D4L: Decentralized Dynamic Discriminative Dictionary Learning

Alec Koppel[*], Garrett Warnell[†], Ethan Stump[†], and Alejandro Ribeiro[*]

Abstract—We consider discriminative dictionary learning in a distributed online setting, where a network of agents aims to learn a common set of dictionary elements of a feature space and model parameters while sequentially receiving observations. 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 achieve a first-order stationarity condition on average. 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 and show that practical performance is comparable to the centralized case. Moreover, in the multi-class setting, the proposed framework empirically allows nodes to make global inferences despite only observing distinct subsets of the feature space. We apply the proposed method to a mobile robotic team performing collaborative navigability assessment in an unknown environment, demonstrating the proposed algorithm's utility in a field setting.

I. INTRODUCTION

We develop a system to allow a multi-agent system to collectively perform high-level signal processing tasks such as regression or classification in unknown dynamic environments. The problem formulation breaks down into three aspects: developing data-driven feature representations, learning task-driven model parameters over these representations, and extending these formulations to dynamic, networked settings. We consider an application to decentralized online texture classification on a mobile robotic team for the purpose of analyzing the navigability of terrain over which the platforms are traversing in real-time.

Consider the problem of computing an alternative representation of a set of vectors where this alternative representation may reveal latent relationships between them. Broadly, this problem is referred to as unsupervised learning, and techniques designed to address it have yielded important advances in a variety of signal processing applications [1]. To learn such a representation, a variety of objectives may be considered. If the vector's dimension is very large, dimensionality reduction is of interest, whereby one aims to find a representation that explains the most data variability across a feature space. Classically this task been approached with principle component analysis [2], which requires orthogonality of the basis elements. Alternatively, if specialized

domain knowledge is available, finding representations based on particularized functions, i.e. wavelets for natural imagery [3], is more appropriate. A more general approach to seeking signal representations of a feature space is to learn the basis elements from data, as in dictionary learning. Dictionary learning has been successfully applied to signal reconstruction tasks such as inpainting or denoising [4]–[6], and higher level signal processing tasks such as classification [7], [8].

An important question recently posed in [9] is *why* one aims to learn a signal representation from data, if not for feeding into a higher-level signal processing task. Thus the authors in [9] propose tailoring the dictionary to a discriminative modeling task, referred to as *discriminative* dictionary learning. Such methods have recently shown promise as compared to their unsupervised counterparts [10]–[12]. The problem of developing a dictionary representation of a signal specifically suited to the learning problem of interest is a difficult optimization problem. In the centralized offline setting, this class of problems has been solved with block coordinate descent [13], [14], or alternating gradient methods [15]; however, these techniques are only effective when the training set is static and not too large. In the centralized online setting, prior approaches have made use of stochastic approximation methods [4], [16].

In this paper, we extend the online discriminative dictionary learning formulation of [9] to networked settings, where a team of agents seeks to learn a common dictionary and model parameters based upon local dynamic information. To do so, we consider tools from stochastic approximation [17] and its decentralized extensions which have incorporated ideas from distributed optimization such as weighted averaging [18]–[21], [25], dual reformulations where each agent ascends in the dual domain [22], [23], and primal-dual methods which combine primal descent with dual ascent [18], [24], [26], [27].

Our main technical contribution is the formulation of the dynamic multi-agent discriminative dictionary learning problem as a distributed stochastic program, and the development of a block variant of the primal-dual algorithm proposed in [27]. Moreover, we establish that the proposed method converges in expectation to a first-order stationary solution of the problem. We describe the discriminative dictionary learning and sparse representation problem in Section II. We extend this problem to multi-agent settings, and derive an algorithmic solution which is a block variant of the saddle point algorithm of Arrow and Hurwicz [24], [27] in Section III. In Section IV, we prove that the proposed algorithm converges in expectation to a first-order stationary condition of the problem. In Section V, we analyze the proposed framework's empirical performance on a texture classification problem based upon image data for a variety of network settings and demonstrate its capacity to solve a new class of collaborative multi-class classification problems in multi-

[*]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: {garrett.a.warnell.civm, ethan.a.stump2.civ}@mail.mil

agent settings. In Section VI we consider the algorithm's use in a mobile robotic team for navigability assessment. We conclude in Section VII.

II. DISCRIMINATIVE DICTIONARY LEARNING

Consider a set of T signals $\{\mathbf{x}_t\}_{t=1}^{T}$ each of which lives in an m-dimensional feature space so that we have $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^m$. We aim to represent the signals $\mathbf{x}_t$ as combinations of a common set of k linear basis elements $\{\mathbf{d}_l\}_{l=1}^{k}$, which are unknown and must also be learned from the data. We group these k basis elements into a dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ and denote the coding of $\mathbf{x}_t$ as $\boldsymbol{\alpha}_t \in \mathbb{R}^k$. For a given dictionary, the coding problem calls for finding a representation $\boldsymbol{\alpha}_t$ such that the signal $\mathbf{x}_t$ is close to its dictionary representation $\mathbf{D}\boldsymbol{\alpha}_t$. This goal can be mathematically formulated by introducing a loss function $f(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t)$ that depends on the proximity between $\mathbf{D}\boldsymbol{\alpha}_t$ and the data point $\mathbf{x}_t$ and formulating the coding problem as [28]

$$\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t) := \underset{\boldsymbol{\alpha}_t \in \mathbb{R}^k}{\operatorname{argmin}} f(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) . \tag{1}$$

Hereafter, we assume that basis elements are normalized to have norms $\|\mathbf{d}_l\| \leq 1$ so that the dictionary is restricted to the convex compact set $\mathcal{D} := \{\mathbf{D} \in \mathbb{R}^{m \times k} : \|\mathbf{d}_l\| \leq 1, \text{ for all } l\}$.

The dictionary learning problem associated with the loss function $f(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t)$ entails finding a dictionary $\mathbf{D}$ such that the signals $\mathbf{x}_t$ are close to their representations $\mathbf{D}\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ for all possible t. Here, however, we focus on discriminative problems where the goal is to find a dictionary that is well adapted to a specific classification or regression task [9]. Formally, we associate with each $\mathbf{x}_t$ a variable $\mathbf{y}_t \in \mathcal{Y}$ that represents a discrete label – in the case of classification problems – or a set of associated vectors $\mathcal{Y} \subset \mathbb{R}^q$ – in the case of regression. We then use the coding $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x})$ in (5) as a feature representation of the signal $\mathbf{x}_t$ and introduce the classifier $\mathbf{w}$ that is used to predict the label or vector $\mathbf{y}_t$ when given the signal $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x})$. The merit of the classifier $\mathbf{w}$ is measured by the smooth loss function $h(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{D}, \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t))$ that captures how well the classifier $\mathbf{w}$ may predict $\mathbf{y}_t$ when given the sparse coding $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ that we compute using the dictionary $\mathbf{D}$. The discriminative dictionary learning problem is then formulated as the joint determination of the dictionary $\mathbf{D} \in \mathcal{D}$ and classifier $\mathbf{w} \in \mathcal{W}$ that minimize the cost $h(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t))$ averaged over the T-element training set,

$$(\mathbf{D}^*, \mathbf{w}^*) := \underset{\mathbf{D} \in \mathcal{D}, \mathbf{w} \in \mathcal{W}}{\operatorname{argmin}} \frac{1}{T} \sum_{t=1}^{T} h(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{D}, \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)). \tag{2}$$

For given dictionary $\mathbf{D}$ and signal sample $\mathbf{x}_t$ we compute the code $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ as per (1), predict $\mathbf{y}_t$ using $\mathbf{w}$, and measure prediction error with the loss function $h(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t))$. The optimal pair $(\mathbf{D}^*, \mathbf{w}^*)$ in (2) is the one that minimizes the cost averaged over the given sample pairs $(\mathbf{x}_t, \mathbf{y}_t)$. Observe that $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ is not a variable in the optimization in (2) but a mapping for a implicit dependence of the loss on the dictionary $\mathbf{D}$. To simplify notation we henceforth write (2) as

$$(\mathbf{D}^*, \mathbf{w}^*) := \underset{\mathbf{D} \in \mathcal{D}, \mathbf{w} \in \mathcal{W}}{\operatorname{argmin}} \frac{1}{T} \sum_{t=1}^{T} h(\mathbf{D}, \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) . \tag{3}$$

The optimization problem in (4) is not assumed to be convex – this would be too restrictive because the dependence of h on

$\mathbf{D}$ is, partly, through the mapping $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ defined by (1). In general, only local minima of (3) can be found.

Our goal in this paper is to study online algorithms that solve (3) as training pairs $(\mathbf{x}_t, \mathbf{y}_t)$ become available. To do so we introduce the assumption that training pairs $(\mathbf{x}_t, \mathbf{y}_t)$ are independently sampled from a common probability distribution and replace (3) by

$$(\mathbf{D}^*, \mathbf{w}^*) := \underset{\mathbf{D} \in \mathcal{D}, \mathbf{w} \in \mathcal{W}}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}, \mathbf{y}} \Big[h(\mathbf{D}, \mathbf{w}; (\mathbf{x}, \mathbf{y})) \Big]. \tag{4}$$

The problems in (4) and (3) are equivalent in the limit of $T \to \infty$ if $(\mathbf{x}_t, \mathbf{y}_t)$ are independently drawn from the joint distribution of the random pair $(\mathbf{x}, \mathbf{y})$. The problem in (4), as the one in (3), is not convex. We clarify the problem formulation in (4) with two representative examples.

Example 1 (Sparse non-discriminative learning) When we have $k < m$, the formulation in (1) aims at finding a dictionary that reduces data dimensionality from m to k. In this paper we are more interested in the overdetermined case in which $k > m$ but we want the codes $\boldsymbol{\alpha}_t$ to be sparse. These sparsity constraints can be written as upper limits on the zero norm of $\boldsymbol{\alpha}_t$ but that would yield computationally intractable formulations. To circumvent this issue, sparsity can be incentivized by adding, e.g., elastic net regularization terms [29], [30], in which case we can write the loss functions $f(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t)$ in (1) as

$$f(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) = \tilde{f}(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) + \zeta_1 \|\boldsymbol{\alpha}_t\|_1 + \frac{\zeta_2}{2} \|\boldsymbol{\alpha}_t\|_2^2 . \tag{5}$$

In (5), $\tilde{f}(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t)$ measures proximity between $\mathbf{x}_t$ and $\mathbf{D}\boldsymbol{\alpha}_t$, the ℓ_1 term $\zeta_1 \|\boldsymbol{\alpha}_t\|_1$ encourages sparsity, and the ℓ_2 term $(\zeta_2/2) \|\boldsymbol{\alpha}_t\|_2^2$ is a smooth regularizer. Common choices for the proximity functions are the Euclidean distance $\tilde{f}(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|/2$ and the l_∞ norm $\tilde{f}(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_\infty = \max_i |\mathbf{x}_{i,t} - \mathbf{D}_i \boldsymbol{\alpha}_{i,t}|$. In a non-discriminative problem we simply want to make $\mathbf{x}_t$ and $\mathbf{D}\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ close to each other across elements of the training set. We achieve that by simply making $h(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{D}, \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) = f(\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{D}; (\mathbf{x}_t, \mathbf{y}_t))$.

