# D4L: Decentralized Dynamic Discriminative Dictionary Learning

Alec Koppel[1], Garrett Warnell[2], Ethan Stump[2], and Alejandro Ribeiro[1]

***Abstract*— We consider discriminative dictionary learning in a distributed online setting, where a team of networked robots aims to jointly learn both a common basis of the feature space and a classifier over this basis from sequentially observed signals. 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. Only neighboring nodes in the communications network need to exchange information, and we penalize the discrepency between the individual feature basis and classifiers using Lagrange multipliers. The application we consider is for a team of robots to collaboratively recognize objects of interest in dynamic environments. As a preliminary performance benchmark, we consider the problem of learning a texture classifier across a network of robots moving around an urban setting where separate training examples are sequentially observed at each robot. Results are shown for both a standard texture dataset and a new dataset from an urban training facility, and we compare the performance of the standard centralized construction to the new distributed algorithm for the case when distinct samples from all classes are seen by the robots. These experiments yield comparable performance between the decentralized and the centralized cases, demonstrating the proposed method's practical utility.**

## I. Introduction

We seek to develop a system to allow a network of robotic agents 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 classifiers over these representations, and extending these formulations in a dynamic, networked setting. Our particular application of interest is for a network of robots driving through an urban environment to perform real-time texture classification for the purpose of mapping, navigability analysis, and object recognition.

Sparse coding, or a representing a feature vector as a linear combination of a small number of basis elements, using a learned dictionary, rather than a predefined one has yielded state of the art results for image processing tasks such as denoising [1] and classification [2], [3]. A classical way to handle this representation problem is principle component analysis [4], which requires orthogonality of the basis elements. In dictionary learning we relax this requirement and seek a data-driven representation of the signal. The task of actually learning the dictionary is a difficult optimization problem, especially in the context of large or dynamic training sets.

A signal $\mathbf{x} \in \mathbb{R}^m$ admits a sparse approximation of $\mathbf{x}$ over a dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$ if it may be represented as a combination of a *small* number of basis elements that is *close* to $\mathbf{x}$. Sparse coding methods have been successfully applied to a variety of signal processing applications [5]. One approach is to use a pre-defined dictionary based upon the application domain, i.e. wavelets for natural imagery [6]. Learning the dictionary from the data rather than using a predefined method has shown to significantly improve signal reconstruction tasks such as inpainting or denoising [1], [7], [8], and has also been successfully applied to higher level signal processing tasks such as classification [2], [9], yet recent extensions which tailor the dictionary to the specific learning task show significant improvements [10]–[12]. We refer to the approach of tailoring the dictionary to the task of learning a predictive model as *discriminative* dictionary learning.

Dictionary learning in the online setting, where training samples are sequentially observed, has been solved as a matrix factorization problem using first [1] and second-order stochastic approximation methods [13]. However, the problem of developing a dictionary representation of a signal specifically suited to the problem setting of interest is more challenging to optimize. Recently, an online framework for large-scale dictionary and discriminative model learning has been proposed based upon alternating stochastic gradient [14] which successfully generalizes the task-specific dictionary methods for classification [15] and compact feature learning, the later of which has also been approached with convolutional neural networks and Boltzmann machines.

In this paper, we extend the online discriminative dictionary learning formulation of [14] to networked settings, where a team of agents seeks to learn a common dictionary and model parameters based upon local dynamic information, which is a discriminative extension of [16]. To develop a framework for solving discriminative dictionary learning problems online in networked settings, we consider tools from stochastic approximation and distributed optimization.

Pertinent to the approach considered here is projected stochastic gradient [17], and its extensions to networks. Such extensions have incorporated approaches from distributed optimization such as weighted averaging [18]–[21], 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]. We note that [16] uses a weighted averaging method for networked stochastic opti-

This work is supported by the Department of Defense SMART Scholarship Program and Army Research Laboratory's MAST CTA.

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

[2]U.S. Army Research Laboratory, Computational and Information Sciences Directorate, 2800 Powder Mill Road, Adelphi, MD 20783. Email: {garrett.a.warnell.civ, ethan.a.stump2.civ}@mail.mil

mization, which as shown in [25], [26], is ill-suited to tasks of this kind as it may seek a consensus which diverges from the globally optimal decision variable. Hence we develop a modification of the primal-dual algorithm proposed in [26] which more effectively solves the problem of learning a common dictionary and discriminative model online in the multi-agent setting.

The paper is organized as follows. We begin in Section II by describing the dictionary learning and sparse representation problem [13], and develop its discriminative extension [14], both of which are stochastic programs. In Section III, we extend this problem to multi-agent systems, and derive an algorithmic solution based upon the saddle point algorithm of Arrow and Hurwicz [24], [26]. In Section IV, we demonstrate the proposed framework's practical utility in the context of mobile robotic teams for collaborative learning tasks and conclude in Section V.

