Task-Driven Dictionary Learning in Distributed Online Settings

Alec Koppel*, Garrett Warnell†, and Ethan Stump†

***Abstract*—We consider task-driven dictionary learning in a decentralized dynamic setting. Here a network of agents while sequentially receiving local information aims to learn a common data-driven signal representation and model parameters. We formulate this problem as a distributed stochastic program with a non-convex objective and present a block variant of the Arrow-Hurwicz saddle point algorithm to solve it. Using Lagrange multipliers to penalize the discrepancy between them, only neighboring nodes exchange model information. We show that decisions made with this saddle point algorithm asymptotically converge to a stationarity condition in expectation under certain conditions. The learning rate depends on the signal source, network, and discriminative task. We illustrate the algorithm performance in an online multi-agent setting for a collaborative image classification task, demonstrating that the performance is comparable to the centralized case and depends on the network topology over which it is run.**

I. Introduction

We consider the problem of task-driven dictionary learning in dynamic multi-agent settings. Dictionary learning is the problem of finding a data-driven signal representation, and it's task-driven extension refers to the problem of tailoring the learned representation to a particular signal processing task of interest. The problem breaks down into three aspects: developing data-driven feature representations, learning task-driven classifiers over these representations, and extending this framework to networks.

Consider a vector $\mathbf{x} \in \mathbb{R}^m$. The signal $\mathbf{x}$ admits a sparse approximation of $\mathbf{x}$ over a dictionary $\mathbf{D} \in \mathbb{R}^{m\times k}$ if it may be represented as a combination of a *small* number of basis elements that is *close* to $\mathbf{x}$. Sparse approximations have been successfully applied to a variety of signal processing applications [1] such as signal reconstruction [4] and classification [2]. Recent extensions which tailor the dictionary a specific signal processing, referred to as *discriminative* dictionary learning, have led to significant improvements [3].

Dictionary learning in the online setting, where training samples are sequentially observed, has been solved as a matrix factorization problem using first [4] and second-order stochastic approximation methods [5]. However, the discriminative dictionary learning is a more challenging to optimize. Recently, an online framework for large-scale dictionary and discriminative model learning has been proposed based upon alternating stochastic gradient [6] which successfully generalizes the task-specific dictionary methods for classification [7], yet no convergence analysis was provided. We extend [6] to networked settings, where a team of agents seeks to learn a common dictionary and model parameters based upon dynamic local information. This extension yields a decentralized non-convex stochastic program, which we solve using tools from stochastic approximation and distributed optimization.

Pertinent to the proposed algorithm is the seminal work of [8] and its extensions to distributed settings. Such extensions incorporate methods from distributed optimization such as weighted averaging [9], dual decomposition [10], and primal-dual methods which combine primal descent with dual ascent [11]. [12] considers a weighted averaging method for networked stochastic optimization in the context of dictionary learning; however, as shown in [13], [14], such methods are only well-suited to problems where averaging is advantageous. In Section II we formulate the task-driven dictionary learning problem and extend it to networked settings. In Section III, we propose a block variant of the primal-dual algorithm in [14] and establish its convergence in expectation to a first-order stationary solution of the problem in Section IV. We then demonstrate the proposed framework's practical utility in the context of a collaborative learning task based upon image data in Section V.

II. Problem Formulation

Consider a set of T signals in an m-dimensional feature space $\{\mathbf{x}_t\}_{t=1}^T \subset \mathcal{X} \subset \mathbb{R}^m$. We aim to represent the signals $\mathbf{x}_t$ as sparse combination of a common set of k basis elements, which are unknown and must also be learned from the data. Denote the dictionary as $\mathbf{D} \in \mathbb{R}^{m\times k}$, the sparse coding as $\boldsymbol{\alpha} \in \mathbb{R}^k$ and associate a loss function $\tilde{f}_t(\boldsymbol{\alpha}, \mathbf{D})$ with each data point which is small when $\boldsymbol{\alpha}$ and $\mathbf{D}$ *sparsely represent* $\mathbf{x}_t$ well. Classically the the dictionary learning and sparse representation problem [15] has been formulated as the empirical loss minimization

$$\min_{\mathbf{D}\in\mathbb{R}^{m\times k}, \boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{T}\sum_{t=1}^{T} \tilde{f}_u(\boldsymbol{\alpha}, \mathbf{D}). \tag{1}$$

where the number of data points T is large, and the signal dimension m is small, i.e. image patches are frequently of size 10×10, so that $m = 100$. We allow for the overdetermined case $k \geq m$. Moreover, $T \gg k$ for big data settings, but each signal uses a small number of basis elements in its representation.

One way to induce sparsity in the coding $\boldsymbol{\alpha}$ would be with an ℓ_0 constraint, but doing so yields a computationally intractable formulation. Instead, we consider methods to compute (1) convex relaxations [17] with an elastic-net (ℓ_1 and ℓ_2) penalty, stated as

$$f_u(\mathbf{D}; \mathbf{x}) := \min_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \zeta_1\|\boldsymbol{\alpha}\|_1 + \frac{\zeta_2}{2}\|\boldsymbol{\alpha}\|_2^2. \tag{2}$$

The subscript u denotes the *unsupervised* data driven method for learning the dictionary. For a fixed $\mathbf{D}$, (2) is an elastic-net [18] problem. Here the ℓ_1-regularizer induces sparsity in $\boldsymbol{\alpha}$, tuned by a regularization parameter ζ_1. The ℓ_2 regularization guarantees (2)

Work in this paper is supported by NSF CCF-1017454, NSF CCF-0952867, ONR N00014-12-1-0997, ARL MAST CTA, and ASEE SMART.

*Department of Electrical and Systems Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Email: {akoppel, aribeiro}@seas.upenn.edu.

†U.S. Army Research Lab, Computation and Information Sciences Directorate, 2800 Powder Mill Road, Adelphi, MD 20783. Email: {ethan.a.stump2.civ, garrett.a.warnell.civ}@mail.mil

is strongly convex and may be solved uniquely, whereby ζ_2 tunes how equitably the sparse coding is spread across its k coordinates. In this paper we solve (2) using the least angle regression (LARS) [19] method, which solves for the entire regularization path, and typically is of comparable speed to soft-thresholding based methods [16], with improved accuracy and robustness.

There is no analytical link between ζ_1 and the sparsity level, and hence values of $\boldsymbol{\alpha}$ may becoming arbitrarily small, which corresponds to the entries of $\mathbf{D}$ from becoming arbitrarily large. To eliminate the scale ambiguity of the bilinear term in (2), constrain the set of feasible dictionaries to be those whose columns are of unit norm, i.e. $\mathcal{D} = \{\mathbf{D} \in \mathbb{R}^{m\times k} : \|\mathbf{d}_l\| \leq 1, l = 1 \dots k\}$.

The empirical loss (1) is an approximation of the expected loss over the entire feature space, which is actually the objective of interest in most signal processing applications and considered here. Thus block coordinate methods [20] may not be applied since the dictionary learning goal of representing an *entire* feature space requires taking the limit of (1) as $T \to \infty$, yielding

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}} \left[f_u(\mathbf{D}, \mathbf{x}) \right]. \tag{3}$$

Here we view the signal $\mathbf{x}$ as a random variable. Solving (3) amounts to finding a signal representation over all possible data realizations, achieving superior generalization capacity than (1).

We modify (3) such that the dictionary learning is *supervised* to the signal processing task of interest as in [6]. Begin by defining $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ as the optimal sparse coding solving (2) and associate with each signal $\mathbf{x}$ a variable $\mathbf{y} \in \mathcal{Y}$ which is drawn from a set of labels for classification or $\mathcal{Y} \subset \mathbb{R}^q$ for regression. We aim to learn model parameters $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^k$ relating the the pair $(\mathbf{x}, \mathbf{y})$ using the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ as a feature representation of the signal, and seek to minimize a convex smooth loss function of the form $f_s(\mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x}))$. The subscript s denotes the supervised component of the learning, which quantifies how well one may predict $\mathbf{y}$ when given the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ over the dictionary $\mathbf{D}$. Particular examples of f_s include the squared, logistic, and squared hinge-loss for linear and logistic regression or support vector classification, respectively.

We view the prediction loss f_s as a function of the model $\mathbf{w}$ *and* the dictionary $\mathbf{D}$, since the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ is dependent on the dictionary, formulating the joint optimization problem

$$\min_{\mathbf{D}\in\mathcal{D}, \mathbf{w}\in\mathcal{W}} \mathbb{E}_{\mathbf{y},\mathbf{x}} \left[f_s(\mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})) \right] + \frac{\xi}{2}\|\mathbf{w}\|^2, \tag{4}$$

where ξ is a regularization parameter guaranteeing the problem is strongly convex in $\mathbf{w}$ for fixed the dictionary and sparse coefficients. [6] establishes that smooth optimization methods solve (4) despite the non-smooth sparsity-inducing norm in (2). Both the model and dictionary are tuned for prediction risk in (4).

We aim to solve (4) in distributed settings, where the signal $\mathbf{y}$ is independently observed by agents of a network which aim to learn a dictionary and model parameters in common with all others while only having access to local information. To so, fix a network $\mathcal{G} = (V, \mathcal{E})$ which is assumed to be symmetric and connected network with node set $V = \{1, \dots, N\}$ and $M = |\mathcal{E}|$ directed edges of the form $e = (i, j)$. Define the neighborhood of i as the set of nodes $n_i := \{j : (i, j) \in \mathcal{E}\}$ that share an edge with i. Suppose the functions f_u in (2) and f_s are node-separable,

$$f_u(\mathbf{D}; \mathbf{x}) = \sum_{v=1}^{N} f_{i,u}(\mathbf{D}_i; \mathbf{x}_i), \tag{5}$$

$$f_s(\mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})) = \sum_{i=1}^{N} f_{i,s}(\mathbf{y}_i, \mathbf{w}_i, \boldsymbol{\alpha}^\star(\mathbf{D}_i; \mathbf{x}_i)). \tag{6}$$

Associated with each node i in the network are the local functions f_u and f_s parameterized by the random variable $\mathbf{x}_i$, whose explicit expressions are given by substituting the local random variable into (2) and f_s, which is dependent on the particular learning task.

The loss functions $f_{i,u}$ and $f_{i,s}$ are the same for all agents i so dictionary and model parameter selections that are good for one agent are also good for another. Thus, a suitable strategy is to learn a dictionary $\mathbf{D}_i$ and model $\mathbf{w}_i$ in the same way for each agent. Since the network $\mathcal{G}$ is assumed to be connected, this relationship can be attained by imposing the constraints $\mathbf{D}_i = \mathbf{D}_j$ and $\mathbf{w}_i = \mathbf{w}_j$ for all pairs of neighboring nodes $(i, j) \in \mathcal{E}$. Substituting (5) into the objective in (4) with these constraints, we obtain the following networked stochastic program:

$$\min_{\mathbf{D}\in\mathcal{D}^N, \mathbf{w}\in\mathcal{W}^N} \quad \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i,\mathbf{x}_i} \left[f_s(\mathbf{y}_i, \mathbf{w}_i, \boldsymbol{\alpha}_i^\star(\mathbf{D}_i; \mathbf{x}_i)) \right] + \frac{\xi}{2}\|\mathbf{w}_i\|^2. \tag{7}$$

$$\text{such that} \qquad \mathbf{D}_i = \mathbf{D}_j, \mathbf{w}_i = \mathbf{w}_j \text{ for all } j \in n_i$$

Here agent i aims to learn a common dictionary $\mathbf{D}_i$ and discriminative model $\mathbf{w}_i$ that asymptotically converges to the solution of (4). Note that when the agreement constraints in (7) are satisfied, the problems (4) and (7) are equivalent. Thus (7) corresponds to a problem in which each agent i, having observed only *local* signals $\mathbf{y}_i$, aims to learn a dictionary and model parameters that are optimal when information is globally aggregated.

III. Block Saddle Point Method

We turn to deriving an algorithmic solution to (7), the dynamic discriminative dictionary learning problem in networks. We build upon the stochastic approximation approach to dictionary learning developed in [4]. Begin by writing the constraints in (7) compactly by defining the vertical block concatenation matrices $\mathbf{D} := [\mathbf{D}_1; \dots; \mathbf{D}_N] \in \mathbb{R}^{Nm\times k}$ and $\mathbf{w} := [\mathbf{w}_1; \dots; \mathbf{w}_N] \in \mathbb{R}^{Nm}$ and an the augmented graph edge incidence matrix associated with each constraint as follows $\mathbf{C}_D : \mathbb{R}^{Nm\times k} \to \mathbb{R}^{Mm\times k}$. The matrix $\mathbf{C}_D$ is formed by $M \times N$ square blocks of dimension mk. If the edge $e = (i, j)$ links node i to node j the block (e, i) is $[\mathbf{C}_D]_{ei} = \mathbf{I}_{mk}$ and the block $[\mathbf{C}_D]_{ej} = -\mathbf{I}_{mk}$, where $\mathbf{I}_{mk}$ denotes the identity matrix of dimension mk. All other blocks are identically null, i.e., $[\mathbf{C}]_{el} = \mathbf{0}_{mk}$ for all edges $e \neq (i, j)$. The matrix $\mathbf{C}_w$ is defined in the exact same way, substituting the model parameter dimension k for the dictionary dimension mk. Then the constraints $\mathbf{D}_i = \mathbf{D}_j$ and $\mathbf{w}_i = \mathbf{w}_j$ for all pairs of neighboring nodes can be written as $\mathbf{C}_D\mathbf{D} = \mathbf{0}, \quad \mathbf{C}_w\mathbf{w} = \mathbf{0}$. The edge incidence matrices $\mathbf{C}_D$ and $\mathbf{C}_w$ have exactly mk and m null singular values, respectively. Denote as $0 < \gamma$ the smallest nonzero singular value of $\mathbf{C} := [\mathbf{C}_D; \mathbf{C}_w]$ and as Γ the largest singular value of $\mathbf{C}$, both of which measure network connectedness.

Imposing the constraints $\mathbf{C}_D\mathbf{D} = \mathbf{0}$ and $\mathbf{C}_w\mathbf{w} = \mathbf{0}$ for all realizations of the local random variables requires global coordination. Instead, we consider a modification of (5) in which we add linear penalty terms to incentivize the selection of coordinated decision variables, which is tantamount to a block stochastic variant of the Arrow-Hurwicz Saddle Point Algorithm [11], [13], [14]. Introduce then dual variables $\boldsymbol{\Lambda}_e = \boldsymbol{\Lambda}_{ij} \in \mathbb{R}^{m\times k}$ associated with the constraint $\mathbf{D}_i - \mathbf{D}_j = \mathbf{0}$ and consider the addition of penalty terms of the form $\text{tr}[\boldsymbol{\Lambda}_{ij}^T(\mathbf{D}_i - \mathbf{D}_j)]$. For an edge that starts at node i, the multiplier $\boldsymbol{\Lambda}_{ij}$ is assumed to be kept at node i. Similarly, introduce

dual variables $\boldsymbol{\nu}_{ij}$ associated with the constraint $\mathbf{w}_i - \mathbf{w}_j = \mathbf{0}$ for all neighboring node pairs and penalty terms $\boldsymbol{\nu}_{ij}^T(\mathbf{w}_i - \mathbf{w}_j)$. By introducing the stacked matrices $\boldsymbol{\Lambda} := [\boldsymbol{\Lambda}_1; \ldots; \boldsymbol{\Lambda}_M] \in \mathbb{R}^{Mm \times k}$ and $\boldsymbol{\nu} := [\boldsymbol{\nu}_1; \ldots; \boldsymbol{\nu}_M] \in \mathbb{R}^{m \times k}$, we define the Lagrangian of the optimization problem (7) as

$$\mathcal{L}(\mathbf{D}, \mathbf{w}, \boldsymbol{\Lambda}, \boldsymbol{\nu}) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i, \mathbf{x}_i} \left[f_s(\mathbf{y}_i, \mathbf{w}_i, \boldsymbol{\alpha}_i^\star(\mathbf{D}_i; \mathbf{x}_i)) \right] + \frac{\xi}{2} \|\mathbf{w}_i\|^2 + \operatorname{tr}\big(\boldsymbol{\Lambda}^T \mathbf{C}_D \mathbf{D}\big) + \boldsymbol{\nu}^T \mathbf{C}_w \mathbf{w} \tag{8}$$

Suppose agent i receives a realization of the local random variables at time t as $\mathbf{x}_{i,t}$ with associated output (label) $\mathbf{y}_{i,t}$. Using this interpretation of the Lagragian we use of the Arrow-Hurwicz saddle point method in alternating block variable updates, which exploits the fact that primal-dual optimal pairs are saddle points of the Lagrangian to work through successive primal alternating gradient descent steps and dual gradient ascent steps.

Definition 1 *For the Lagrangian in* (8)*, the primal update of the saddle point algorithm takes the form*

$$\mathbf{D}_{t+1} = \mathcal{P}_{\mathcal{D}}\big[\mathbf{D}_t - \epsilon_t \nabla_{\mathbf{D}} \hat{\mathcal{L}}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\big], \tag{9}$$

$$\mathbf{w}_{t+1} = \mathcal{P}_{\mathcal{W}}\big[\mathbf{w}_t - \epsilon_t \nabla_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\big] \ . \tag{10}$$

Likewise, the dual update is given as

$$\boldsymbol{\Lambda}_{t+1} = \mathcal{P}_{\mathfrak{L}}\big[\boldsymbol{\Lambda}_t + \epsilon_t \nabla_{\boldsymbol{\Lambda}} \hat{\mathcal{L}}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\big], \tag{11}$$

$$\boldsymbol{\nu}_{t+1} = \mathcal{P}_{N}\big[\boldsymbol{\nu}_t + \epsilon_t \nabla_{\boldsymbol{\nu}} \hat{\mathcal{L}}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\big], \tag{12}$$

where ϵ_t is a given stepsize, $\mathcal{P}_{\mathfrak{L}}(\boldsymbol{\lambda})$, $\mathcal{P}_N(\nu)$ denotes projection of dual variables on given convex compact set $\mathfrak{L}$ and N. The notation $\mathcal{P}_{\mathcal{D}}(\mathbf{D})$ denotes projection onto the set of feasible primal variables so that we have $\mathbf{D}_i \in \mathcal{D}$ for all the N blocks of the matrix $\mathbf{D} := [\mathbf{D}_1; \ldots; \mathbf{D}_N]$, and similarly for $\mathbf{w} \in \mathcal{W}$.

We assume that the set of multipliers Λ can be written as a Cartesian product of sets Λ_{jk} so that the projection of $\boldsymbol{\lambda}$ into Λ is equivalent to the separate projection of the components $\boldsymbol{\lambda}_{jk}$ into the sets Λ_{jk}, and similarly for N_{jk}. See [21], Section III.

Consider the optimality conditions of (2) (see [6]), which may be uniquely satisfied due to the strongly convex regularization. Define $Z_{i,t} \subset \{1, \ldots, k\}$ as the set of nonzero entries of $\boldsymbol{\alpha}_{i,t}^\star$ and a vector $\beta_{i,t}^\star \in \mathbb{R}^m$ as

$$\begin{aligned} \beta^\star_{Z_{i,t}} &= ([\mathbf{D}_i]_Z^T [\mathbf{D}_i]_Z + \zeta_2 I)^{-1} \nabla_{\boldsymbol{\alpha}_{Z_{i,t}}} f_s(\mathbf{y}_{i,t}, \mathbf{w}_{i,t}, \boldsymbol{\alpha}^\star_{i,t}(\mathbf{D}_{i,t}; \mathbf{x}_{i,t})), \\ \beta^\star_{Z^c_{i,t}} &= 0, \end{aligned} \tag{13}$$

which is the result of substituting the solution of (2) into f_s and applying the chain rule. Then at time t, applying Proposition 1 of [6], we obtain a completely decentralized algorithm, stated in the following proposition, and derived in detail in [21], Section III.

Proposition 1 *The updates in* (9)-(17) *may be separated along the components $\mathbf{D}_{j,t}$,$\mathbf{w}_{j,t}$ associated with node j, yielding $2N$ respective parallel updates of the form*

$$\mathbf{D}_{i,t+1} = \mathcal{P}_{\mathcal{D}}\Big[\mathbf{D}_{i,t} - \epsilon_t\Big(- \mathbf{D}_{i,t} \beta^\star_{i,t} \boldsymbol{\alpha}^\star_{i,t} + (\mathbf{x}_{i,t} - \mathbf{D}_{i,t} \alpha^\star_{i,t}) {\beta^\star_{i,t}}^T + \sum_{j \in n_i} (\boldsymbol{\Lambda}_{ij,t} - \boldsymbol{\Lambda}_{ji,t})\Big)\Big]. \tag{14}$$

$$\mathbf{w}_{i,t+1} = \mathcal{P}_{\mathcal{W}}\Big[\mathbf{w}_{i,t} - \epsilon_t\Big(\nabla_{\mathbf{w}_i} f_s(\mathbf{y}_i, \mathbf{w}_i, \boldsymbol{\alpha}^\star_{i,t}) + \xi \mathbf{w}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\nu}_{ij,t} - \boldsymbol{\nu}_{ji,t})\Big)\Big]. \tag{15}$$

Likewise, along edge (j,k), Lagrange multipliers $\boldsymbol{\Lambda}_{jk,t}$,$\mathbf{N}_{jk,t}$ are updated as

$$\boldsymbol{\Lambda}_{ij,t+1} = \mathcal{P}_{\mathfrak{L}_{ij}}\Big[\boldsymbol{\Lambda}_{ij,t} + \epsilon_t \left(\mathbf{D}_{i,t} - \mathbf{D}_{j,t}\right)\Big] \tag{16}$$

$$\boldsymbol{\nu}_{ij,t+1} = \mathcal{P}_{\mathfrak{N}}\Big[\boldsymbol{\nu}_{ij,t} + \epsilon_t \left(\mathbf{w}_{i,t} - \mathbf{w}_{j,t}\right)\Big] \tag{17}$$

where we have used $\boldsymbol{\alpha}^\star_{i,t}$ as shorthand for $\boldsymbol{\alpha}^\star_{i,t}(\mathbf{D}_{i,t}; \mathbf{x}_{i,t})$, and the set projections are as in Definition 1.

Node j can implement (14)-(17) using local variables and receiving dual variables $\boldsymbol{\Lambda}_{jk}$, $\boldsymbol{\nu}_{jk}$ which are sent along network communication links.

IV. Convergence Analysis

We establish that the saddle point algorithm in (9)-(17) asymptotically converges to a stationary point of the problem (7) and consequently solve (3) in a decentralized manner. In order to obtain these results, some conditions are required of the algorithm step-size, data distribution, dual variables, and network. All proofs may be found in [21]. We state these assumptions below.

(A1) The network $\mathcal{G}$ is connected. The smallest nonzero singular value of the incidence matrix $\mathbf{C}$ is γ, the largest singular value is Γ, and the network diameter is D.

(A2) The Lagrangian has Lipschitz continuous gradients in the primal and dual variables with constants $L_{\mathbf{D}}$, $L_{\mathbf{w}}$, $L_{\boldsymbol{\Lambda}}$, and $L_{\boldsymbol{\nu}}$. This implies that, e.g.,

$$\|\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{w}, \boldsymbol{\Lambda}, \boldsymbol{\nu}) - \nabla_{\mathbf{D}} \mathcal{L}(\tilde{\mathbf{D}}, \mathbf{w}, \boldsymbol{\Lambda}, \boldsymbol{\nu})\| \leq L_{\mathbf{D}} \|\mathbf{D} - \tilde{\mathbf{D}}\|_F \ . \tag{18}$$

Moreover, the gradients of the Lagrangian in the primal and dual variables are bounded with block constants $G_{\mathbf{D}}$, $G_{\mathbf{w}}$, $G_{\boldsymbol{\Lambda}}$, and $G_{\boldsymbol{\nu}}$, which implies that, e.g.,

$$\|\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{w}, \boldsymbol{\Lambda}, \boldsymbol{\nu})\| \leq G_{\mathbf{D}} \ . \tag{19}$$

(A3) The algorithm step size $\epsilon_t > 0$ for all t is such that (i) $\sum_{t=0}^{\infty} \epsilon_t = \infty$ (ii) and $\sum_{t=0}^{\infty} \epsilon_t^2 < \infty$

(A4) (Stochastic Approximation Error) The stochastic gradients of the Lagrangian are unbiased estimators for the true gradients, which for instance implies

$$\mathbb{E}\left[\nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\right] = \nabla_{\mathbf{D}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t) \ . \tag{20}$$

Moreover, let $\mathcal{F}_t$ be a sigma algebra that measures the history of the system up until time t. Then, the conditional second moments of the stochastic gradients are bounded by σ^2 for all times t, which for example allows us to write

$$\mathbb{E}\left[\|\nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\|^2 \,\middle|\, \mathcal{F}_t\right] \leq \sigma^2. \tag{21}$$

Assumption 1 is standard in distributed algorithms. Assumption 2 is satisfied in most applications intrinsically by the data. The step size rules in Assumption 3 are standard in stochastic approximation literature. Moreover, Assumption 4 is standard in approximation – see [8]With these bounds established, we may state our main result: the proposed algorithm asymptotically converges in expectation to a stationarity condition of the Lagrangian associated with the optimization problem stated in (7).

Theorem 1 *Denote $(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)$ as the sequence generated by the block saddle point algorithm in* (9)-(17)*. If Assumptions 1 - 4 hold true, then the first-order stationary condition with respect to the primal variables*

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0 \ , \tag{22}$$

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0 \tag{23}$$

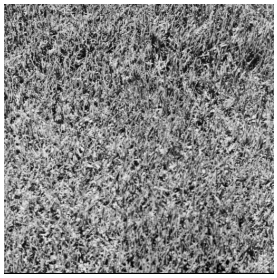



Fig. 1: Sample images from the Brodatz texture database.

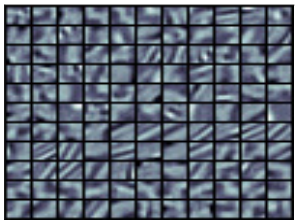
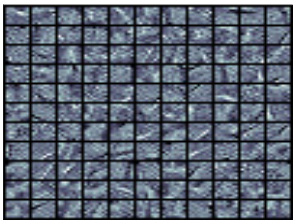



Fig. 2: Initialized (left) and final (right) dictionary for 8-by-8 grayscale patches. These dictionaries were computed using the centralized ($N = 1$) algorithm with step-size $\epsilon = 0.25$.

is asymptotically achieved in expectation. Moreover, the asymptotic feasibility condition

$$\lim_{t\to\infty} \mathbb{E}[\|\nabla_{\mathbf{\Lambda}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0 \tag{24}$$

$$\lim_{t\to\infty} \mathbb{E}[\|\nabla_{\boldsymbol{\nu}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0 \tag{25}$$

is attained in an expected sense.

Theorem 1 guarantees that the block saddle point method as stated in (9) - (17) solves the problem of learning a dictionary and discriminative model over that dictionary representation of the feature space in a decentralized online manner. In particular, a first-order stationarity condition of the Lagrangian associated with this problem is achieved asymptotically in expectation.

V. Simulations

We turn to the practical consequences of Theorem 1 by studying the algorithm performance on a canonical computer vision task. In particular, we study the performance of D4L for a multi-class texture classification problem on the Brodatz dataset [23] in order to understand the numerical properties of the algorithm for a variety of network sizes and topologies. In the case of studying the impact of network size, we also compare the algorithm performance to the centralized case, i.e. $N = 1$. For the subsequent experiments we restrict ourselves to $C = 4$ class labels {grass, bark, straw, herringbone_weave} of the Brodatz textures, samples of which are shown in Figure 1. This subset contains one grayscale image per texture, which amounts to thirteen 512-by-512 images in total consisting of $956,484$ overlapping 24×24 patches.

A. Feature Generation

Inspired by the two-dimensional textons in [24], we generate texture features to classify, $\mathbf{z}$, as the sum of the sparse dictionary representations of sub-patches of size 24×24 by first extracting the nine non-overlapping 8-by-8 sub-patches within it and vectorizing each normalized sub-patch to obtain a matrix $\mathbf{X} = \left[\mathbf{x}^{(1)}; \cdots; \mathbf{x}^{(9)}\right]$. We then compute the feature $\mathbf{z}_{i,t}$ at agent i at time t as the aggregate over sparse codings of sub-patches as $\mathbf{z}_i(\mathbf{X}_{i,t}, \mathbf{D}_{i,t}) = \sum_{l=1}^{9} \boldsymbol{\alpha}^\star(\mathbf{D}_{i,t}; \mathbf{x}_{i,t}^{(l)})$, which means that at time t the local stochastic gradient of the dictionary $[\nabla_{\mathbf{D}_i} \hat{f}_{i,s}]_t$ is the sum of contributions from each sub-patch representation.

We cast texture classification as a multi-class logistic regression problem in which agent i receives signals $\mathbf{x}_{i,t}$ and selects a binary decision variable $\mathbf{y}_{i,t} \in \{0,1\}^C$ where C is the number of classes, whose c_{th} entry is a binary indicator of whether the signal belongs to class c. The supervised local loss $f_{i,s}$ for this problem specification is the negative log-likelihood of the corresponding probabilistic model (see [25]) stated as

$$f_{i,s}(\mathbf{y}_i, \mathbf{W}_i, \mathbf{z}_i) = \log\left(\sum_{c=1}^{C} e^{\mathbf{w}_{i,c}^T \mathbf{z}_i + w_{i,c}^0}\right) - \sum_{c=1}^{C} y_{i,c} \mathbf{w}_{i,c}^T \mathbf{z}_i + w_{i,c}^0 \,, \tag{26}$$

with activation functions $g_c(\mathbf{z}_i) = e^{\mathbf{w}_{i,c}^T \mathbf{z}_i}$ computed using the c^{th} column $\mathbf{w}_c$ of the weight matrix $\mathbf{W}_i \in \mathbb{R}^{(k+1)\times C}$. To ensure identifiability, every element of the last column of $\mathbf{W}_i$ is set to zero and $w_{i,c}^0$ is a bias term for each class c. With $\mathbf{W}_i$, the probability that $\mathbf{z}_i$ belongs to class c is given by $g_c(\mathbf{z}_i)/\sum_{c'} g_{c'}(\mathbf{z}_i)$, the classification decision is made by selecting the maximum likelihood class label, i.e. $\tilde{c} = \operatorname{argmax}_c g_c(\mathbf{z}_i)/\sum_{c'} g_{c'}(\mathbf{z}_i)$ and consequently the only nonzero element of $\mathbf{y}_i$ is its $\tilde{c}$th entry.

Besides the local loss $f_{i,s}$, whose stochastic gradient almost surely converges in magnitude to null as a consequence of Theorem 1, we also study the network average classification accuracy $\sum_{i=1}^{N} P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})/N$ at each iteration. Here $\mathbf{y}_{i,t}$ denotes the true texture label, $\hat{\mathbf{y}}_{i,t}$ denotes the predicted label, and $P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})$ represents the empirical classification rate on a fixed test set of size $\tilde{T} = 4.096 \times 10^3$. We also consider the relative variation of the classifiers, stated as

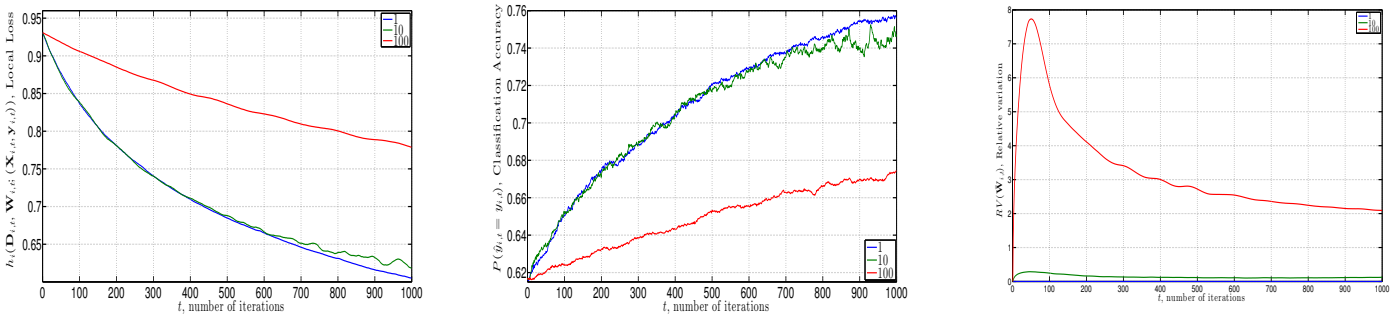
$$\text{RV}(\bar{\mathbf{W}}_{i,t}) = \frac{1}{N} \sum_{j=1}^{N} \|\bar{\mathbf{W}}_{i,t} - \bar{\mathbf{W}}_{j,t}\|_F, \tag{27}$$

where $\bar{\mathbf{W}}_{i,t} = \sum_{s=1}^{t} \mathbf{W}_{i,s}/t$ which quantifies how far individual agents' classifiers are from consensus . We consider time averages $\bar{\mathbf{W}}_{i,t}$instead of the plain estimates $\mathbf{W}_{i,t}$ because the latter tend to oscillate around the stationary point $\bar{\mathbf{W}}^*$ and agreement between estimates of different agents is difficult to visualize.

We subsequently describe the problem parameters used in our implementation. Following [6], we select regularization parameters $\zeta_1 = 0.125$, $\zeta_2 = 0$, $\xi = 10^{-9}$, and adopt the learning-rate selection strategy discussed in [6]: select the initial step-size ϵ by executing a grid search over a fixed small number of iterations ($\tilde{T} = 2 \times 10^2$) and selecting the one that minimized the cross-validation error. We set the step-size $\epsilon_t = \min(\epsilon, \epsilon t_0/t)$, where $t_0 = T/2$. We select dictionaries with $k = 128$ atoms via a numerical study of the dictionary dimension (see [21]).

We adopt a mini-batching procedure: we replace the single labeled patch with a small batch of 256 randomly-drawn labeled patches, and then average the gradient values contributed by each individual patch. This procedure reduces the variance of the local stochastic gradients. Moreover, we initialize $\mathbf{D}$ using unsupervised dictionary learning [5] for a small set of randomly-drawn initialization data, and use the associated data labels to initialize the classifier parameters $\mathbf{W}$. All experiments are run from a common initialization. Experimentally we observe that values of ϵ which yield convergent behavior are smaller than effective values for the centralized version [6] by an order of magnitude or more, and hence we select ϵ that yields convergence for both settings. For the Brodatz dataset, we found that $\epsilon = 5 \times 10^{-2}$ led to convergence.

To investigate the dependence of the convergence rate in Theorem 1 on the network size N we run (14)-(17) for problem instances with $N = 1$ (centralized), $N = 10$, and $N = 100$



(a) Local loss versus iteration t (b) Classification accuracy vs. iteration t (c) Relative variation $\mathrm{RV}(\bar{\mathbf{W}}_{i,t})$ vs. iteration t

Fig. 3: Learning achieved by an arbitrary agent in networks of size $N = 1$ (centralized), $N = 10$, and $N = 100$ with nodes randomly connected with prob. $\rho = 0.2$. 3a-3b show $f_{i,s}(\mathbf{y}_{i,t},\mathbf{W}_{i,t},\mathbf{z}_{i,t})$ and $\sum_{i=1}^{N} P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})/N$ versus iteration t, both of which decline more quickly in smaller networks. Figure 3c shows that network disagreement in terms of $RV(\bar{\mathbf{W}}_{j,t})$ becomes more stable and declines faster with smaller N.

nodes. For the later two cases, cnnections between nodes are random, with the probability of two nodes being connected set to $\rho = 0.2$. In this experiment, each agent observes training examples from all label classes. Figure 3 shows the results of this numerical experiment for a randomly selected agent in the network. Figure 3a shows $f_{i,s}(\mathbf{y}_{i,t},\mathbf{W}_{i,t},\mathbf{z}_{i,t})$ over iteration t. Observe that as N increases, the log-likelihood $f_{i,s}(\mathbf{y}_{i,t},\mathbf{W}_{i,t},\mathbf{z}_{i,t})$ declines at comparable rates for networks of moderate size, yet it significantly slower for the $N = 100$ node network. To be specific, both the centralized and $N = 10$ node network achieve $f_{i,s}(\mathbf{y}_{i,t},\mathbf{W}_{i,t},\mathbf{z}_{i,t}) \leq 0.62$ by $T = 10^3$, while the $N = 100$ node network remains at 0.77 over its run. We may observe this performance discrepancy more concretely in Figure 3b which shows the classification accuracy on a fixed test set over iteration t. The centralized algorithm achieves an accuracy near 76%, whereas the decentralized methods achieve an accuracy of 75% and 67% for the $N = 10$ and $N = 100$ node networks by $T = 10^3$ iterations, respectively. We next investigate how far the agents are from consensus as measured by $\mathrm{RV}(\bar{\mathbf{W}}_{j,t})$ over iteration t in Figure 3c. Observe that for the $N = 10$ and $N = 100$ node networks the algorithm achieves $\mathrm{RV}(\bar{\mathbf{W}}_{j,t}) \leq 1.3 \times 10^{-1}$ by $t \geq 312$ and $\mathrm{RV}(\bar{\mathbf{W}}_{j,t}) \leq 3$ for $t \geq 400$. Thus the $N = 100$ node network converges to consensus an order of magnitude slower than the $N = 10$ node network. Overall networks of moderate size achieve comparable performance to the centralized case.

REFERENCES

[1] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[2] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 759–766. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273592

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis." in *CVPR*. IEEE Computer Society, 2008. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2008.html#MairalBPSZ08

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Trans. Img. Proc.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIP.2006.881969

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756008

[6] F. Bach, J. Mairal, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, Dec. 2008. [Online]. Available: http://dx.doi.org/10.1007/s10994-007-5040-8

[8] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177729586

[9] D. Jakovetic, J. M. F. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *CoRR*, vol. abs/1112.2972, Apr. 2011.

[10] F. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 450–466, Feb. 2013.

[11] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, ser. Stanford Mathematical Studies in the Social Sciences. Stanford University Press, Stanford, Dec. 1958, vol. II.

[12] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*. IEEE, 2013, pp. 133–136.

[13] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization." in *Proc. Int. Conf. Acoust. Speech Signal Process.*, May 4-9 2014, pp. 8292–8296.

[14] A. Koppel, F. Jakubeic, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, p. 14, Sept. 2014, available at http://www.seas.upenn.edu/ aribeiro/wiki.

[15] M. Aharon, M. Elad, and A. Bruckstein, "-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1137/080716542

[17] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," Preprint arXiv:0904.3523, Tech. Rep., 2009.

[18] W. J. Fu, "Penalized regressions: the bridge versus the lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[20] P. Tseng and C. O. L. Mangasarian, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim Theory Appl*, pp. 475–494, 2001.

[21] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4l: Decentralized dynamic discriminative dictionary learning," *in preparation*, 2015.

[22] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *ArXiv preprint 1408.2597v2*, Aug. 2014.

[23] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Dover, 1966.

[24] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearence of Materials using Three-dimensional Textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 1999.

[25] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.