Example 2 (Sparse logistic regression) Given a training set of pairs $(\mathbf{x}_t, y_t)$ where $\mathbf{x}_t \in \mathbb{R}^p$ is a feature vector with associated binary label $y_t \in \{-1, 1\}$, we seek a decision hyperplane $\mathbf{w} \in \mathbb{R}^k$ which best separates data points with distinct labels. However, instead of looking for linear separation in the original space, we seek for linear separation in a sparse coded space. Thus, let $\boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ be the sparse coding of $\mathbf{x}_t$ computed through (1) when using the loss function in (5). We want to find a classifier $\mathbf{w}$ such that $\mathbf{w}^T \boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t) > 0$ when $y_t = 1$ and $\mathbf{w}^T \boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t) < 0$ when $y_t = 0$. This hyperplane need not exist but we can always model the probability of observing $y_t = 1$ through its odds ratio relative to $y_t = -1$. This yields the optimal classifier $\mathbf{w}^*$ as the one that minimizes the logistic loss

$$h(\mathbf{D}, \mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) = \mathrm{P}\left(y_t = \pm 1 \,\middle|\, \boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t), \mathbf{w}\right)$$
$$= \frac{1}{1 + \exp(-y_t \mathbf{w}^T \boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t))} . \tag{6}$$

For a feature vector $\mathbf{x}_t$, (6) models the probability of the label y_t being 1 or -1 as determined by the inner product $\mathbf{w}^T \boldsymbol{\alpha}^*(\mathbf{D}; \mathbf{x}_t)$ through the given logistic transformation. Substituting (6) into (4) yields the discriminative dictionary learning problem for logistic regression with sparse features.

A. Decentralized discriminative learning

We want to solve (4) in distributed settings where signal and observation pairs are independently observed by agents of a network. Agents aim to learn a dictionary and model parameters that are common with all others while having access to local information only. In particular, associated with each agent i is a local random variable and associated output variable $(\mathbf{x}_i, \mathbf{y}_i)$ and each agent's goal is to learn over the aggregate training domain $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$. To this end, let $\mathcal{G} = (V, \mathcal{E})$ be a symmetric and connected network with node set $V = \{1, \dots, N\}$ and $M = |\mathcal{E}|$ directed edges of the form $e = (i, j)$ and further define the neighborhood of i as the set of nodes $n_i := \{j : (i, j) \in \mathcal{E}\}$ that share an edge with i. When each of the N agents observes a pair $(\mathbf{x}_i, \mathbf{y}_i)$, the function h in (4) can be written as a sum of local losses,

$$h(\mathbf{D}, \mathbf{w}; (\mathbf{x}, \mathbf{y})) = \sum_{i=1}^{N} h_i(\mathbf{D}_i, \mathbf{w}_i; (\mathbf{x}_i, \mathbf{y}_i)), \qquad (7)$$

where we have defined the vertically concatenated dictionary $\mathbf{D} := [\mathbf{D}_1; \dots; \mathbf{D}_N] \in \mathbb{R}^{Nm \times k}$ and model parameter $\mathbf{w} := [\mathbf{w}_1; \dots; \mathbf{w}_N] \in \mathbb{R}^{Nk}$.

Substituting (7) into the objective in (4) yields a problem in which the agents learn dictionaries and classifiers that depend on their local observations only. The problem to be formulated here is one in which the agents learn common dictionaries $\mathbf{D}_i = \mathbf{D}_j$ and models $\mathbf{w}_i = \mathbf{w}_j$. 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 (7) into the objective in (4) with these constraints yields the distributed stochastic program

$$\{\mathbf{D}_i^*, \mathbf{w}_i^*\}_{i=1}^N := \operatorname*{argmin}_{\mathbf{D}_i \in \mathcal{D}, \mathbf{w}_i \in \mathcal{W}} \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i, \mathbf{x}_i} \left[h_i(\mathbf{D}_i, \mathbf{w}_i; (\mathbf{x}_i, \mathbf{y}_i)) \right]$$
$$\text{s.t.} \qquad \mathbf{D}_i = \mathbf{D}_j, \ \mathbf{w}_i = \mathbf{w}_j, \ j \in n_i. \qquad (8)$$

When the agreement constraints in (8) are satisfied, the objective is equivalent to one in which all the observations are made at a central location and a single dictionary and model are learnt. Thus, (8) corresponds to a problem in which each agent i, having only observed the local pairs $(\mathbf{x}_i, \mathbf{y}_i)$, aims to learn a dictionary representation and model parameters that are optimal when information is aggregated globally over the network. The decentralized discriminative learning problem is to develop an iterative algorithm that relies on communication with neighbors only so that agent i learns the optimal (common) dictionary $\mathbf{D}_i^* = \mathbf{D}_j^*$ and discriminative model $\mathbf{w}_i^* = \mathbf{w}_j^*$. We present in the following section an algorithm that is shown in Section IV to converge to a local optimum of (8).

III. BLOCK SADDLE POINT METHOD

To write the constraints in (8) more compactly, define the augmented graph edge incidence matrix $\mathbf{C}_D \in \mathbb{R}^{(M \times N)mk}$ associated with the dictionary constraint. 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., $[\mathbf{C}]_{ei} = [\mathbf{C}]_{ej} = \mathbf{0}_{mk}$ when $e \neq (i, j)$. Likewise, the matrix

$\mathbf{C}_w \in \mathbb{R}^{Mk \times Nk}$ is defined by $M \times N$ blocks of dimension k with $[\mathbf{C}_w]_{ei} = \mathbf{I}_k$ and $[\mathbf{C}_w]_{ej} = -\mathbf{I}_k$ when $e = (i, j)$ and $[\mathbf{C}]_{ei} = [\mathbf{C}]_{ej} = \mathbf{0}_k$ otherwise. With these definitions 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

$$\mathbf{C}_D \mathbf{D} = \mathbf{0}, \qquad \mathbf{C}_w \mathbf{w} = \mathbf{0}. \qquad (9)$$

The edge incidence matrices $\mathbf{C}_D$ and $\mathbf{C}_w$ have exactly mk and k null singular values, respectively. We 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 $\mathbf{C}$. The singular values γ and Γ are measures of network connectedness.

Imposing the constraints in (9) for all realizations of the local random variables requires global coordination – indeed, the formulation would be equivalent to the centralized problem in (4). Instead, we consider a modification of (7) in which we add linear penalty terms to incentivize the selection of coordinated actions. Introduce then dual variables $\mathbf{\Lambda}_e = \mathbf{\Lambda}_{ij} \in \mathbb{R}^{m \times k}$ associated with the constraint $\mathbf{D}_i - \mathbf{D}_j = \mathbf{0}$ and consider the addition of penalty terms of the form $\operatorname{tr}[\mathbf{\Lambda}_{ij}^T(\mathbf{D}_i - \mathbf{D}_j)]$. For an edge that starts at node i, the multiplier $\mathbf{\Lambda}_{ij}$ 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 $\mathbf{\Lambda} := [\mathbf{\Lambda}_1; \dots; \mathbf{\Lambda}_M] \in \mathbb{R}^{Mm \times k}$ and $\boldsymbol{\nu} := [\boldsymbol{\nu}_1; \dots; \boldsymbol{\nu}_M] \in \mathbb{R}^{Mk}$ which are restricted to compact convex sets $\mathfrak{L}$ and $\mathcal{N}$, we can write the Lagrangian of this problem as

$$\mathcal{L}(\mathbf{D}, \mathbf{w}, \mathbf{\Lambda}, \boldsymbol{\nu}) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i, \mathbf{x}_i} \left[h_i(\mathbf{D}_i, \mathbf{w}_i; (\mathbf{x}_i, \mathbf{y}_i)) \right]$$
$$+ \operatorname{tr}(\mathbf{\Lambda}^T \mathbf{C}_D \mathbf{D}) + \boldsymbol{\nu}^T \mathbf{C}_w \mathbf{w}. \qquad (10)$$

Recall that in the applications considered here the optimization problem in (8) is not convex. Thus, we use the dual formulation in (10) to develop an iterative distributed algorithm that converges to a KKT point of (8).

To do so, suppose agent i receives local observation pairs $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ at time t and define the instantaneous Lagrangian as the stochastic approximation of (10) evaluated with the observations $\{(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})\}_{i=1}^N$ aggregated across the network as

$$\hat{\mathcal{L}}_t(\mathbf{D}, \mathbf{w}, \mathbf{\Lambda}, \boldsymbol{\nu}) = \sum_{i=1}^{N} h_i(\mathbf{D}_i, \mathbf{w}_i; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}))$$
$$+ \operatorname{tr}(\mathbf{\Lambda}^T \mathbf{C}_D \mathbf{D}) + \boldsymbol{\nu}^T \mathbf{C}_w \mathbf{w}. \qquad (11)$$

We consider the use of the Arrow-Hurwicz saddle point method to solve (8) by alternating block variable updates, in order to exploit the fact that primal-dual stationary pairs are saddle points of the Lagrangian to work through successive primal alternating gradient descent steps and dual gradient ascent steps. Particularized to the Lagrangian in (10), the primal iteration of the saddle point algorithm takes the form

$$\mathbf{D}_{t+1} = \mathbf{D}_t - \epsilon_t \nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t), \qquad (12)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \nabla_{\mathbf{w}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t), \qquad (13)$$

where $\nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$ and $\nabla_{\mathbf{w}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$, are projected stochastic subgradients of the Lagrangian with respect to $\mathbf{D}$ and $\mathbf{w}$, respectively. By projected stochastic subgradients,

we mean stochastic subgradients whose outward normal components to the feasible set are set to null when the iterates are close to the set boundary. Likewise, the dual iteration is defined as

$$\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t + \epsilon_t \nabla_{\mathbf{\Lambda}} \hat{\mathcal{L}}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) , \quad (14)$$

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \epsilon_t \nabla_{\boldsymbol{\nu}} \hat{\mathcal{L}}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) , \quad (15)$$

where $\nabla_{\mathbf{\Lambda}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$ and $\nabla_{\boldsymbol{\nu}} \hat{\mathcal{L}}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$ are the projected stochastic subgradients of the Lagrangian with respect to $\mathbf{\Lambda}$ and $\boldsymbol{\nu}$, respectively. Moreover, ϵ_t is a step size chosen as $O(1/t)$ – see Section IV.

We now show that the algorithm specified by (12)-(15) yields an effective tool for discriminative learning in multi-agent settings.

Proposition 1 *The gradient computations in* (12)-(13) *may be separated along the local primal variables* $\mathbf{D}_{i,t}$ *and* $\mathbf{w}_{i,t}$ *associated with node i, yielding $2N$ parallel updates*

$$\mathbf{D}_{i,t+1} = \mathbf{D}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{D}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}))$$
$$+ \tilde{\mathbf{D}}_{i,t} + \sum_{j \in n_i} (\mathbf{\Lambda}_{ij,t} - \mathbf{\Lambda}_{ji,t}) \Big) , \quad (16)$$

$$\mathbf{w}_{i,t+1} = \mathbf{w}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{w}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}))$$
$$+ \tilde{\mathbf{w}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\nu}_{ij,t} - \boldsymbol{\nu}_{ji,t}) \Big) , \quad (17)$$

where $\tilde{\mathbf{D}}_{i,t}$ and $\tilde{\mathbf{v}}_{i,t}$ are elements of the normal cones of $\mathcal{D}$ and $\mathcal{W}$ when the respective iterates $\mathbf{D}_{i,t}$ and $\mathbf{w}_{i,t}$ are near the boundary of sets $\mathcal{D}$ and $\mathcal{W}$, and null otherwise. Moreover, the dual gradients in the updates of $\mathbf{\Lambda}_{ij,t}$ and $\boldsymbol{\nu}_{ij,t}$ respectively in (14)-(15) *may separated into $2M$ parallel updates associated with edge (i,j)*

$$\mathbf{\Lambda}_{ij,t+1} = \mathbf{\Lambda}_{ij,t} + \epsilon_t \left(\mathbf{D}_{i,t+1} - \mathbf{D}_{j,t+1} + \tilde{\mathbf{\Lambda}}_{ij,t} \right) , \quad (18)$$

$$\boldsymbol{\nu}_{ij,t+1} = \boldsymbol{\nu}_{ij,t} + \epsilon_t \left(\mathbf{w}_{i,t+1} - \mathbf{w}_{j,t+1} + \tilde{\boldsymbol{\nu}}_{ij,t} \right) , \quad (19)$$

which allows for distributed computation across the network.

Proof: See Appendix A. ∎

The D4L algorithm follows by letting node i implement (16)-(17) as we summarize in Algorithm 1. To do so, node i utilizes its local primal iterates $\mathbf{D}_{i,t}$ and $\mathbf{w}_{i,t}$, its local dual iterates $\mathbf{\Lambda}_{ij}$ and $\boldsymbol{\nu}_{ij}$, and its local instantaneous observed pair $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$. Node i also needs access to the neighboring multipliers $\mathbf{\Lambda}_{ji}$ and $\boldsymbol{\nu}_{ji}$ to implement (16) and (17) as well as to the neighboring primal iterates $\mathbf{D}_{j,t}$ and $\mathbf{w}_{j,t}$ to implement (18) and (19). The core steps of D4L in Algorithm 1 are the primal iteration in Step 5 and the dual iteration in Step 7. Steps 4 and 6 refer to the exchange of dual and primal variables that are necessary to implement steps 5 and 7, respectively. Step 1 refers to the acquisition of the signal and observation pair and Step 2 to the computation of the code in (1) using the local current dictionary iterate $\mathbf{D}_{i,t}$. We discuss the specific use of Algorithm 1 to learning discriminative sparse signal representations in a distributed setting to clarify ideas.

Example 3 (Distributed sparse discriminative learning)
Consider a multi-agent system in which signals are independently observed at each agent, and the data domain has latent structure which may be revealed via learning discriminative representations

---

Algorithm 1 D4L: Decentralized Dynamic Discriminative Dictionary Learning

Require: $\mathbf{D}_0$, initial dictionary, $\mathbf{y}_u$, local random variables, $\zeta \in \mathbb{R}$, regularization parameter.
1: **for** $t = 0, 1, 2, \ldots$ **do**
2:     Acquire local signal and observation pair $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$.
3:     Coding [cf. (1)], $\boldsymbol{\alpha}^*_{i,t+1} := \operatorname{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^k} f(\boldsymbol{\alpha}, \mathbf{D}_{i,t}; \mathbf{x}_{i,t})$.
4:     Send $(\mathbf{\Lambda}_{ij,t}, \boldsymbol{\nu}_{ij,t})$ and receive $(\mathbf{\Lambda}_{ji,t}, \boldsymbol{\nu}_{ji,t})$ for all $j \in n_i$.
5:     Update dictionary and model parameters [cf. (16) and (17)]

$$\mathbf{D}_{i,t+1} = \mathbf{D}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{D}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}))$$
$$+ \tilde{\mathbf{D}}_{i,t} + \sum_{j \in n_i} (\mathbf{\Lambda}_{ij,t} - \mathbf{\Lambda}_{ji,t}) \Big) ,$$

$$\mathbf{w}_{i,t+1} = \mathbf{w}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{w}_i} h_i(\mathbf{D}_i, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}))$$
$$+ \tilde{\mathbf{w}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\nu}_{ij,t} - \boldsymbol{\nu}_{ji,t}) \Big) .$$

6:     Send $(\mathbf{D}_{i,t}, \mathbf{w}_{i,t})$ and receive $(\mathbf{D}_{j,t}, \boldsymbol{\nu}_{j,t})$ for all $j \in n_i$.
7:     Update Lagrange multipliers [cf. (18) and (19)]

$$\mathbf{\Lambda}_{ij,t+1} = \mathbf{\Lambda}_{ij,t} + \epsilon_t \left(\mathbf{D}_{i,t+1} - \mathbf{D}_{j,t+1} + \tilde{\mathbf{\Lambda}}_{ij,t} \right) ,$$

$$\boldsymbol{\nu}_{ij,t+1} = \boldsymbol{\nu}_{ij,t} + \epsilon_t \left(\mathbf{w}_{i,t+1} - \mathbf{w}_{j,t+1} + \tilde{\boldsymbol{\nu}}_{ij,t} \right) .$$

8: **end for**

---

that are sparse. In this case, we select the particular form of f in (1) as the elastic net [cf. (5)] with the Euclidean distance $\tilde{f}(\boldsymbol{\alpha}_t, \mathbf{D}; \mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|/2$. Then the dictionary update in (16) may be derived from the subgradient optimality conditions of the elastic-net (see [30]):

$$\mathbf{d}_l(\mathbf{x}_{i,t} - \mathbf{D}\boldsymbol{\alpha}^*) - \zeta_2 \boldsymbol{\alpha}^*_l = \zeta_1 \operatorname{sgn}(\boldsymbol{\alpha}^*_l) \quad \text{if } \boldsymbol{\alpha}^*_l \neq 0 ,$$
$$\mathbf{d}_l(\mathbf{x}_{i,t} - \mathbf{D}\boldsymbol{\alpha}^*) - \zeta_2 \boldsymbol{\alpha}^*_l \leq \zeta_1 \quad \text{otherwise} , \quad (20)$$

where $\operatorname{sgn}(\boldsymbol{\alpha}^*)$ is a vector of signs of $\boldsymbol{\alpha}^*$. Proceeding as in the Appendix of [9], define $Z \subset \{1, \ldots, k\}$ as the set of nonzero entries of $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\mathbf{x}, \mathbf{D})$. Then $\boldsymbol{\alpha}^*$ is the solution to the system of linear inequalities in (20), i.e.

$$\boldsymbol{\alpha}^*_Z = (\mathbf{D}_Z^T \mathbf{D}_Z + \zeta_2 I)^{-1} (\mathbf{D}_Z \mathbf{x} - \zeta_1 \operatorname{sgn}(\boldsymbol{\alpha}^*)) . \quad (21)$$

At time t, to compute the stochastic gradient with of (10) respect to a local dictionary for agent i, apply Proposition 1 of [9] which yields the explicit form

$$\nabla_{\mathbf{D}_i} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t) = -\mathbf{D}_{i,t} \boldsymbol{\beta}_{i,t} \boldsymbol{\alpha}^*_{i,t} + (\mathbf{x}_{i,t} - \mathbf{D}_{i,t} \boldsymbol{\alpha}^*_{i,t}) \boldsymbol{\beta}_{i,t}^T$$
$$+ \tilde{\mathbf{D}}_{i,t} + \sum_{j \in n_i} (\mathbf{\Lambda}_{ij,t} - \mathbf{\Lambda}_{ji,t}) . \quad (22)$$

$\boldsymbol{\alpha}^*_{i,t} = \boldsymbol{\alpha}^*_{i,t}(\mathbf{D}_{i,t}; \mathbf{x}_{i,t})$ is shorthand for (5) and $Z_{i,t}$ is defined as the set of indices associated with nonzero entries of $\boldsymbol{\alpha}^*_{i,t}$. Moreover, we define $\boldsymbol{\beta}^{i,t} \in \mathbb{R}^k$ as

$$\boldsymbol{\beta}^{i,t}_l = ([\mathbf{D}_{i,t}]_l^T [\mathbf{D}_{i,t}]_l + \zeta_2 I)^{-1} \times$$
$$\nabla_{\boldsymbol{\alpha}_l} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t})) \text{ if } l \in Z_{i,t} ,$$
$$\boldsymbol{\beta}^{i,t}_l = 0 \text{ if } l \notin Z_{i,t} , \quad (23)$$

as in [9], Proposition 1. This result is established via a perturbation analysis of the elastic-net optimality conditions. Note that this follows from substituting the solution of (5) into h_i and applying the chain rule in the gradient computation.

IV. CONVERGENCE ANALYSIS

We turn to establishing that the saddle point algorithm in (12)-(15) asymptotically converges to a stationary point of the problem (8). In order to obtain these results, some conditions are required of the network, loss functions, and stochastic approximation errors which we state below.

Assumption 1 (*Network connectivity*) *The network* $\mathcal{G}$ *is connected with diameter* D. *The singular values of the incidence matrix* $\mathbf{C}$ *are respectively upper and lower bounded by* Γ *and* $\gamma > 0$.

Assumption 2 (*Smoothness*) *The Lagrangian has Lipschitz continuous gradients in the primal and dual variables with constants* $L_{\mathbf{D}}$, $L_{\mathbf{w}}$, $L_{\mathbf{\Lambda}}$, *and* $L_{\boldsymbol{\nu}}$. *This implies that, e.g.,*

$$\|\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D}, \mathbf{w}, \mathbf{\Lambda}, \boldsymbol{\nu}) - \nabla_{\mathbf{D}}\mathcal{L}(\tilde{\mathbf{D}}, \mathbf{w}, \mathbf{\Lambda}, \boldsymbol{\nu})\| \leq L_{\mathbf{D}}\|\mathbf{D} - \tilde{\mathbf{D}}\|_F .$$
(24)

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

$$\|\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D}, \mathbf{w}, \mathbf{\Lambda}, \boldsymbol{\nu})\| \leq G_{\mathbf{D}} .$$
(25)

Assumption 3 (*Diminishing step-size rules*) *The step-size* ϵ_t *is chosen as* $O(1/t)$, *i.e.* ϵ_t *satisfies*
(i) $\sum_{t=1}^{\infty} \epsilon_t = \infty$, *(non-summability)*
(ii) $\sum_{t=0}^{\infty} \epsilon_t^2 < \infty$, *(square-summability)*.

Assumption 4 (*Stochastic Approximation Error*) *The stochastic gradients of the Lagrangian are unbiased estimators for the true gradients, which for instance implies*

$$\mathbb{E}\left[\nabla_{\mathbf{D}}\hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\right] = \nabla_{\mathbf{D}}\mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) .$$
(26)

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*

$$\mathbb{E}\left[\|\nabla_{\mathbf{D}}\hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|^2 \,\big|\, \mathcal{F}_t\right] \leq \sigma^2.$$
(27)

Assumption 1 is standard in distributed algorithms and Assumption 2 is common in analysis of descent methods. Assumption 3 specifies that a diminishing step-size condition for the algorithm and Assumption 4 provides conditions on the stochastic approximation errors, both of which are typical in stochastic optimization. They are all satisfied in most cases.

Moreover, the projected stochastic gradients in the updates (12) - (13) implies that the primal variables themselves are contained in compact sets $\mathcal{D}^N$ and $\mathcal{W}^N$, which allows us to write

$$\|\mathbf{D}\|_F \leq \sqrt{Nk} , \qquad \|\mathbf{w}\| \leq K_{\mathbf{w}} ,$$
(28)

for all dictionaries $\mathbf{D} \in \mathcal{D}^N$ and model parameters $\mathbf{w} \in \mathcal{W}^N$. The compactness of dual sets $\mathfrak{L}$ and $\mathcal{N}$ ensure the primal gradients may be bounded in (25), and the dual gradients with respect to $\mathbf{\Lambda}$ and $\boldsymbol{\nu}$ in (25) are bounded with respective constants $G_{\mathbf{\Lambda}} = \Gamma\sqrt{Nk}$ and $G_{\boldsymbol{\nu}} = \Gamma K_{\mathbf{w}}$.

Before stating the main theorem, we present a lemma which will be used in its proof, and appears as Proposition 1.2.4 in [33].

Lemma 1 *Let* $\{a_t\}$ *and* $\{b_t\}$ *be two nonnegative scalar sequences such that* $\sum_{t=1}^{\infty} a_t = \infty$ *and* $\sum_{t=1}^{\infty} a_t b_t < \infty$. *Then*

$$\liminf_{t \to \infty} b_t = 0.$$
(29)

Furthermore, if $|b_{t+1} - b_t| \leq B a_t$ *for some constant* $B > 0$, *then*

$$\lim_{t \to \infty} b_t = 0.$$
(30)

With these preliminary results in place, we are ready to state our main result, which says that the proposed algorithm asymptotically achieves in expectation a first-order stationarity condition of the Lagrangian associated with the optimization problem stated in (8).

Theorem 1 *Denote* $(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$ *as the sequence generated by the block saddle point algorithm in* (12)-(15). *If Assumptions 1 - 4 hold true, then the first-order stationary condition with respect to the primal variables*

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0 ,$$
(31)

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0$$
(32)

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

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\mathbf{\Lambda}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0$$
(33)

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla_{\boldsymbol{\nu}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)\|] = 0$$
(34)

is attained in an expected sense.

Proof: The analysis is broken up into distinct components for the primal and dual variables. In the primal variables, we consider the Lagrangian difference of iterates at the next and current time. We expand terms, use properties of the stochastic gradients and function smoothness, and take conditional expectations on past information to establish a decrement property. We then mirror this analysis in the dual domain. At this point we leverage the step-size rules and apply (29). Then we consider the magnitude of block gradient differences which we bound by a term that diminishes with the step-size, which implies (30) holds, yielding an the expected asymptotic convergence to a stationary solution.

Begin by considering the difference of Lagrangians evaluated at the primal variables at the next and current time, and apply Taylor's Theorem to quadratically approximate the former term

$$\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) - \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$$
$$\leq \left[\begin{array}{c} \nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) \\ \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t) \end{array} \right]^T \left[\begin{array}{c} \mathbf{D}_{t+1} - \mathbf{D}_t \\ \mathbf{w}_{t+1} - \mathbf{w}_t \end{array} \right]$$
$$+ L_{\mathbf{D}}\|\mathbf{D}_{t+1} - \mathbf{D}_t\|_F^2 + L_{\mathbf{w}}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2,$$
(35)

where we have applied the Lipschitz gradient property the Lagrangian to the final two terms as stated in (24) to the last term of (35). The difference of the current and next iterates may be written as $\mathbf{D}_{t+1} - \mathbf{D}_t = -\epsilon_t \nabla_{\mathbf{D}}\hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$ and $\mathbf{w}_{t+1} - \mathbf{w}_t = -\epsilon_t \nabla_{\mathbf{w}}\hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \boldsymbol{\nu}_t)$, which we substitute

into the right hand side of (35), yielding

$$
\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) - \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t)
$$
$$
\leq -\epsilon_t \begin{bmatrix} \nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{bmatrix}^T \begin{bmatrix} \nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{bmatrix}
$$
$$
+ \epsilon_t^2 \Big(L_{\mathbf{D}} \| \nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \|_F^2
$$
$$
+ L_{\mathbf{w}} \| \nabla_{\mathbf{w}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \|^2 \Big), \tag{36}
$$

Take the expectation of (36) conditional on the filtration $\mathcal{F}_t$, and apply Assumption 4 regarding the fact that the stochastic gradients are unbiased estimators of the true gradients [cf. (26)] with finite conditional second moments as stated in (27), so that almost surely we have

$$
\mathbb{E}[\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) - \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \mid \mathcal{F}_t]
$$
$$
\leq -\epsilon_t \left\| \begin{matrix} \nabla_{\mathbf{D}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2 + \sigma^2 (L_{\mathbf{D}} + L_{\mathbf{w}}) , \tag{37}
$$

where we have applied the triangle inequality to the middle term. We set this analysis aside for now and consider Taylor expansion around Lagrangian evaluated at the dual iterates at the next and current time

$$
\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{\nu}_{t+1}) - \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t)
$$
$$
\leq \begin{bmatrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{bmatrix}^T \begin{bmatrix} \mathbf{\Lambda}_{t+1} - \mathbf{\Lambda}_t \\ \mathbf{\nu}_{t+1} - \mathbf{\nu}_t \end{bmatrix} \tag{38}
$$

The difference of the current and next iterates may be written as $\mathbf{\Lambda}_{t+1} - \mathbf{\Lambda}_t = \epsilon_t \nabla_{\mathbf{\Lambda}} \hat{\mathcal{L}}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t)$ and $\mathbf{\nu}_{t+1} - \mathbf{\nu}_t = \epsilon_t \nabla_{\mathbf{\nu}} \hat{\mathcal{L}}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t)$, which we substitute into the right hand side of (38), yielding

$$
\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{\nu}_{t+1}) - \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \tag{39}
$$
$$
\leq \epsilon_t \begin{bmatrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{bmatrix}^T \begin{bmatrix} \nabla_{\mathbf{\Lambda}} \hat{\mathcal{L}}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \hat{\mathcal{L}}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{bmatrix}
$$

Now compute the expectation conditional on the algorithm history $\mathcal{F}_t$ up to time t and apply the unbiased property of the dual stochastic gradient of the Lagrangian [cf. (26)] to write

$$
\mathbb{E}[\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{\nu}_{t+1}) - \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \mid \mathcal{F}_t]
$$
$$
\leq \epsilon_t \left\| \begin{matrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2 \tag{40}
$$

We make use of the convergence of the primal gradient sequence to establish the dual sequence converges by summing the inequalities in (37) and (40) together, which yields

$$
\mathbb{E}[\mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{\nu}_{t+1}) - \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \mid \mathcal{F}_t]
$$
$$
\leq -\epsilon_t \Big(\left\| \begin{matrix} \nabla_{\mathbf{D}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2
$$
$$
+ \left\| \begin{matrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2 \Big) + \sigma^2 (L_{\mathbf{D}} + L_{\mathbf{w}}) , \tag{41}
$$

Observe that the left hand side of (41) is telescopic, and hence if we sum this relations over t we obtain a finite quantity in expectation. Moreover, by applying the step-size rules stated in Assumption 3 with the fact that $\mathcal{L}$ is lower-bounded since the primal and dual domains are compact, the following hold true in expectation

$$
\sum_{t=1}^{\infty} \epsilon_t \mathbb{E}\Big[\left\| \begin{matrix} \nabla_{\mathbf{D}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2
$$
$$
+ \left\| \begin{matrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2 \Big] < \infty , \tag{42}
$$

which as a consequence of (29) as stated in Lemma 1, implies that the expected limit infinum of the sequence converges to null, i.e.

$$
\liminf_{t \to \infty} \mathbb{E}\Big[\left\| \begin{matrix} \nabla_{\mathbf{D}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2
$$
$$
+ \left\| \begin{matrix} \nabla_{\mathbf{\Lambda}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \\ \nabla_{\mathbf{\nu}} \mathcal{L}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \end{matrix} \right\|^2 \Big] = 0 \tag{43}
$$

Using the convergence in (43), we proceed now to establish that the each whole sequence of partial gradients of the Lagrangian converge. Since the logic is equivalent in each block variable, it is enough to consider just the primal-dual pair $(\mathbf{D}, \mathbf{\Lambda})$. We subsequently use the shorthand $\nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} := \nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_t, \mathbf{\nu}_t)$ and $\nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t := \nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t)$. Consider the expected absolute difference of the Lagrangian gradients evaluated at the next and current iterate, and apply Jensen's inequality to write

$$
\Big| \| \mathbb{E}[\nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1}] \|_F^2 - \| \mathbb{E}[\nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t] \|_F^2 \Big|
$$
$$
\leq \mathbb{E}[\| \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} + \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t \|_F
$$
$$
\times \| \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} - \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t \|_F] . \tag{44}
$$

Apply the triangle inequality and Lipschitz continuity to right hand side of (44) to express this gradient difference in terms of the difference between the next and current iterate as

$$
\mathbb{E}[\| \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} + \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t \|_F
$$
$$
\times \| \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} - \nabla_{\mathbf{D}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_t \|_F]
$$
$$
\leq 2 G_{\mathbf{D}} L_{\mathbf{D}} \mathbb{E}[\| \mathbf{D}_{t+1} - \mathbf{D}_t \|] , \tag{45}
$$

where we have used the gradient bounds and Lipschitz gradient properties [cf. (25) and (24)] in the last inequality to bound the magnitude of $\mathbf{D}_t$. Substitute $\mathbf{D}_{t+1} - \mathbf{D}_t = -\epsilon_t \nabla_{\mathbf{D}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t)$ into the previous expression and apply with the bound on the second conditional moment of the stochastic gradient stated in (27) of Assumption 4 to write

$$
2 G_{\mathbf{D}} L_{\mathbf{D}} \epsilon_t \mathbb{E}[\| \nabla_{\mathbf{w}} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \|] \leq 2 G_{\mathbf{D}} L_{\mathbf{D}} \sigma \epsilon_t . \tag{46}
$$

With (46), the second condition of Lemma 1 is satisfied, whereby we may conclude the gradient sequence converges in expectation

$$
\lim_{t \to \infty} \mathbb{E}[\| \nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \|] = 0 . \tag{47}
$$

Since the other primal block sequence $\mathbf{w}_t$ is updated in an analogous manner to that of $\mathbf{D}$, the analysis with the same logic, implying that a first order stationary condition of the Lagrangian is achieved asymptotically almost surely, i.e. $\mathbb{E}[\| \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{D}_t, \mathbf{w}_t, \mathbf{\Lambda}_t, \mathbf{\nu}_t) \|] \to 0$.

We now establish that the whole dual gradient sequence with respect to $\mathbf{\Lambda}$ is converging in magnitude to null. We use the shorthand notation $\nabla_{\mathbf{\Lambda}} \mathcal{L}(\cdot, \cdot, \cdot, \cdot)_{t+1} := \nabla_{\mathbf{\Lambda}} \mathcal{L}(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{\nu}_{t+1})$ and

$\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_t := \nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\mathbf{D}_{t+1},\mathbf{w}_{t+1},\boldsymbol{\Lambda}_t,\boldsymbol{\nu}_t)$. Continue by considering the expected absolute difference of the Lagrangian gradients evaluated at the next and current dual iterate, and applying Jensen's inequality to write

$$\left| \|\mathbb{E}[\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_{t+1}]\|_F^2 - \|\mathbb{E}[\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_t]\|_F^2 \right|$$
$$\leq \mathbb{E}[\|\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_{t+1} + \nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_t\|_F$$
$$\times \|\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_{t+1} - \nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\cdot,\cdot,\cdot,\cdot)_t\|_F]$$
$$= \mathbb{E}[\|\mathbf{C}_{\mathbf{D}}\mathbf{D}_{t+1} + \mathbf{C}_{\mathbf{D}}\mathbf{D}_t\|_F \|\mathbf{C}_{\mathbf{D}}\mathbf{D}_{t+1} - \mathbf{C}_{\mathbf{D}}\mathbf{D}_t\|_F] \quad (48)$$

where the last equality in (48) follows from the computation of the dual gradient of the Lagrangian in (58). Now apply the triangle inequality and the compactness of the set $\mathcal{D}$ to express the right hand side of (48) in terms of the difference between the next and current iterate, yielding

$$\mathbb{E}[\|\mathbf{C}_{\mathbf{D}}\mathbf{D}_{t+1} + \mathbf{C}_{\mathbf{D}}\mathbf{D}_t\|_F \|\mathbf{C}_{\mathbf{D}}\mathbf{D}_{t+1} - \mathbf{C}_{\mathbf{D}}\mathbf{D}_t\|_F]$$
$$\leq 2\Gamma\sqrt{Nk}\mathbb{E}[\|\mathbf{D}_{t+1} - \mathbf{D}_t\|] , \quad (49)$$
$$= 2\Gamma\sqrt{Nk}\epsilon_t\mathbb{E}[\|\nabla_{\mathbf{D}}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\boldsymbol{\Lambda}_t,\boldsymbol{\nu}_t)\|] \leq 2\Gamma\sqrt{Nk}\sigma\epsilon_t ,$$

where the second equality comes from the substitution $\mathbf{D}_{t+1} - \mathbf{D}_t = -\epsilon_t\nabla_{\mathbf{D}}\hat{\mathcal{L}}(\mathbf{D}_t,\mathbf{w}_t,\boldsymbol{\Lambda}_t,\boldsymbol{\nu}_t)$, and the last inequality comes from applying the bound in (27). As in the analysis of the primal gradient sequence, we may now apply (30) in Lemma 1, which implies that the expected dual gradient sequence converges to null in magnitude, i.e.

$$\lim_{t\to\infty} \mathbb{E}[\|\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\mathbf{D}_t,\mathbf{w}_t,\boldsymbol{\Lambda}_t,\boldsymbol{\nu}_t)\|] = 0. \quad (50)$$

By noting that the analysis for $\boldsymbol{\Lambda}$ is analogous to that of the other dual variable $\boldsymbol{\nu}$, we may also conclude $\mathbb{E}[\|\nabla_{\boldsymbol{\nu}}\mathcal{L}(\mathbf{D}_t,\mathbf{w}_t,\boldsymbol{\Lambda}_t,\boldsymbol{\nu}_t)\|] \to 0$. ∎

Theorem 1 guarantees that the block saddle point method as stated in (12) - (15) 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, the algorithm asymptotically converges to a KKT point of the problem (8) in expectation. This implies that the primal variables converge to a local minimum of the objective, and the dual variables asymptotically enforce feasibility, i.e. the network agreement constraints are satisfied in expectation. We next turn to the practical consequences of this theorem by studying the algorithm performance on a canonical computer vision task.

V. SIMULATIONS

We are chiefly concerned with the task of recognizing visual patterns in dynamic multi-agent settings. In this setting, learning sparse representations have recently yielded key insights into the data domain [8], motivating our choice of selection of f in (1) as a sparse coding problem via the elastic-net (5) with the Euclidean distance $\tilde{f}(\boldsymbol{\alpha}_t,\mathbf{D};\mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|/2$. Using this loss, which can be efficiently computed via least angle regression [35], we study the performance of Algorithm 1 on a multi-class texture classification problem in order to understand its empirical performance in a simple setting.

To this end, we conduct numerical experiments on the Brodatz dataset [34] 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. $N = 1$.
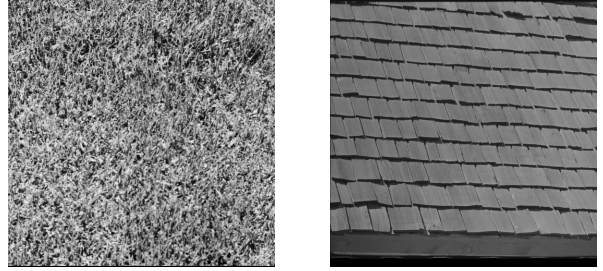

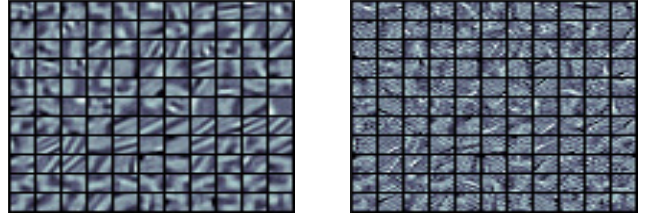Fig. 1: Sample images from the Brodatz texture database.

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 $\epsilon = 0.25$.

Moreover, we consider the case where each agent observes training examples which are incomplete random subsets of the total class labels, yet aims to learn a classifier over all possible classes. We use the numerical insight gained in this section as design principles for the robotic experiments in Section VI.

We restrict ourselves to $C = 4$ class labels {*grass, bark, straw, herringbone_weave* } in the Brodatz texture database. Sample images from this data set are shown in Figure 1. This data subset consists of one grayscale image per texture, which amounts to four 512×512 images in total consisting of $1,820$ overlapping patches of size 24×24.

A. Feature Generation

Inspired by the two-dimensional texton features discussed in [36], we generate texture features to classify, $\tilde{\boldsymbol{\alpha}}^*$, as the sum of the sparse dictionary representations of sub-patches. That is, we classify each image patch of size 24-by-24 by first extracting the nine non-overlapping 8-by-8 sub-patches $\mathbf{x}^{(i)}$ within it. We vectorize (column-major order) each sub-patch, subtract off the sample mean, and divide by its norm such that ℓ_2 norm so that $\mathbf{x}^{(i)}$ is zero-mean and has unit ℓ_2 norm. Stacking these sub-patches column wise in a matrix $\mathbf{X} = [\mathbf{x}^{(1)}; \cdots ; \mathbf{x}^{(9)}]$, we compute the aggregate sparse coding $\tilde{\boldsymbol{\alpha}}^*_{i,t}$ at agent i at time t according to

$$\tilde{\boldsymbol{\alpha}}^*(\mathbf{X}_{i,t},\mathbf{D}_{i,t}) = \sum_{l=1}^{9} \boldsymbol{\alpha}^*(\mathbf{D}_{i,t};\mathbf{x}_{i,t}^{(l)}) , \quad (51)$$

which implies that the local stochastic gradient of the dictionary $\nabla_{\mathbf{D}_i}\tilde{h}_{i,t}(\mathbf{D}_{i,t},\mathbf{w}_{i,t};(\mathbf{X}_{i,t},\mathbf{y}_{i,t}))$ is the sum of contributions $\nabla_{\mathbf{D}_i}h_{i,t}(\mathbf{D}_{i,t},\mathbf{w}_{i,t};(\mathbf{x}_{i,t},\mathbf{y}_{i,t}))$ from each sub-patch.

B. Loss Function and Performance Metrics

We cast texture classification as a multi-class logistic regression problem in which agent i receives signals $\mathbf{x}_i$ and is charged with outputting a decision variable $\mathbf{y}_i, \in \{0,1\}^C$ where C is the number of classes. Each component $y_{i,c}$ of the vector

$\mathbf{y}_i \in \{0,1\}^C$ is a binary indicator of whether the signal falls into class c. The local loss h_i is the negative log-likelihood of the probabilistic model [37], stated as

$$h_i(\mathbf{D}_i, \mathbf{W}_i; (\mathbf{X}_i, \mathbf{y}_i)) = \log\left(\sum_{c=1}^C e^{\mathbf{w}_{i,c}^T \tilde{\boldsymbol{\alpha}}_i^* + w_{i,c}^0}\right) \tag{52}$$
$$- \sum_{c=1}^C \left(y_{i,c} \mathbf{w}_{i,c}^T \tilde{\boldsymbol{\alpha}}_i^* + w_{i,c}^0\right) + \xi \|\mathbf{W}_i\|_F^2,$$

where the activation functions $g_c(\tilde{\boldsymbol{\alpha}}_i^*) = e^{\mathbf{w}_{i,c}^T \tilde{\boldsymbol{\alpha}}_i^* + w_{i,c}^0}$, are computed using the cth column $\mathbf{w}_c$ of the weight matrix $\mathbf{W}_i \in \mathbb{R}^{(k+1) \times C}$. Moreover, $w_{i,c}^0$ is a bias term for each class c. Recall that h_i is implicitly a function of $\mathbf{D}_i$ since the aggregate sparse coding $\tilde{\boldsymbol{\alpha}}_i^*$ depends on the dictionary over which it is represented. Moreover, we add l_2 regularization to prevent over-fitting.

To ensure identifiability, every element of the last column of $\mathbf{W}_i$ is set to zero. With $\mathbf{W}_i$, the probability that $\tilde{\boldsymbol{\alpha}}_i^*$ belongs to class c is given by $g_c(\tilde{\boldsymbol{\alpha}}_i^*)/\sum_{c'} g_{c'}(\tilde{\boldsymbol{\alpha}}_i^*)$, the classification decision is made by selecting the maximum-likelihood class label, i.e. $\tilde{c} = \arg\max_c g_c(\tilde{\boldsymbol{\alpha}}_i^*)/\sum_{c'} g_{c'}(\tilde{\boldsymbol{\alpha}}_i^*)$ is the only nonzero entry of $\mathbf{y}_{i,t}$.

Besides the local loss h_i which we known converges to a KKT point 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 $\tilde{T} = 4096$. We also consider the relative variation of the average classifiers, stated as

$$\text{RV}(\bar{\mathbf{W}}_{i,t}) = \frac{1}{N} \sum_{j=1}^N \|\bar{\mathbf{W}}_{i,t} - \bar{\mathbf{W}}_{j,t}\|_F, \tag{53}$$

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 $\mathbf{W}^*$ and agreement between estimates of different agents is difficult to visualize.

C. Implementation Details

(i) Dictionary Size To select a dictionary of the appropriate size, we investigate its effect on classification accuracy on the Brodatz textures. Because the number of atoms k in the dictionary will similarly affect both the centralized and decentralized algorithms, we conduct out this experiment for the centralized ($N = 1$) algorithm due to computational expedience. As in [9], we find that increasing the size of the dictionary led to better classifier performance. However, because of diminishing performance improvement with increasing k beyond a certain level, we select $k = 128$ in all subsequent experiments. In theory, one could make k arbitrarily large, but as k increases so do the computational demands of the algorithm. We show the initialized and final 128-element, 8-by-8 patch dictionaries in Figure 2.

(ii) Mini-Batching In our implementation of D4L, we adopted a mini-batching procedure: at each iteration, we replace the single labeled patch with a small batch of $\hat{T} = 200$ randomly-drawn labeled patches. The procedure for generating this batch is as follows: for each patch, a label is first drawn uniformly at random from the set of all possible labels. Then, the patch is selected uniformly at random from the set of all patches with that label.

We then compute the dictionary and classifier gradient values for the iteration by averaging the gradient values generated by each individual patch within the mini-batch. Practically, this process reduces the variance of the local stochastic gradients, which often empirically yields improved convergence.

(iii) Initialization We initialized $\mathbf{D}$ using unsupervised dictionary learning for a small set of randomly-drawn initialization data [16]. We then used the labels and the dictionary representations of the data to initialize the classifier parameters $\mathbf{W}$. All experiments are run from this common initialization.

(iv) Regularization and Step-size Selection The D4L algorithm requires several parameters to be specified. Following [9], we used $\zeta_1 = 0.125$, $\zeta_2 = 0$, $\xi = 10^{-9}$. We also adopted the learning-rate selection strategy discussed in [9], which is to select the initial step-size ϵ by implementing a grid search over a fixed small number of iterations ($\tilde{T} = 200$) and selecting the one that minimized cross-validation error. We set $\epsilon_t = \min(\epsilon, \epsilon t_0/t)$, where $t_0 = T/2$. This selection amounts to using a step-size of ϵ for the first half of the iterations after which we follow a $1/t$ annealing rate for those that remain, enforcing convergence.

We note here that, due to the non-convexity of the objective, the algorithm may diverge if ϵ is too large. This follows from the fact that convergence guarantees for stochastic gradient algorithms in non-convex settings only occur in cases where a diminishing step-size mitigates the stochastic approximation error, which may not occur if the step-size is too large. Moreover, we have experimentally observed that values of ϵ which avoid this behavior are smaller than effective values for the centralized version by an order of magnitude or more. Consequently, when comparing D4L to its centralized counterpart, we select ϵ that yield convergence for both settings, i.e., the smaller values appropriate for D4L. For the Brodatz dataset, we found that $\epsilon = 0.05$ led to convergence in all cases.

D. Results on Texture Database

(i) Network Size To investigate the dependence of the convergence rate in Theorem 1 on the network size N we run Algorithm 1 for problem instances with $N = 1$ (centralized), $N = 10$, and $N = 100$ nodes. For the later two cases, connections 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. In Figure 3a, we show $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ over iteration t. Observe that as N increases, the log-likelihood $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ 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 $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t})) \leq 0.62$ by $T = 1000$, 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. Observe that 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 $T = 1000$ iterations, respectively.
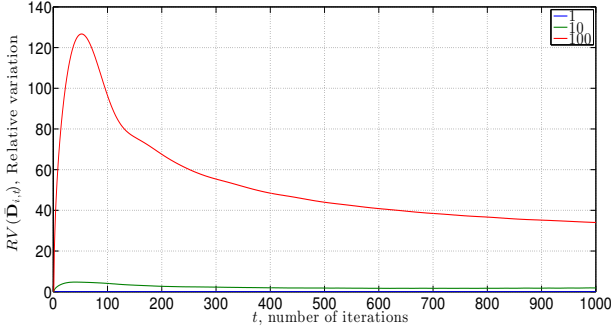
In Figure 3c we investigate how far the agents are from consensus as measured by $\text{RV}(\bar{\mathbf{D}}_{i,t})$ over iteration t. Observe
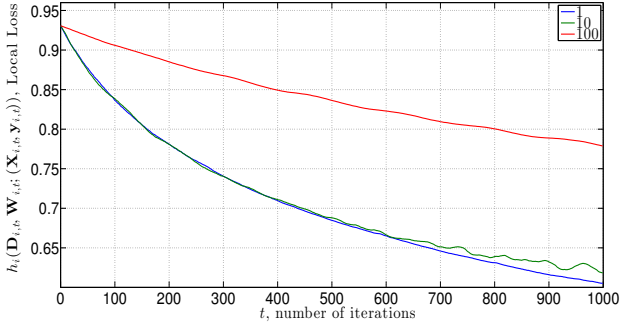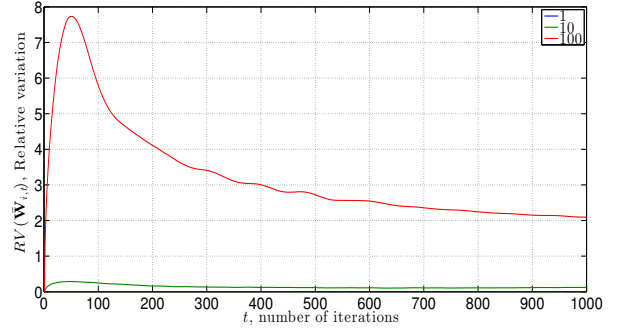
(a) Local loss versus iteration t
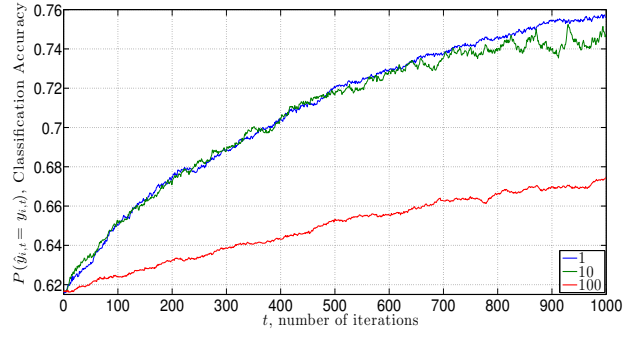
(b) Classification accuracy vs. iteration t

(c) Relative variation $\mathrm{RV}(\bar{\mathbf{D}}_{i,t})$ vs. iteration t

(d) 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 $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ and $\sum_{i=1}^{N} P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})/N$ versus iteration t, both of which decline faster in smaller networks. Figures 3c-3d show that network disagreement in terms of $\mathrm{RV}(\bar{\mathbf{D}}_{i,t})$ and $\mathrm{RV}(\bar{\mathbf{W}}_{i,t})$ becomes more stable and declines faster with smaller N. Algorithm performance in moderate sized networks is comparable to the centralized case.

that the network experiences greater disagreement, as measured with $\mathrm{RV}(\bar{\mathbf{D}}_{i,t})$, with larger N. Trivially $\mathrm{RV}(\bar{\mathbf{D}}_{i,t}) = 0$ for the centralized case, but for the $N = 10$ and $N = 100$ node networks the algorithm achieves $\mathrm{RV}(\bar{\mathbf{D}}_{i,t}) \leq 1.9$ by $t \geq 417$ and $\mathrm{RV}(\bar{\mathbf{D}}_{i,t}) \leq 35$ by $T = 1000$ respectively. Thus in larger networks information diffuses more slowly. Moreover, the agreement constraints are more difficult to satisfy and delay the convergence to stationarity. This difference in consensus achievement may also be observed in Figure 3d, which shows $\mathrm{RV}(\bar{\mathbf{W}}_{i,t})$ over iteration t. That is, for the $N = 10$ and $N = 100$ node networks the algorithm achieves $\mathrm{RV}(\bar{\mathbf{W}}_{i,t}) \leq 1.3 \times 10^{-1}$ by $t \geq 312$ and $\mathrm{RV}(\bar{\mathbf{W}}_{i,t}) \leq 3$ for $t \geq 400$ respectively. Thus we observe an order of magnitude difference in the relative variation between the $N = 10$ and $N = 100$ node networks for both the dictionary and model parameters.

(ii) Network Topology and Diameter We study the dependence of the convergence rate of Algorithm 1 in Theorem 1 on the network topology by fixing the network size to $N = 20$ and running (12) - (15) over random graphs, small world graphs, cycles, and grids. In the first two, the probability that node pairs are randomly connected is fixed at $\rho = 0.2$. The latter two are deterministically generated. A cycle is a closed directed chain of nodes. Grids are formed by taking the two-dimensional integer lattice of size $\sqrt{N} \times \sqrt{N}$, with $\sqrt{N}$ rounded to the nearest integer. Connections are drawn between adjacent nodes in the lattice as well as between remainder nodes at the boundary. Cycles, grids and random networks have progressively larger number of connections per node and smaller diameter. Random networks

have small degree and small diameter; see [38], [39].

We present the results of this experiment in Figure 4. In Figure 4a, we plot $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ over iteration t. Observe that the rate at which $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ decreases is comparable across the different networks, yet we can differentiate the learning achieved in the different settings by the benchmark $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t})) \leq 6.5 \times 10^{-1}$. To surpass this bound, the algorithm requires $t = 700$ iterations for grids and cycles, whereas the random and small world networks oscillate around the interval $[0.65, 0.70]$ for $t \geq 650$. This indicates that structured deterministic networks are an easier setting for finding good signal representations in a decentralized manner. This insight is corroborated in the plot of classification performance over iteration t in Figure 4b. Observe that the algorithm achieves an accuracy near 75% for cycle and grid networks as compared to 72% and 73% for random and small world networks, respectively.

We study how the network topology impacts the algorithm's convergence to consensus in 4c, where we plot $\mathrm{RV}(\bar{\mathbf{D}}_{i,t})$ over time t. Observe that the initial burn-in period is comparable across the different networks except for the cycle, i.e. $\mathrm{RV}(\bar{\mathbf{D}}_{i,t}) \leq 7$ for $t \geq 300$ the grid, small world, and random networks as compared to $\mathrm{RV}(\bar{\mathbf{D}}_{i,t}) \leq 10$ for the cycle setting. However, by $t \geq 600$, the random and small world networks experience greater numerical oscillations. This difference in convergence to consensus as measured my the relative variation is corroborated in 4d, where we plot $\mathrm{RV}(\bar{\mathbf{W}}_{i,t})$ versus iteration t. Observe that the cycle yields the slowest convergence rate, yet is more stable than the small world and random networks. Surprisingly, the grid
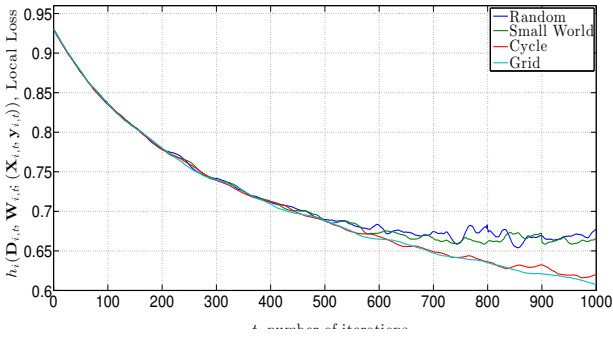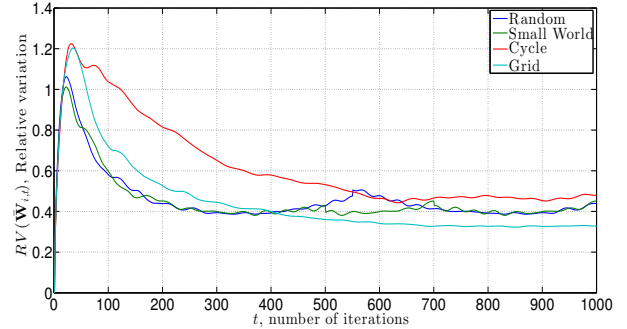
(a) Local loss versus iteration t
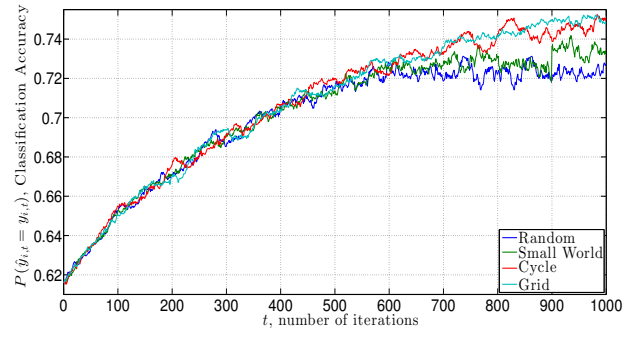
(b) Classification accuracy vs. iteration t
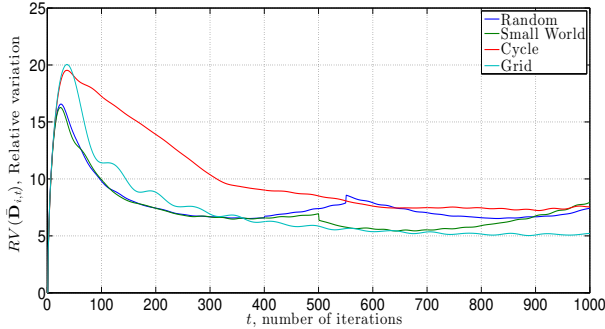
(c) Relative variation $RV(\bar{\mathbf{D}}_{i,t})$ vs. iteration t

(d) Relative variation $RV(\bar{\mathbf{W}}_{i,t})$ vs. iteration t

Fig. 4: D4L run on $N = 20$ node cycle, grid, random and small world networks, where edges are generated randomly between agents with probability $\rho = 0.2$ in the later two. Figure 4a-4b show $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ and $\sum_{i=1}^{N} P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})/N$, respectively, over iteration t for an arbitrarily chosen agent $i \in V$. Learning slows and numerical oscillations become more prevalent in networks with random connectivity patterns. Structured deterministic networks such as grids and cycles have larger diameter than small world and random networks, yet achieve superior performance. Figures 4c-4d shows that the agents reach consensus in terms of $RV(\bar{\mathbf{D}}_{i,t})$ and $RV(\bar{\mathbf{W}}_{i,t})$ at comparable rates across the different network topologies.

network has superior convergence to consensus both in terms of dictionary and model parameters.

(iv) Complete vs. Incomplete Sampling We now turn to studying the performance of D4L in the more challenging context of what we term *incomplete sampling*, which refers to the case that each agent in the network observes only training examples from a fixed random subset of the total number of class labels, yet is charged with the task of learning a classifier which can identify all classes. This problem setting is closer to our robotic application of interest, which is discussed in further detail in Section VI. We run the algorithm on a $N = 10$ node random network with connection probability $\rho = 0.2$. We display these results in Figure 5, where we also plot for comparison results generated in the complete sampling setting.

Figure 5 shows the results of this numerical experiment for a randomly selected agent in the network. In Figure 5a, we plot $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ over iteration t. Note that during an initial burn-in period of $t \le 300$ the local losses decline at comparable rates, after which the algorithm experiences greater numerical oscillations and its convergence rate slows for the incomplete case. In particular, for the benchmark $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t})) \le 6.7 \times 10^{-1}$, the incomplete sampling case requires $t = 676$ iterations as compared with $t = 592$ for the complete sampling case. This oscillatory behavior and reduced convergence rate may also be observed in the plot of classification performance versus iteration t in Figure 5b. Observe that by $t = 350$ both cases achieve an accuracy of 70%; however, the incomplete sampling oscillates around this benchmark whereas

the complete case continues to improve. Succinctly, increased dithering around stationarity occurs when agents observe training examples drawn from incomplete subsets of the total labels.

We study how the incomplete sampling, or the implicit partition of the feature space across the network, impacts the network disagreement in Figure 5c, where we plot $RV(\bar{\mathbf{D}}_{i,t})$ over time t. Observe during an initial burn-in period of $t \le 100$ that the relative variation is smaller in the case of incomplete sampling for the dictionary, yet by $t \ge 600$ the complete sampling case more closes enforces consensus. Moreover, $RV(\bar{\mathbf{D}}_{i,t})$ slowly climbs despite the convergence of the algorithm to a neighborhood of a stationary point. We observe a similar phenomenon in the plot of $RV(\bar{\mathbf{W}}_{i,t})$ over time t in Figure 5d, i.e. for $t \ge$, $RV(\bar{\mathbf{W}}_{i,t}) \le 5 \times 10^{-1}$ for the incomplete sampling case.

VI. ROBOTIC EXPERIMENTS

In this section, we consider how the tools developed in Section III may be used for a team of robots to perform collaborative pattern recognition in unknown domains. This task may be viewed as providing the capability to perform object recognition or analyze the navigability of the environment in which the multi-agent system operates. To this end, we have deployed a robotic network in a cluttered urban setting with little a priori information and recorded visual information along distinct paths for each robot. Note that each robot in the network only has access to information about the environment based on the path it has traversed, which may omit regions of the feature space crucial for recognizing patterns. By communicating with other robots
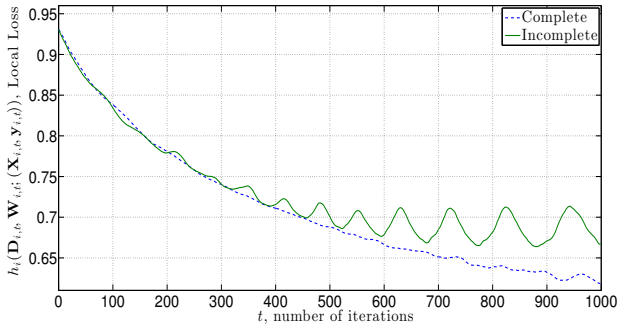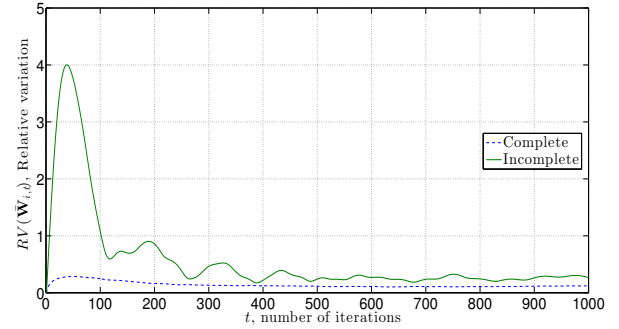
(a) Local loss versus iteration t
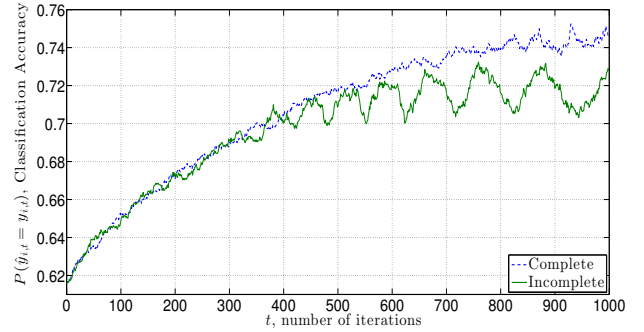
(b) Classification accuracy vs. iteration t
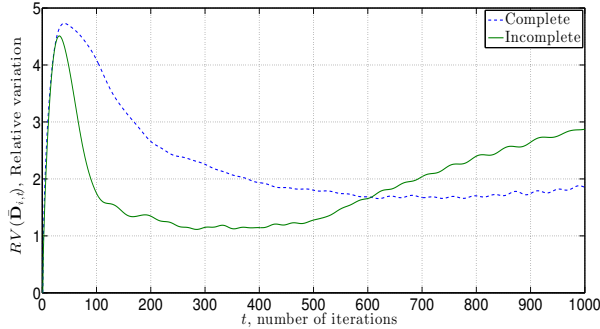
(c) Relative variation $RV(\bar{\mathbf{D}}_{i,t})$ vs. iteration t

(d) Relative variation $RV(\bar{\mathbf{W}}_{i,t})$ vs. iteration t

Fig. 5: Algorithm 1 run on a $N = 10$ node random networks, where edges are generated randomly between agents with probability $\rho = 0.2$. "Incomplete" refers to the case where each agent observes training examples that comprise a random incomplete subset of the total data labels. Figure 4a-4b show $h_i(\mathbf{D}_{i,t}, \mathbf{W}_{i,t}; (\mathbf{X}_{i,t}, \mathbf{y}_{i,t}))$ and $\sum_{i=1}^{N} P(\hat{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t})/N$, respectively, over iteration t for an arbitrarily chosen agent $j \in V$, as compared with the case where this agent observes training examples from all classes. Observe that learning is still achieved for this more challenging context, yet the algorithm exhibits increased oscillatory behavior and decreased accuracy for the incomplete case. Figures 4c-4d shows that the agents reach consensus in terms of $RV(\bar{\mathbf{D}}_{i,t})$ and $RV(\bar{\mathbf{W}}_{i,t})$ at comparable rates for complete cases and incomplete cases, albeit with more oscillations in the later. Moreover, the algorithm still converges despite the instability in $RV(\bar{\mathbf{D}}_{i,t})$.

in the network, individuals may learn over a broader domain associated with that which has been explored by the whole mobile network.

This problem is challenging due to the reliance of many computer vision algorithms on static data and centralized processing. Recent works have made progress towards online object recognition [40], [41], yet solving this task in dynamic distributed settings remains an open problem. As a preliminary benchmark towards this goal, we consider decentralized online texture classification in teams of robots. In this case, each robot in the network sequentially observes images based on its traveled path, partitions them into small patches, and identifies properties of each patch. We interpret this problem setting as providing multi-robot systems the means to dynamically analyze the navigability of unknown domains: certain textures correspond to environments which are easier to traverse than others, and discerns the intractable from the traversable.

Experimentally, we consider cases where each robot observes training examples which contain all of the class types, and is able to communicate with all others in the network. We collected images that were sequentially observed by a $N = 3$ agent network of Husky robots at Camp Lejeune, a cluttered urban setting, and labeled the images offline. See Figure 6 for an example of the images taken by a prototypical Husky platform. We call this data the Integrated Research Assessment for the U.S. Army's Robotics Collaborative Technology Alliance, abbreviated as IRA. The associated texture classes for the IRA field data are {*sky,*
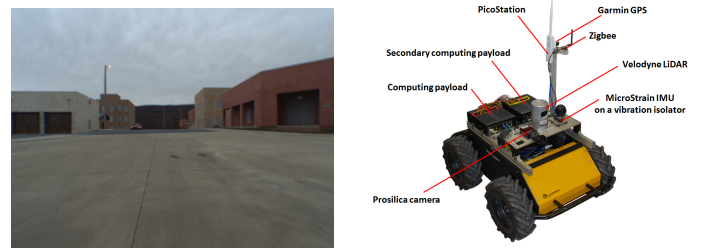

Fig. 6: Sample image from the IRA dataset (left) associated with an $N = 3$ node network of Husky robots (right) moving around a cluttered urban setting.

grass, building, concrete_ floor}. Running the D4L algorithm on these image observations resembles a field implementation of a robotic network. The IRA dataset consists of 16 images converted to grayscale of size 320 by 240. Using the human-generated label masks, we are able to extract $610,528$ label-homogeneous, overlapping 24-by-24 patches that have labels within our subset. Values of ϵ that exhibit convergent behavior varied between the two problem settings (Brodatz vs. IRA). In the IRA setting, we select $\epsilon = 5 \times 10^{-3}$.

Results for the implementation on a $N = 3$ robotic network associated with the IRA dataset as compared with the centralized processing of each robot's data in aggregate is shown in Figure 7. Due to the more challenging nature of field data, reducing
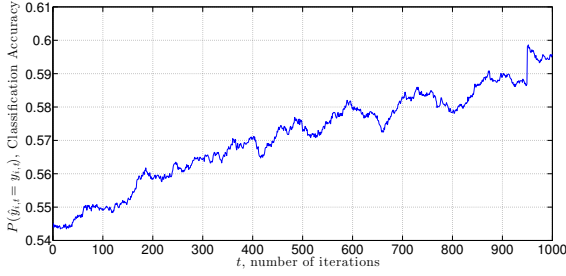
Fig. 7: D4L classifier performance improvement for the IRA texture dataset with step-size $\epsilon = 5 \times 10^{-3}$. "D4L, Complete" refers to a fully-connected, three-node network where each node has access to training data from all classes. Classifier accuracy is averaged over all robots. Increasing the step-size ϵ may improve performance faster, but may yield divergence in the decentralized cause. The field setting is more challenging for pattern recognition, yet the decentralized and centralized algorithms still achieve comparable performance.

the algorithm step-size to $\epsilon = 5 \times 10^{-3}$ to ensure convergence was necessary. However, such a small step-size makes learning occur at a slow rate. We do observe that learning occurs, albeit slowly: the accuracy of the algorithm continues to climb as more data is accumulated, though by $T = 10^3$ classification accuracy approaches %60.

VII. CONCLUSION

This work represents the first attempt to extend the discriminative dictionary learning problem of [9] to networked settings. To do so we consider cases where losses are node-separable and introduced agreement constraints, yielding a decentralized stochastic non-convex optimization problem. By considering the Lagranian relaxation of an agreement-constrained system, we develop a block variant of the Arrow-Hurwicz saddle point method to solve it. Moreover, we establish the convergence of the algorithm to a KKT point of the problem in expectation.

Experiments on a texture classification problem demonstrated comparable classifier performance between the centralized and decentralized settings, and illustrated the convergence rate dependence on the network topology. Learning occurred at slower rates in larger networks and networks with random connectivity patterns. Moreover, the proposed method allows multi-agent systems to learn over a new class of pattern recognition problems. The small step-sizes required for convergence yielded slow learning rates, yet are essential for avoiding divergent behavior in non-convex settings. While pattern recognition methods for centralized settings [9], [36], [42] outperform that which is developed here, they may not be readily adapted to distributed settings. We applied this method to a mobile robotic network deployed in an unknown domain charged with the task of collaboratively analyzing the navigability of distinct paths traversed by each robot. In doing so, we gained insight into how to more effectively design collaboration strategies in robotic teams for future hardware development.

VIII. ACKNOWLEDGEMENTS

To compute the primal stochastic gradient of the Lagrangian [cf. (10)] with respect to a local dictionary $\mathbf{D}_{i,t}$ for a signal-output pair $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$, apply the node-separability of the global cost in (7) to the first term in (10), and note that all terms of the derivative of the second term with respect to $\mathbf{D}_i$ in (10) are null except those associated with node i and neighbors j. With this reasoning we obtain

$$\nabla_{\mathbf{D}_i} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t) = \nabla_{\mathbf{D}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t})) + \tilde{\mathbf{D}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\Lambda}_{ij,t} - \boldsymbol{\Lambda}_{ji,t}) \quad (54)$$

Here $\tilde{\mathbf{D}}_{i,t}$ is an element of the normal cone of $\mathcal{D}$ when $\mathbf{D}_{i,t}$ is near the boundary of the set, and null otherwise. comes from the fact that we are using projected stochastic gradients rather than stochastic gradients. Substituting this into the update given in (12) and separating update along direction associated with agent i, we obtain

$$\mathbf{D}_{i,t+1} = \mathbf{D}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{D}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t})) + \tilde{\mathbf{D}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\Lambda}_{ij,t} - \boldsymbol{\Lambda}_{ji,t}) \Big), \quad (55)$$

To compute the primal stochastic gradient of (10) with respect to the model parameters $\mathbf{w}_i$ of agent i, we apply the same logic to terms involving $\mathbf{w}_i$ to write

$$\nabla_{\mathbf{w}_i} \hat{\mathcal{L}}_t(\mathbf{D}_t, \mathbf{w}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t) = \nabla_{\mathbf{w}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t})) + \tilde{\mathbf{w}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\nu}_{ij,t} - \boldsymbol{\nu}_{ji,t}) \quad (56)$$

Again, the stochastic gradient of agent i is decoupled from all others except the Lagrange multipliers it receives from its neighbors, and hence yields a decentralized implementation. The particular form of the model parameter update follows

$$\mathbf{w}_{i,t+1} = \mathbf{w}_{i,t} - \epsilon_t \Big(\nabla_{\mathbf{w}_i} h_i(\mathbf{D}_{i,t}, \mathbf{w}_{i,t}; (\mathbf{x}_{i,t}, \mathbf{y}_{i,t})) + \tilde{\mathbf{w}}_{i,t} + \sum_{j \in n_i} (\boldsymbol{\nu}_{ij,t} - \boldsymbol{\nu}_{ji,t}) \Big). \quad (57)$$

To develop the dual variable updates, compute the stochastic subgradient of (10) with respect to the Lagrange multipliers associated with edge (i, j) and the dictionary agreement constraint. By noting that all terms in the sum $\text{tr}(\boldsymbol{\Lambda}^T \mathbf{C} \mathbf{D})$ are null except those associated with edge (i, j), we obtain

$$\nabla_{\boldsymbol{\Lambda}_{ij}} \hat{\mathcal{L}}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t) = \mathbf{D}_{i,t+1} - \mathbf{D}_{j,t+1} + \tilde{\boldsymbol{\Lambda}}_{ij,t}. \quad (58)$$

This local subgradient corresponds to the communication link between agent i and agent j. Separating the update in (15) along variables associated with edge (i, j), we obtain the local update

$$\boldsymbol{\Lambda}_{ij,t+1} = \boldsymbol{\Lambda}_{ij,t} + \epsilon_t \left(\mathbf{D}_{i,t+1} - \mathbf{D}_{j,t+1} + \tilde{\boldsymbol{\Lambda}}_{ij,t} \right). \quad (59)$$

With analogous reasoning regarding the agreement constraint slack term for $\mathbf{w}$ in (10), we obtain

$$\nabla_{\boldsymbol{\nu}_{ij}} \hat{\mathcal{L}}_t(\mathbf{D}_{t+1}, \mathbf{w}_{t+1}, \boldsymbol{\Lambda}_t, \boldsymbol{\nu}_t) = \mathbf{w}_{i,t+1} - \mathbf{w}_{j,t+1} + \tilde{\boldsymbol{\nu}}_{ij,t}. \quad (60)$$

With the same reasoning, the dual update for the Lagrange multipliers associated with the model parameter agreement constraints may be expressed as

$$\boldsymbol{\nu}_{ij,t+1} = \boldsymbol{\nu}_{ij,t} + \epsilon_t \left(\mathbf{w}_{i,t} - \mathbf{w}_{j,t} + \tilde{\boldsymbol{\nu}}_{ij,t} \right). \quad (61)$$

which is as stated in Proposition 1.

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] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[3] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

[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, J. Mairal, M. Elad, M. Elad, G. Sapiro, and G. Sapiro, "Sparse representation for color image restoration," in *the IEEE Trans. on Image Processing*. ITIP, 2007, pp. 53–69.

[6] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Comput.*, vol. 13, no. 4, pp. 863–882, Apr. 2001. [Online]. Available: http://dx.doi.org/10.1162/089976601300014385

[7] 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

[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2008.79

[9] 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.

[10] 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

[11] D. Bradley and Bagnell, "Differentiable Sparse Coding," in *Proceedings of Neural Information Processing Systems 22*, Dec. 2008.

[12] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd." in *CVPR*. IEEE, 2011, pp. 1697–1704. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2011.html#JiangLD11

[13] 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.

[14] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural Parallel & Scientific Comp.*, vol. 11, no. 4, pp. 351–368, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=964886

[15] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *arXiv preprint arXiv:1410.1386*, 2014.

[16] 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

[17] 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

[18] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J Optimiz. Theory App.*, vol. 142, no. 1, pp. 205–228, Aug. 2009.

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

[20] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J Optimiz. Theory App.*, vol. 147, no. 3, pp. 516–545, Sep. 2010.

[21] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *ArXiv e-prints 1310.7063*, Oct. 2013.

[22] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *IEEE 6th Workshop Signal Process. Adv. in Wireless Commun Process.*, Jun. 5-8 2005, pp. 1088–1092.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Convex optimization with sparsity-inducing norms," in *Optimization for Machine Learning*. MIT Press, 2011.

[31] S. Boyd and L. Vanderberghe, *Convex Programming*. New York, NY: Wiley, 2004.

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

[33] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

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

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

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

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

[38] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[39] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, Sept. 2000.

[40] R. G. J. Wijnhoven, "Fast Training of Object Detection Using Stochastic Gradient Descent," in *International Conference on Pattern Recognition*, 2010, pp. 424–427.

[41] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.167

[42] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.