## II. Problem Formulation

### A. Dictionary Learning from Data

Consider a set of $T$ signals in an $m$-dimensional feature space $\{\mathbf{x}_t\}_{t=1}^T \subset \mathcal{X} \subset \mathbb{R}^m$. We aim to represent the signals $\{\mathbf{x}_t\}_{t=1}^T$ as a sparse combination of a common set of $k$ basis elements, which are unknown and must also be learned from the data. Denote the dictionary as $\mathbf{D} \in \mathbb{R}^{m\times k}$, the sparse coding as $\boldsymbol{\alpha} \in \mathbb{R}^k$ and associate a loss function $\tilde{f}_t(\boldsymbol{\alpha}, \mathbf{D})$ with each data point which is small when $\boldsymbol{\alpha}$ and $\mathbf{D}$ *sparsely represent* $\mathbf{x}_t$ well. Classically the the dictionary learning and sparse representation problem [27] has been formulated as the empirical loss minimization

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

Often the number of data points $T$ is large, and the signal dimension $m$ is small.

An ideal way to induce sparsity in the coding $\boldsymbol{\alpha}$ would be with an $\ell_0$ constraint which yields a NP-hard combinatorial optimization problem. To circumvent this issue, [28] develop soft thresholding methods based upon proximal operators. These methods are computationally efficient and converge quickly, yet have been shown to be less numerically stable than convex relaxations [29], [30] with an elastic-net ($\ell_1$ and $\ell_2$) penalty, which we state as

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

The subscript $u$ denotes the *unsupervised* data driven method for learning the dictionary. For a fixed $\mathbf{D}$, (2) is an elastic net problem, also known as $\ell_2$ regularized lasso [31] or basis pursuit [5], for which efficient exact solvers exist [32]. The $\ell_1$-regularizer induces sparsity in $\boldsymbol{\alpha}$. Moreover, $\zeta_1$ denotes a regularization parameter tuning the sparsity level of the coefficients and $\zeta_2$ tunes how equitably the sparse coding is spread across its $k$ coordinates. The $\ell_2$ regularization also guarantees (2) is strongly convex and may be solved uniquely [33]. There is no analytical link between $\zeta_1$ and the sparsity level, and hence values of $\boldsymbol{\alpha}$ may becoming arbitrarily small, which corresponds to the entries of $\mathbf{D}$ from becoming arbitrarily large. To eliminate the scale ambiguity of the bilinear term in (2), constrain the set of feasible dictionaries to be those whose columns are of unit norm, i.e.

$$\mathcal{D} = \{\mathbf{D} \in \mathbb{R}^{m\times k} : \|\mathbf{d}_l\| \leq 1, l = 1 \dots k\}. \tag{3}$$

### B. Dictionary Learning for Discriminative Modeling

Following [14], we modify the stochastic optimization problem formulated in [13] such that the dictionary learning is *supervised* to the signal processing task of interest, which has yielded promising results in image [10] and audio [2] applications. To do so, begin by defining the optimal sparse coding of (2) as

$$\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x}) := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|_2^2+\zeta_1\|\boldsymbol{\alpha}\|_1+\zeta_2\|\boldsymbol{\alpha}\|_2^2, \tag{4}$$

where we associate with each signal $\mathbf{x}$ a variable $\mathbf{y} \in \mathcal{Y}$. Here $\mathcal{Y}$ denotes a set of labels in the case of classification or $\mathcal{Y} \subset \mathbb{R}^q$ in the case of regression. We aim to discern the input-output relationship associated with the pair $(\mathbf{x}, \mathbf{y})$ by learning model parameters $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^k$. We use the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ in (4) as a feature representation of the signal, and seek to minimize a convex smooth loss function of the form $f_s(\mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x}))$, where the subscript $s$ denotes the *supervised* component of the learning. This loss captures how well one may predict $\mathbf{y}$ when given the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ for the dictionary $\mathbf{D}$. The structure of $f_s$ is dependent on the learning task of interest, examples of which include the squared, logistic, and squared hinge-loss for linear and logistic regression or support vector classification, respectively.

We view the prediction loss $f_s$ as a function of both the model $\mathbf{w}$ and the dictionary selection $\mathbf{D}$, since the sparse coding $\boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x})$ is clearly dependent on the selection of basis elements. Hence we seek to learn $\mathbf{D}$ and $\mathbf{w}$ jointly by solving

$$\min_{\mathbf{D}\in\mathcal{D}, \mathbf{w}\in\mathcal{W}} \mathbb{E}_{\mathbf{y},\mathbf{x}}\left[f_s(\mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}^\star(\mathbf{D}; \mathbf{x}))\right] + \frac{\xi}{2}\|\mathbf{w}\|_F^2. \tag{5}$$

where $\xi$ is a regularization parameter guaranteeing the problem is strongly convex in $\mathbf{w}$ when the dictionary and sparse coefficients are fixed. By using the analysis in [14], we may use smooth optimization methods to solve (5) despite the non-smooth sparsity-inducing norm in (2). Thus both the model and dictionary are tuned for prediction risk in (5).

### C. Extension to Networks

We propose solving (5) in distributed settings, where the signal $\mathbf{y}$ is independently observed by agents of a network which aim to learn a dictionary and model parameters in common with all others while only having access to local information. To this end, fix a network $\mathcal{G} = (V, \mathcal{E})$ which is assumed to be symmetric and connected network with node set $V = \{1, \dots, N\}$ and $M = |\mathcal{E}|$ directed edges of the form $e = (i, j)$. That the network is symmetric means that if $e = (i, j) \in \mathcal{E}$ it must also be that $e' = (j, i) \in \mathcal{E}$. That the network is connected means that all pairs of

nodes are connected by a chain of edges. We also define the neighborhood of $i$ as the set of nodes $n_i := \{j : (i,j) \in \mathcal{E}\}$ that share an edge with $i$. Suppose the functions $f_u$ in (2) and $f_s$ may be written as a sum of local losses available at different nodes of a network, i.e.

$$f_u(\mathbf{D};\mathbf{x}) = \sum_{v=1}^{N} f_{i,u}(\mathbf{D}_i;\mathbf{x}_i), \tag{6}$$

$$f_s(\mathbf{y},\mathbf{w},\boldsymbol{\alpha}^\star(\mathbf{D};\mathbf{x})) = \sum_{i=1}^{N} f_{i,s}(\mathbf{y}_i,\mathbf{w}_i,\boldsymbol{\alpha}^\star(\mathbf{D}_i;\mathbf{x}_i)). \tag{7}$$

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

Since the loss functions $f_{i,u}$ and $f_{i,s}$ are the same for all agents $i$, dictionary and model parameter selections that are good for one agent are also good for another. Thus, a suitable strategy is to learn a dictionary $\mathbf{D}_i$ and model $\mathbf{w}_i$ in the same way for each agent. Since the network $\mathcal{G}$ is assumed to be connected, this relationship can be attained by imposing the constraints $\mathbf{D}_i = \mathbf{D}_j$ and $\mathbf{w}_i = \mathbf{w}_j$ for all pairs of neighboring nodes $(i,j) \in \mathcal{E}$. Substituting (6) into the objective in (5) with these constraints, we obtain the following networked stochastic program:

$$\begin{aligned} \min_{\mathbf{D}\in\mathcal{D}^N,\mathbf{w}\in\mathcal{W}^N} \quad & \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i,\mathbf{x}_i}[f_s(\mathbf{y}_i,\mathbf{w}_i,\boldsymbol{\alpha}_i^\star(\mathbf{D}_i;\mathbf{x}_i))] + \frac{\xi}{2}\|\mathbf{w}_i\|^2. \\ \text{such that} \quad & \mathbf{D}_i = \mathbf{D}_j, \mathbf{w}_i = \mathbf{w}_j \text{ for all } j \in n_i \end{aligned} \tag{8}$$

Here each agent $i$ aims to learn a common dictionary $\mathbf{D}_i$ and discriminative model $\mathbf{w}_i$ that asymptotically converges to the solution of (5). Note that when the agreement constraints in (8) are satisfied, the problems (5) and (8) are equivalent. Thus (8) corresponds to a problem in which each agent $i$, having only observed the *local* signals $\mathbf{y}_i$, aims to learn a dictionary representation and model parameters that are optimal when information is aggregated globally over the network

## III. Block Saddle Point Method

We turn to deriving an algorithmic solution to (8), the dynamic discriminative dictionary learning problem in networks. We build upon the stochastic gradient approach of [1] which is competitive with other stochastic approximation based dictionary learning methods.

To derive the saddle point algorithm for this problem, we need a manner for computing the sparse coding [cf. (2)] efficiently. The loss function in (2) is a regularized least squares problem, for which several approaches have been proposed. Those based upon coordinate descent with soft-thresholding converge quickly [28], [31], yet lack the numerical stability of those based upon homotopy methods [34]. We compute the sparse codings using the Elastic-Net modification of Least Angle Regression (LARS-EN) [33] method for solving lasso and elastic-net regression problems, which solves for the entire regularization path.

Formulating a distributed algorithm is not possible if the agreement constraint in (8) is enforced for each realization of the random variable $\mathbf{y}$. By considering a Lagrangian relaxation of the agreement constraint, we develop a block stochastic variant of the Arrow-Hurwicz Saddle Point Algorithm [24]–[26]. In the dictionary and model updates, we implement a stochastic gradient method with a dual correction term to account for local discrepancy. The Lagrange multipliers are updated via a dual ascent step which penalizes local dictionary and model parameter disagreement, and are transmitted across network communication links.

First, write the constraints in (8) more compactly by defining the vertical block concatenation matrices $\mathbf{D} := [\mathbf{D}_1;\ldots;\mathbf{D}_N] \in \mathbb{R}^{Nm\times k}$ and $\mathbf{w} := [\mathbf{w}_1;\ldots;\mathbf{w}_N] \in \mathbb{R}^{Nk}$. We define an the augmented graph edge incidence matrix associated with each constraint as follows $\mathbf{C}_D : \mathbb{R}^{Nm\times k} \to \mathbb{R}^{Mm\times k}$. The matrix $\mathbf{C}_D$ is formed by $M \times N$ square blocks of dimension $mk$. If the edge $e = (i,j)$ links node $i$ to node $j$ the block $(e,i)$ is $[\mathbf{C}_D]_{ei} = \mathbf{I}_{mk}$ and the block $[\mathbf{C}_D]_{ej} = -\mathbf{I}_{mk}$, where $\mathbf{I}_{mk}$ denotes the identity matrix of dimension $mk$. All other blocks are identically null, i.e., $[\mathbf{C}]_{el} = \mathbf{0}_{mk}$ for all edges $e \neq (i,j)$. The matrix $\mathbf{C}_\mathbf{w}$ is defined in the exact same way, substituting the model parameter dimension $np$ for the dictionary dimension $mk$. 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

$$\begin{aligned} \mathbf{C}_D\mathbf{D} &= \mathbf{0}, \\ \mathbf{C}_\mathbf{w}\mathbf{w} &= \mathbf{0}. \end{aligned} \tag{9}$$

The edge incidence matrices $\mathbf{C}_D$ and $\mathbf{C}_\mathbf{w}$ have exactly $mk$ and $m$ null singular values, respectively. We denote as $0 < \gamma$ the smallest nonzero singular value of $\mathbf{C} := [\mathbf{C}_D;\mathbf{C}_\mathbf{w}]$ and as $\Gamma$ the largest singular value of $\mathbf{C}$. The singular values $\gamma$ and $\Gamma$ 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 (5). Instead, we consider a modification of (6) in which we add linear penalty terms to incentivize the selection of coordinated actions. Introduce then dual variables $\boldsymbol{\Lambda}_e = \boldsymbol{\Lambda}_{ij} \in \mathbb{R}^{m\times k}$ associated with the constraint $\mathbf{D}_i - \mathbf{D}_j = \mathbf{0}$ and consider the addition of penalty terms of the form $\mathrm{tr}[\boldsymbol{\Lambda}_{ij}^T(\mathbf{D}_i - \mathbf{D}_j)]$. For an edge that starts at node $i$, the multiplier $\boldsymbol{\Lambda}_{ij}$ is assumed to be kept at node $i$. Similarly, introduce dual variables $\mathbf{N}_{ij} \in \mathbb{R}^{n\times p}$ associated with the constraint $\mathbf{w}_i - \mathbf{w}_j = \mathbf{0}$ for all neighboring node pairs and penalty terms $\mathbf{N}_{ij}^T(\mathbf{w}_i - \mathbf{w}_j)$. By introducing the stacked matrices $\boldsymbol{\Lambda}_\cdot = [\boldsymbol{\Lambda}_1;\ldots;\boldsymbol{\Lambda}_M] \in \mathbb{R}^{Mm\times k}$ and $\mathbf{N} := [\mathbf{N}_1;\ldots;\mathbf{N}_M] \in \mathbb{R}^{Mn\times p}$, we may define the Lagrangian of the decentralized dynamic discriminative dictionary learning problem as

$$\mathcal{L}(\mathbf{D},\mathbf{w},\mathbf{\Lambda},\mathbf{N}) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{y}_i,\mathbf{x}_i}[f_s(\mathbf{y}_i,\mathbf{w}_i,\boldsymbol{\alpha}_i^\star(\mathbf{D}_i;\mathbf{x}_i))] + \frac{\xi}{2}\|\mathbf{w}_i\|^2 + \mathrm{tr}\big(\mathbf{\Lambda}^T\mathbf{C}_D\mathbf{D}\big) + \mathrm{tr}\big(\mathbf{N}^T\mathbf{C}_\mathbf{w}\mathbf{w}\big) \quad (10)$$

This function is a smooth non-convex function of the primal variables $\mathbf{D}, \mathbf{w}$ and a concave function of its Lagrange multipliers $\mathbf{\Lambda}, \mathbf{N}$. Suppose agent $i$ receives a realization of the local random variables at time $t$ as $\mathbf{x}_{i,t}$ with associated output (label) $\mathbf{y}_{i,t}$. Using this interpretation of the Lagragian we consider the use of the Arrow-Hurwicz saddle point method in parallel block variable updates. This method exploits 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) with fixed sparse coefficients $\boldsymbol{\alpha}_t^\star$, the saddle point algorithm takes the form

**Algorithm 1** D4L: Decentralized Dynamic Discriminative Dictionary Learning

**Require:** $\mathbf{D}_0$ (initial dictionary); $\mathbf{y}_u$ (local random variables); $\zeta_1, \zeta_2, \xi \in \mathbb{R}$ (regularization parameters)

1: **for** $t = 0, 1, 2, \ldots$ **do**

2: **for** Agent $i = 1, \ldots, N$ **do**

3: Acquire independent local signal-output pair $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$

4: Sparse coding: compute using LARS-EN

$$\boldsymbol{\alpha}_{i,t+1}^\star = \underset{\tilde{\boldsymbol{\alpha}}\in\mathbb{R}^k}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x}_{i,t+1} - \mathbf{D}_{i,t}\tilde{\boldsymbol{\alpha}}\|_2^2 + \zeta_1\|\tilde{\boldsymbol{\alpha}}\|_1 + \frac{\zeta_2}{2}\|\tilde{\boldsymbol{\alpha}}\|_2^2.$$

5: Send Lagrange multipliers $\mathbf{\Lambda}_{ij,t}, \boldsymbol{\nu}_{ij,t}$ to neighbors $j \in n_i$, receive $\mathbf{\Lambda}_{ji,t}, \boldsymbol{\nu}_{ji,t}$

6: Compute active set $Z_{i,t}$ as the indices associated with nonzero entries of $\boldsymbol{\alpha}_{i,t+1}^\star$.

7: Compute $\boldsymbol{\beta}_{i,t}^\star$: Set $[\boldsymbol{\beta}_{i,t}]_{Z_{i,t}^C} = 0$ and

$$[\boldsymbol{\beta}_{i,t}]_{Z_{i,t}} = \Big([\mathbf{D}_{i,t}]_{Z_{i,t}}^T[\mathbf{D}_{i,t}]_{Z_{i,t}} + \zeta_2\mathbf{I}\Big)^{-1}\nabla_{\boldsymbol{\alpha}_{Z_{i,t}}} f_s(\mathbf{y}_i,\mathbf{w}_i,\boldsymbol{\alpha}_{i,t}^\star).$$

8: Update dictionary and model parameters

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

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

9: Update Lagrange Multipliers at communication link $(i, j)$

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

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

10: **end for**

11: **end for**

$$\mathbf{D}_{t+1} = \mathcal{P}_\mathcal{D}\big[\mathbf{D}_t - \epsilon_t\nabla_\mathbf{D}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)\big], \quad (11)$$

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

Similarly, the update in the dual domain which tracks the price of dictionary and model parameter disagreement, takes the form

$$\mathbf{\Lambda}_{t+1} = \mathcal{P}_\mathfrak{L}\big[\mathbf{\Lambda}_t + \epsilon_t\nabla_\mathbf{\Lambda}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)\big], \quad (13)$$

$$\mathbf{N}_{t+1} = \mathcal{P}_N\big[\mathbf{N}_t + \epsilon_t\nabla_\mathbf{N}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)\big], \quad (14)$$

where $\nabla_\mathbf{D}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)$, $\nabla_\mathbf{w}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)$, $\nabla_\mathbf{\Lambda}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)$, and $\nabla_\mathbf{N}\hat{\mathcal{L}}_t(\mathbf{D}_t,\mathbf{w}_t,\mathbf{\Lambda}_t,\mathbf{N}_t)$ are the stochastic subgradients of the Lagrangian, which are approximates of the gradient of the expectation term evaluated at the current realizations of the signals $\mathbf{x}_{i,t}$.$\mathcal{P}_\mathcal{X}$ denotes the projection onto the set $\mathcal{X}$. Moreover, $\epsilon_t$ is a step size which is usually chosen as $O(1/t)$, and will be discussed further in Section IV. The quantity $\nabla_\mathbf{D}\mathbb{E}_{\mathbf{y},\mathbf{x}}[f_s(\mathbf{y},\mathbf{w},\boldsymbol{\alpha}^\star(\mathbf{D};\mathbf{x}))]$ is derived in [14]. The remaining derivatives are easily computed from (10) and are incorporated into the description of Algorithm 1.

## IV. Experiments

Our goal is for a team of robots to identify objects of interest in a real-time decentralized manner when deployed in dynamic environments. However, each robot 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 achieving this task. By communicating with other robots in the network, each agent may learn over a broader domain associated with that which has been explored by the whole robotic network, and hence more effectively identify objects of interest.

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 [35], [36], 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, partitions them into small patches, and identifies properties of each patch. Experimentally we consider cases where each agent observes training examples which contain all of the class types, and is able to communicate with all others in the network.

### A. Feature Generation

Inspired by the two-dimensional texton features discussed in [37], we generate texture features to classify, $\mathbf{z}$, as the sum of the sparse dictionary representations of sub-patches. That is, we classify image patches of size 24-by-24 by first extracting the nine non-overlapping 8-by-8 sub-patches within it. We vectorize (column-major order) and normalize (zero mean and unit $\ell_2$ norm) each sub-patch and use this collection of vectors as columns in a matrix

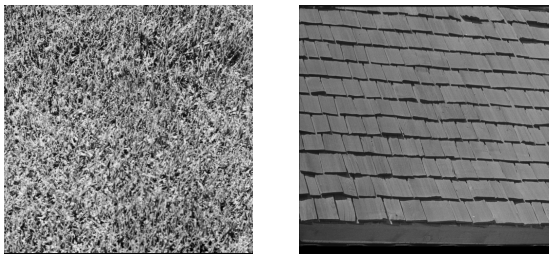

Fig. 1. Sample images from the Brodatz texture database.

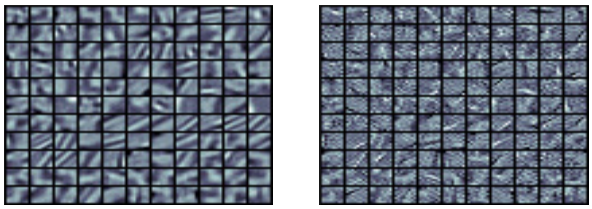

Fig. 2. Initialized (left) and final (right) dictionary for 8-by-8 grayscale patches. These dictionaries were computed using the centralized ($N = 1$) algorithm with step-size $\epsilon = 0.25$.

$\mathbf{X} = \left[\mathbf{x}^{(1)}; \cdots; \mathbf{x}^{(9)}\right]$. We then compute the feature $\mathbf{z}_{i,t}$ at robot $i$ at time $t$ according to

$$\mathbf{z}_i(\mathbf{X}_{i,t}, \mathbf{D}_{i,t}) = \sum_{l=1}^{9} \boldsymbol{\alpha}^{\star}(\mathbf{D}_{i,t}; \mathbf{x}_{i,t}^{(l)}) \,. \tag{15}$$

Note that this form for $\mathbf{z}$ means that at time $t$ the local stochastic gradient of the dictionary $[\nabla_{\mathbf{D}_i} \hat{f}_{i,s}]_t$ is the sum of contributions from each sub-patch representations, i.e.,

$$[\nabla_{\mathbf{D}_i} \hat{f}_{i,s}]_t = \sum_{l=1}^{9} -\mathbf{D}_{i,t}\boldsymbol{\beta}_{i,t}^{\star(l)}\boldsymbol{\alpha}_{i,t}^{\star(l)} + (\mathbf{x}_{i,t} - \mathbf{D}_{i,t}\alpha_{i,t}^{\star(l)})\boldsymbol{\beta}_{i,t}^{\star(l)^T} \,, \tag{16}$$

where $\boldsymbol{\alpha}_{i,t}^{\star(l)} = \boldsymbol{\alpha}_{i,t}^{\star(l)}(\mathbf{D}_{i,t}; \mathbf{x}_{i,t})$ is used to compute $\boldsymbol{\beta}_{i,t}^{\star(l)}$ as defined in Algorithm 1.

### B. Classifier and Loss Function

We cast texture classification as a multi-class logistic regression problem in which agent $i$ receives signals $\mathbf{x}_{i,t}$ and is charged with outputting a decision variable $\mathbf{y}_{i,t} \in \{0,1\}^C$ where $C$ is the number of classes. Each component $[\mathbf{y}_{i,t}]_c$ of the vector $\mathbf{y}_{i,t}$ is a binary indicator of whether the signal falls into class $c$. The supervised local loss $f_{i,s}$ for this problem specification is the negative log-likelihood of the corresponding probabilistic model (see [38] for more detail) stated as

$$f_{i,s}(\mathbf{y}_i, \mathbf{W}_i, \mathbf{z}_i) = \log\left(\sum_{c=1}^{C} e^{\mathbf{w}_{i,c}^T \mathbf{z}_i + w_{i,c}^0}\right) - \sum_{c=1}^{C} y_{i,c}\mathbf{w}_{i,c}^T\mathbf{z}_i + w_{i,c}^0 \,, \tag{17}$$

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

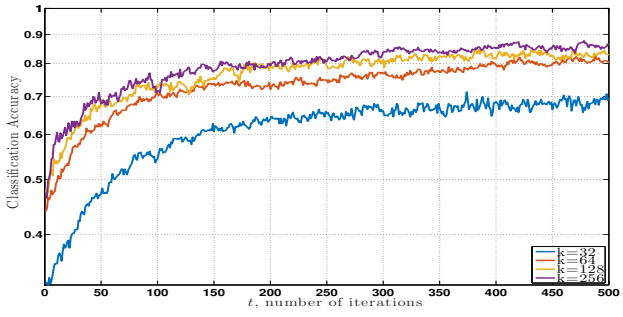


Fig. 3. We plot the classification accuracy over iteration number $t$ for various dictionary sizes $k$ for the Brodatz texture dataset in the centralized case with constant step-size $\epsilon = 0.25$. Observe that increasing the dictionary size improves performance with diminishing returns for $k \geq 128$.

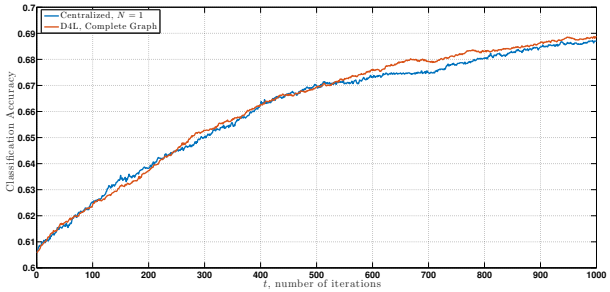


Fig. 4. D4L classifier performance improvement for the Brodatz texture dataset with step-size $\epsilon = 10^{-2}$. "D4L, Complete" refers to a fully-connected, three-node network where each node has access to training data from all classes. Classifier performance is averaged over all nodes. An accuracy of 0.70 is achieved by iteration $T = 10^3$, although the accuracy continues to improve as additional training examples are processed. Observe that the decentralized and centralized algorithms yield comparable performance.

### C. Design Considerations

Before considering the implementation on a robotic network, we seek to understand which problem parameters yield acceptable empirical performance. To do so, we study the learning achieved by the D4L algorithm on the Brodatz texture database [39]. In this case, the data is made up of four class labels {grass, bark, straw, herringbone_weave}. Sample images from this data set are shown in Figure 1. The Brodatz texture dataset consists of one grayscale image per texture. For the subset we consider here, this amounts to four 512-by-512 images that together consist of $956,484$ overlapping patches of size 24 by 24.

*(i) Dictionary Size* To select a dictionary of the appropriate size suited to this problem, we investigate its affect on performance using the Brodatz texture dataset. 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 only. The performance of the resulting classifier on the testing set is shown in Figure 3. As in [14], we find that

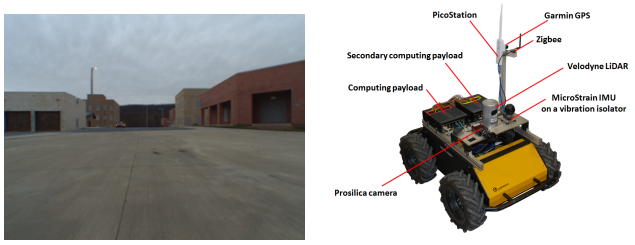


Fig. 5. Sample image from the IRA dataset (left) associated with an $N = 3$ node network of Husky robots (right) moving around a cluttered urban setting.

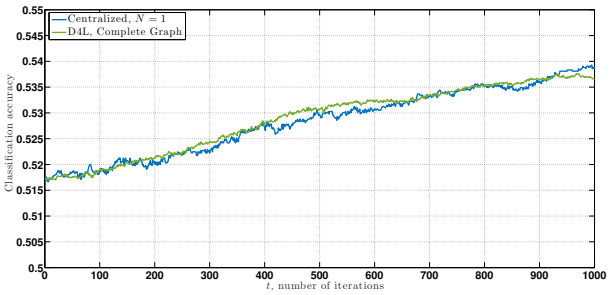


Fig. 6. D4L classifier performance improvement for the IRA texture dataset with step-size $\epsilon = 5 \times 10^{-3}$. "D4L, Complete" refers to a fully-connected, three-node network where each node has access to training data from all classes. Classifier accuracy is averaged over all robots. Increasing the step-size $\epsilon$ 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.

increasing the size of the dictionary led to better classifier performance. However, because of diminishing performance returns, we select $k = 128$ in all subsequent experiments due to computation time considerations. 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. That is, at each iteration, we replaced the single labeled patch with a small batch of 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 behavior. [1]

*(iii) Initialization* We initialized $\mathbf{D}$ using unsupervised dictionary learning [13] for a small set of randomly-drawn initialization data. We then used the labels and the dictionary representations of the data to initialize the classifier parameters $\mathbf{W}$.

*(iv) Parameter Selection* The D4L algorithm requires several parameters to be specified. Following [14], we used $\zeta_1 = 0.125$, $\zeta_2 = 0$, $\xi = 10^{-9}$. We also adopted the learning-rate selection strategy discussed in [14], which is to select the initial step-size $\epsilon$ by implementing a grid search over a fixed small number of iterations ($T = 2 \times 10^2$) and using the one that minimized the cross-validation error. When implementing the mini-batch stochastic algorithm, we set $\epsilon_t = \min(\epsilon, \epsilon t_0/t)$, where $t_0 = T/2$. This selection amounts to using a step-size of $\epsilon$ for the first half of the iterations before and then following 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 $\epsilon$ is too large. This follows from the fact convergence guarantees for stochastic gradient algorithms in non-convex settings only occur under certain conditions on the distributions of the stochastic gradient errors, which may not hold if the step-size is too large.

[1] Another variance reduction technique one may consider is computing the running empirical average of the local stochastic gradients, which introduces memory into the algorithm.

Moreover, we have experimentally observed that values of $\epsilon$ 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 $\epsilon$ that yield convergence for both settings, i.e., the smaller values appropriate for D4L. For the Brodatz dataset, we found that $\epsilon = 10^{-2}$ led to convergence.

*(v) Results on Texture Database* We study the performance of D4L for multi-class texture classification on the Brodatz dataset for two cases: *(A)* the centralized case (see [14]), and *(B)* a $N = 3$ node fully-connected network where each node has access to observations from every class, which is the experimental setting of our robotic network that we describe in the next section. We quantified performance by using a small testing set to compute the average empirical loss and global time-average classifier 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 labels, $\hat{\mathbf{y}}_{i,t}$ denotes the predicted labels, and $P(\hat{\mathbf{y}}_{i,t} \neq \mathbf{y}_{i,t})$ represents the empirical misclassification rate on a fixed test set of size $\tilde{T} = 4.096 \times 10^3$.

Results for both settings *(A)* and *(B)* using the Brodatz dataset are shown in Figure 4. With this choice of $\epsilon$, observe that algorithm behavior is extremely similar under both *(A)* and *(B)*. In particular, both methods achieve a classification accuracy of .69 by $T = 10^3$ and continue to improve at similar rates, demonstrating that we achieve comparable performance in the centralized and decentralized cases.

### D. Robotic Experiments

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 5 for an example of the images taken by a prototypical Husky platform. Running the D4L algorithm on these image observations resembles a field implementation of a robotic network. 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, grass, building, concrete_floor}. 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 $\epsilon$ that avoided divergence 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 6. Due to the more challenging nature of field data and hence distributions of stochastic gradient errors associated with the updates (11)-(14), reducing 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. Observe that the accuracy of the algorithm continues to climb as more data is accumulated, though by $T = 10^3$ we approach an accuracy of only .54.

In the centralized case, step-sizes which are orders of magnitude larger still yield convergent behavior, and hence learning occurs at a faster rate, as may be observed for $k = 128$ in Figure 3 as compared with Figure 4. With the larger $\epsilon$ tolerated by the centralized algorithm, a much larger improvement is seen after $T = 10^3$ iterations. We note that for larger $T$, we still observe improvement, but the experiments have been truncated due to computation time. Typically in distributed algorithm one expects learning to occur more slowly than in centralized formulations, and this expectation is verified empirically in both the Brodatz and IRA robotic network settings.

## V. Conclusion

This work represents the first attempt to extend the discriminative dictionary learning problem of [14] to networked settings. To do so we formulated 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.

Our main goal is to design strategies for teams of robots in dynamic environments to collaboratively perform object recognition. Due to the technical limitations of computer vision algorithms to date, we consider texture classification in dynamic networked settings as a preliminary benchmark for this problem. To do so we use sub-patches and consider a multiclass logistic regression formulation of this problem.

Our experiments demonstrated comparable classifier performance between the centralized and decentralized settings, but it is important to note that the the final classifier is not competitive with existing approaches to texture classification (e.g. [37], [40]). These existing approaches utilize nearest-neighbor classifiers, however it is not clear how such a classifier can be adapted to the framework of [14] and therefore what was presented here.

Though the D4L algorithm does result in improvement of the joint classification performance, the overall improvement was much smaller than that seen by using our texture classification scheme in the centralized implementation of [14] due to the small step sizes required for convergence. The asymptotic convergence of D4L is established in [41], yet the largest learning rate able to achieve this convergence is still unknown. By better understanding the feasible step-sizes in the distributed case, we expect better performance to be possible. However, because algorithms for non-convex stochastic optimization in distributed settings is a still incipient research area, achieving convergence to stationarity in such settings is more challenging than the centralized case.

Although the gains were small, we must note that the networked performance *did increase*, confirming that discriminative dictionary learning in networks was achieved, providing a baseline for which decentralized collaborative object recognition may be achieved in dynamic robotic networks. It remains future work to modify the algorithm to better handle the equality constraints without destroying the convergence properties of the original discriminative dictionary learning, thus improving the convergence rate in the decentralized case.

## VI. Acknowledgements

We would like to thank Arne Suppe and Luis Navarro-Serment from Carnegie Mellon University for the use of the IRA data which allowed us to simulate a field implementation.

## References

[1] 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.

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

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, 2008, pp. 1033–1040.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[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.

[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.

[13] 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.

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

[15] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, Dec. 2008.

[16] 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.

[17] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

[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, Sept. 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] 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.

[26] A. Koppel, F. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, p. 14, