable risks is strongly-convex, namely, for any $x, y \in \mathbb{R}^M$:*

$$(x-y)^\mathsf{T} \cdot \sum_{k=1}^N p_k \left(\nabla_w J_k(x) - \nabla_w J_k(y) \right) \geq \lambda_L \|x - y\|^2 \tag{8}$$

Assumption 3 (Regularizers)**.** *For each k, $R_k(\cdot)$ is closed convex.*

2. ALGORITHM FORMULATION

2.1. Construction of Smooth Approximation

Smoothing non-differentiable costs is a popular technique in the optimization literature [8, 25]. Nevertheless, this method has been mainly applied to the solution of *deterministic* optimization problems by *single* stand-alone agents. In this work, we are pursuing an extension in two non-trivial directions. First, we consider networked agents (rather than a single agent) working together to solve the aggregate optimization problem (3) or (6) and, second, the risk functions involved are now *stochastic* costs defined as the expectations of certain loss functions. In this case, the probability distribution of the data is unknown and, therefore, the risks themselves are not known but can only be approximated. The challenge is to devise a distributed strategy that is able to converge to the desired Pareto solution despite these difficulties.

To begin with, we explain how smoothing of the regularizers is performed. Thus, recall that the conjugate function, denoted by $R_k^\star(w)$, of a regularizer $R_k(w)$ is defined as

$$R_k^\star(w) \triangleq \sup_{u \in \mathbf{dom}\, R_k} \left\{ w^\mathsf{T} u - R_k(u) \right\}. \tag{9}$$

A useful property of conjugate functions is that $R_k^\star(w)$ is always closed convex regardless of whether $R_k(w)$ is convex or not [26,27].

Definition 1 (Distance function)**.** *A distance function $\mathrm{dist}(\cdot)$ for a closed convex set C is a continuous, strongly-convex function with $C \subseteq \mathrm{dom}\, \mathrm{dist}(\cdot)$. We normalize the function so that*

$$\min_{w \in C} \mathrm{dist}(w) = 0, \qquad \mathrm{dist}(w) \geq \frac{1}{2}\|w - w_{\mathrm{cent}}\|^2 \tag{10}$$

for some w_{cent}, which means that the strong-convexity constant is set to one. □

Definition 2 (Smooth approximation)**.** *We choose a distance function over $C = \mathrm{dom}\, R_k^\star(w)$ and define the smooth approximation of $R_k(\cdot)$ as:*

$$\begin{aligned} R_k^\delta(w) &\triangleq \max_{u \in \mathbf{dom}\, R_k^\star} \left\{ w^\mathsf{T} u - R_k^\star(u) - \delta \cdot \mathrm{dist}(u) \right\} \\ &= (R_k^\star + \delta \cdot \mathrm{dist})^\star (w) \end{aligned} \tag{11}$$

□

Thus, observe that the smooth approximation for $R_k(w)$, which we are denoting by $R_k^\delta(w)$, is obtained by first perturbing the conjugate function $R_k^\star(u)$ by $\delta \cdot \mathrm{dist}(u)$ and then conjugating the result again. The perturbation makes the sum $R_k^\star(u) + \delta \cdot \mathrm{dist}(u)$ a strongly-convex function. The motivation behind this construction is that the conjugate of a strongly-convex function is *differentiable* everywhere and, therefore, $R_k^\delta(w)$ is differentiable everywhere. This intuition is formalized in the following statement [8].

Theorem 1 (Smooth approximation)**.** *Any $R_k^\delta(w)$ constructed according to (11) satisfies (5) and is differentiable with gradient vector given by*

$$\nabla_w R_k^\delta(w) = \arg\max_{u \in \mathbf{dom}\, R_k^\star} \left\{ w^\mathsf{T} u - R_k^\star(u) - \delta \cdot \mathrm{dist}(u) \right\}. \tag{12}$$

Furthermore, the gradient is co-coercive, i.e., it satisfies for any x, y:

$$(x-y)^\mathsf{T} \left(\nabla_w R_k^\delta(x) - \nabla_w R_k^\delta(y) \right) \geq \delta \|\nabla_w R_k^\delta(x) - \nabla_w R_k^\delta(y)\|^2 \tag{13}$$

□

From this result, we can establish that:

$$\lim_{\delta \to 0} \|w^o - w_\delta^o\| = 0 \tag{14}$$

Obviously, the feasibility of stochastic-gradient algorithms for the minimization of (6) hinges on the assumption that (12) can be evaluated in closed form or at least easily. Fortunately, this is the case for a large class of regularizers of interest – see [28] for the case $\mathrm{dist}(\cdot) = \frac{1}{2}\|\cdot\|^2$ and [8] for other distance choices.

2.2. Regularized Diffusion Strategy

Now that we have established a method for constructing a differentiable approximation for each regularizer, we can solve for the minimizer of (6) by resorting to the following (adapt-then-combine form of the) diffusion strategy [3, 23, 24]:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) - \mu\nabla_w R_k^\delta(\boldsymbol{w}_{k,i-1}) \quad (15)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^N a_{\ell k}\boldsymbol{\psi}_{\ell,i} \quad (16)$$

where $\mu > 0$ is a small step-size parameter. In this implementation, each agent k first performs the stochastic-gradient update (15), starting from its existing iterate value $\boldsymbol{w}_{k,i-1}$, and obtains an intermediate iterate $\boldsymbol{\psi}_{k,i}$. Subsequently, agent k consults with its neighbors and combines their intermediate iterates into $\boldsymbol{w}_{k,i}$ according to (16). Motivated by the construction in [14], we can refine (15)–(16) further as follows. We introduce an auxiliary variable $\boldsymbol{\phi}_{k,i}$ and perform (15) in two successive steps by writing:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) \quad (17)$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i} - \mu\nabla_w R_k^\delta(\boldsymbol{w}_{k,i-1}) \quad (18)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^N a_{\ell k}\boldsymbol{\psi}_{\ell,i} \quad (19)$$

We can now appeal to an incremental-type argument [29, 30] by noting that it is reasonable to expect $\boldsymbol{\phi}_{k,i}$ to be an improved estimate for w_δ^o compared to $\boldsymbol{w}_{k,i-1}$. Therefore, we replace $\boldsymbol{w}_{k,i-1}$ in (18) by $\boldsymbol{\phi}_{k,i}$ and arrive at the following regularized diffusion implementation.

Algorithm: (Regularized Diffusion Strategy)

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) \quad (20)$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i} - \mu\nabla_w R_k^\delta(\boldsymbol{\phi}_{k,i}) \quad (21)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^N a_{\ell k}\boldsymbol{\psi}_{\ell,i} \quad (22)$$

Example 1 (Proximal diffusion learning)**.** Choosing $\mathrm{dist}(w) = \frac{1}{2}\|w\|^2$ turns the smooth approximation (11) into

$$R_k^\delta(w) = \left(R_k^\star(w) + \frac{\delta}{2}\|w\|^2\right)^\star \quad (23)$$

which is the well-known Moreau envelope [15, 28, 31]. It can be rewritten equivalently as

$$R_k^\delta(w) \triangleq \min_u \left(R_k(u) + \frac{1}{2\delta}\|w-u\|^2\right) \quad (24)$$

where the minimizing argument is identified as the proximal operator:

$$\mathrm{prox}_{\delta R_k}(w) = \arg\min_u \left(R_k(u) + \frac{1}{2\delta}\|w-u\|^2\right). \quad (25)$$

For many costs $R_k(w)$, the proximal operator can be evaluated in closed form. The gradient of the Moreau envelope (23) can be verified to be given by

$$\nabla_w R_k^\delta(w) = \frac{1}{\delta}\left(w - \mathrm{prox}_{\delta R_k}(w)\right). \quad (26)$$

In this way, recursions (20)–(22) reduce to:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) \quad (27)$$

$$\boldsymbol{\psi}_{k,i} = \left(1 - \frac{\mu}{\delta}\right)\boldsymbol{\phi}_{k,i} + \frac{\mu}{\delta}\mathrm{prox}_{\delta R_k}(\boldsymbol{\phi}_{k,i}) \quad (28)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^N a_{\ell k}\boldsymbol{\phi}_{\ell,i} \quad (29)$$

which is a damped variation of the proximal diffusion algorithm proposed in [22] under the stronger assumption of small Lipschitz continuous regularizers (setting $\mu = \delta$ in (28) leads to the algorithm in [22]). □

3. CONVERGENCE ANALYSIS

3.1. Centralized Recursion

We now examine the convergence properties of the diffusion strategy (20)–(22). To do so, it is useful to introduce the following centralized recursion to serve as a frame of reference:

$$\bar{w}_i = \bar{w}_{i-1} - \mu\sum_{k=1}^N p_k\left\{\nabla_w J_k(\bar{w}_{i-1}) + \nabla_w R_k^\delta(\bar{w}_{i-1})\right\} \quad (30)$$

This recursion amounts to a gradient-descent iteration applied to the smoothed aggregate cost in (6) under the assumption that the risk functions are known. For convenience of presentation, we introduce the central operator $T_c(x) : \mathbb{R}^M \to \mathbb{R}^M$ defined as follows:

$$T_c(x) \triangleq x - \mu\sum_{k=1}^N p_k\left\{\nabla_w J_k(x) + \nabla_w R_k^\delta(x)\right\} \quad (31)$$

so that the reference recursion (30) becomes $\bar{w}_i = T_c(\bar{w}_{i-1})$. The proof of the following result is omitted for brevity.

Lemma 1 (Contraction mapping)**.** *For sufficiently small μ, the mapping $T_c(\cdot)$ is a strict contraction and the recursion $\bar{w}_i = T_c(\bar{w}_{i-1})$ converges exponentially to the minimizer, w_δ^o, of problem (6).*

□

3.2. Network Basis Transformation and Error Bounds

We are now ready to examine the behavior of the diffusion strategy (20)–(22). We employ the following common assumption on the perturbations caused by the gradient noise [3, 23, 24].

Assumption 4 (Gradient noise process)**.** *For each k, the gradient noise process is defined as*

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) = \widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) - \nabla_w J_k(\boldsymbol{w}_{k,i-1}) \quad (32)$$

and satisfies

$$\mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\boldsymbol{\mathcal{F}}_{i-1}\right] = 0 \quad (33a)$$

$$\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}\right] \le \beta^2\|\boldsymbol{w}_{k,i-1}\|^2 + \sigma^2 \quad (33b)$$

for some non-negative constants $\{\beta^2, \sigma^2\}$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration generated by the random processes $\{\boldsymbol{w}_{\ell,j}\}$ for all $\ell = 1, 2, \ldots, N$ and $j \le i-1$. □

We begin by introducing the following extended vectors and matrices, which collect quantities of interest from across all agents in the network:

$$\boldsymbol{w}_i \triangleq \mathrm{col}\left\{\boldsymbol{w}_{1,i},\ldots,\boldsymbol{w}_{N,i}\right\} \tag{34}$$

$$\boldsymbol{\phi}_i \triangleq \mathrm{col}\left\{\boldsymbol{\phi}_{1,i},\ldots,\boldsymbol{\phi}_{N,i}\right\} \tag{35}$$

$$\mathcal{A} \triangleq A \otimes I_M \tag{36}$$

$$\widehat{g}(\boldsymbol{w}_i) \triangleq \mathrm{col}\left\{\widehat{\nabla_w J_1}(\boldsymbol{w}_{1,i}),\ldots,\widehat{\nabla_w J_N}(\boldsymbol{w}_{N,i})\right\} \tag{37}$$

$$\widehat{r}(\boldsymbol{\phi}_i) \triangleq \mathrm{col}\left\{\nabla_w R_1^{\delta}(\boldsymbol{\phi}_{1,i}),\ldots,\nabla_w R_N^{\delta}(\boldsymbol{\phi}_{N,i})\right\} \tag{38}$$

where $\otimes$ denotes the Kronecker product [32, Ch. 13]. Using these definitions, iterations (20)–(22) show that the network vector $\boldsymbol{w}_i$ evolves according to the following dynamics:

$$\boldsymbol{w}_i = \mathcal{A}^{\mathsf{T}}\boldsymbol{w}_{i-1} - \mu\mathcal{A}^{\mathsf{T}}\left(\widehat{g}(\boldsymbol{w}_{i-1}) + \widehat{r}(\boldsymbol{\phi}_{i-1})\right) \tag{39}$$

By construction, the combination matrix A is left-stochastic and primitive and hence admits a Jordan decomposition of the form $A = V_\epsilon J V_\epsilon^{-1}$ with [3,24]:

$$V_\epsilon = \left[\begin{array}{c|c} p & V_R \end{array}\right], \quad J = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & J_\epsilon \end{array}\right], \quad V_\epsilon^{-1} = \left[\begin{array}{c} \mathbb{1}^{\mathsf{T}} \\ \hline V_L^{\mathsf{T}} \end{array}\right] \tag{40}$$

where J_ϵ is a block Jordan matrix with the eigenvalues $\lambda_2(A)$ through $\lambda_N(A)$ on the diagonal and ϵ on the first lower sub-diagonal. The extended matrix $\mathcal{A}$ then satisfies $\mathcal{A} = \mathcal{V}_\epsilon \mathcal{J} \mathcal{V}_\epsilon^{-1}$ with $\mathcal{V}_\epsilon = V_\epsilon \otimes I_M$, $\mathcal{J} = J \otimes I_M$, $\mathcal{V}_\epsilon^{-1} = V_\epsilon^{-1} \otimes I_M$. Multiplying both sides of (39) by $\mathcal{V}_\epsilon^{\mathsf{T}}$ and introducing the transformed iterate vector $\boldsymbol{w}_i' \triangleq \mathcal{V}_\epsilon^{\mathsf{T}}\boldsymbol{w}_i$, we obtain

$$\boldsymbol{w}_i' = \mathcal{J}^{\mathsf{T}}\boldsymbol{w}_{i-1}' - \mu\mathcal{J}^{\mathsf{T}}\mathcal{V}_\epsilon^{\mathsf{T}}\left(\widehat{g}(\boldsymbol{w}_{i-1}) + \widehat{r}(\boldsymbol{\phi}_{i-1})\right) \tag{41}$$

Motivated by the arguments in [3, 33], we partition the transformed network vector into $\boldsymbol{w}_i' = \mathrm{col}\left\{\boldsymbol{w}_{c,i}, \boldsymbol{w}_{e,i}\right\}$, where $\boldsymbol{w}_{c,i} \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{w}_{e,i} \in \mathbb{R}^{(N-1)M\times 1}$. Then, recursion (41) can be decomposed as

$$\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu\left(p^{\mathsf{T}} \otimes I_N\right)\left(\widehat{g}(\boldsymbol{w}_{i-1}) + \widehat{r}(\boldsymbol{\phi}_{i-1})\right) \tag{42}$$

$$\boldsymbol{w}_{e,i} = \mathcal{J}_\epsilon^{\mathsf{T}}\boldsymbol{w}_{e,i-1} - \mu\mathcal{J}_\epsilon^{\mathsf{T}}\mathcal{V}_R^{\mathsf{T}}\left(\widehat{g}(\boldsymbol{w}_{i-1}) + \widehat{r}(\boldsymbol{\phi}_{i-1})\right) \tag{43}$$

It can be verified from $\boldsymbol{w}_i' = \mathcal{V}_\epsilon^{\mathsf{T}}\boldsymbol{w}_i$, that $\boldsymbol{w}_{c,i} = \sum_{k=1}^N p_k \boldsymbol{w}_{k,i}$. That is, $\boldsymbol{w}_{c,i}$ is the weighted centroid vector of all iterates $\boldsymbol{w}_{k,i}$ across the network. From $\boldsymbol{w}_i = \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\boldsymbol{w}_i'$ on the other hand, it follows that $\boldsymbol{w}_i = \mathbb{1}\otimes\boldsymbol{w}_{c,i} + \mathcal{V}_L\boldsymbol{w}_{e,i}$, so that $\mathcal{V}_L\boldsymbol{w}_{e,i}$ can be interpreted as the deviation of individual estimates from the weighted centroid vector $\boldsymbol{w}_{c,i}$ across the network.

Further inspection of recursion (42) for the centroid vector $\boldsymbol{w}_{c,i}$ and comparison to the central recursion (30) reveal that (42) is a perturbed version of (30), where $\nabla_w J_k(\bar{w}_{i-1})$ is replaced by $\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1})$ and $\nabla_w R_k^{\delta}(\bar{w}_{i-1})$ is replaced by $\nabla_w R_k^{\delta}(\boldsymbol{\phi}_{k,i-1})$. It is therefore reasonable to expect that $\boldsymbol{w}_{c,i}$ will evolve close to the central variable $\bar{w}_i$ from (30), which was already shown to converge to w_δ^o in Lemma 1. This observation is formalized as follows. Let $\widetilde{\boldsymbol{w}}_{c,i} = w_\delta^o - \boldsymbol{w}_{c.i}$ denote the error vector relative to the smoothed Pareto solution w_δ^o. The proof of the following result is omitted for brevity.

Theorem 2 (Mean-square-error dynamics)**.** *Given δ, the variances of $\widetilde{\boldsymbol{w}}_{c,i}$ and $\boldsymbol{w}_{e,i}$ are coupled and recursively bounded as*

$$\begin{bmatrix}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \\ \mathbb{E}\|\boldsymbol{w}_{e,i}\|^2\end{bmatrix} \preceq \Gamma \begin{bmatrix}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 \\ \mathbb{E}\|\boldsymbol{w}_{e,i-1}\|^2\end{bmatrix} + O(\mu^2) \tag{44}$$

where

$$\Gamma = \begin{bmatrix}1-O(\mu) & O(\mu) \\ O(\mu^2) & \|\mathcal{J}_\epsilon\| + O(\mu^2)\end{bmatrix} \tag{45}$$

and $\|\mathcal{J}_\epsilon\| < 1$. It follows that, there exists small enough μ such that

$$\limsup_{i\to\infty} \begin{bmatrix}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \\ \mathbb{E}\|\boldsymbol{w}_{e,i}\|^2\end{bmatrix} \preceq \begin{bmatrix}O(\mu) \\ O(\mu^2)\end{bmatrix} \tag{46}$$

From (14), it then follows that

$$\lim_{\mu,\delta\to 0}\ \limsup_{i\to\infty}\ \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = 0. \tag{47}$$

□

4. APPLICATION TO PATTERN CLASSIFICATION

Consider random binary class variables $\boldsymbol{\gamma}_k = \pm 1$ and denote by $\boldsymbol{h}_k \in \mathbb{R}^M$ the corresponding feature vectors. During the training phase, at each time instant i, agent k receives $\{\boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}\}$. Using a logistic regression formulation, we are interested in finding a global decision rule, parametrized by w^o, such that $\widehat{\boldsymbol{\gamma}}_{\boldsymbol{k}}(i) = \boldsymbol{h}_{k,i}^{\mathsf{T}} w^o$ and

$$w^o \triangleq \arg\min_w \sum_{k=1}^N p_k \left\{\mathbb{E}\,\ln\left[1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right] + \rho_1\|w\|_1 + \rho_2\|w\|_2^2\right\} \tag{48}$$

where we are employing elastic-net regularization [4]. Let

$$J_k(w) = \mathbb{E}\,\ln\left[1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right] + \rho_2\|w\|_2^2 \tag{49}$$

and $R_k(w) = \rho_1\|w\|_1$. Then, problem (48) is of the form (3). The Moreau envelope of $\|w\|_1$ is the Huber penalty. Its proximal operator is available in closed form and given by the soft-threshold or shrinkage operator [31, 34]. Each agent in the network can hence solve for w^o by iterating (27)–(29).

The performance is illustrated in Fig. 1. The network consists of 10 agents and $\boldsymbol{h}_k \in \mathbb{R}^{20}$ is constructed according to $\boldsymbol{h}_k = \boldsymbol{\gamma}_k \cdot \mathrm{col}\left\{1, 1, 0, \ldots, 0\right\} + \boldsymbol{v}_k$, where $\boldsymbol{v}_k \in \mathbb{R}^{20}$ is drawn from $\mathcal{N}(0, \sigma_{v,k}^2 I)$. Note that only the first two elements of each feature vector contain information about $\boldsymbol{\gamma}_k(i)$ and hence w^o is sparse, as is often the case in large-scale machine learning problems. At each iteration, classification accuracy is evaluated on a seperate testing set. We compare performance to the optimal linear classifier in terms of error-propability, which, for this data model can be evaluated in closed form from the statistical properties.

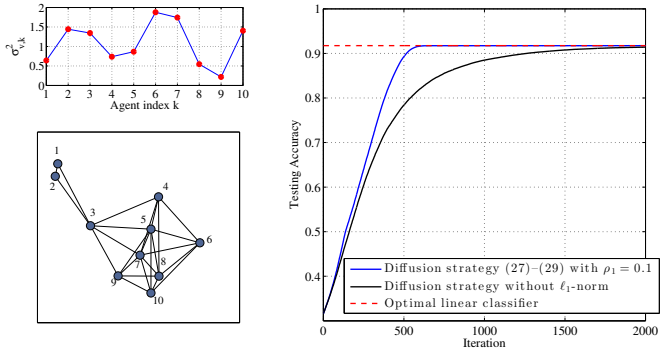

Fig. 1. Noise profile, network structure and algorithm performance, $\mu = 0.01, \delta = 0.01, \rho_2 = 0.001$.

5. REFERENCES

[1] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2003.

[2] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem: Some of its applications," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 62–75, March 2005.

[3] 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.

[4] R. Tibshirani T. Hastie and J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.

[5] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. on Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[7] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

[8] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

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

[10] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse diffusion LMS for distributed adaptive estimation," in *Proc. IEEE ICASSP*, Kyoto, Japan, March 2012, pp. 3281–3284.

[11] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, March 2013.

[12] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[13] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug 2012.

[14] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Trans. on Signal Process.*, vol. 62, no. 15, pp. 3924–3938, Aug. 2014.

[15] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2013.

[16] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, Allerton, USA, Oct. 2012, pp. 601–608.

[17] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized nondifferentiable optimization," in *Proc. IEEE ICASSP*, Brisbane, Australia, April 2015, pp. 2964–2968.

[18] P. L. Combettes and J.-C. Pesquet, "Stochastic approximations and perturbations in forward-backward splitting for monotone operators," available as arXiv:1507.07095, Jul. 2015.

[19] W. M. Wee and I. Yamada, "A proximal splitting approach to regularized distributed adaptive estimation in diffusion networks," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 5420–5424.

[20] P. Di Lorenzo, "Diffusion adaptation strategies for distributed estimation over Gaussian Markov random fields," *IEEE Transactions on Signal Process.*, vol. 62, no. 21, pp. 5748–5760, Nov. 2014.

[21] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," available as arXiv:1509.01360, Sep. 2015.

[22] S. Vlaski and A. H. Sayed, "Proximal diffusion for stochastic costs with non-differentiable regularizers," in *Proc. IEEE ICASSP*, Brisbane, Australia, April 2015, pp. 3352–3356.

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

[24] J. Chen and A.H. Sayed, "On the learning behavior of adaptive networks - Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.

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

[26] J.-B. Hiriat-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*, Cambridge University Press, New York, NY, USA, 1993.

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

[28] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pp. 185–221, Springer, NY, 2011.

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

[30] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[31] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[32] A. J. Laub, *Matrix Analysis for Scientists and Engineers*, Society for Industrial and Applied Mathematics, 2005.

[33] J. Chen and A.H. Sayed, "On the learning behavior of adaptive networks - Part II: Performance analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3518–3548, June 2015.

[34] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1001–1016, Feb. 2015.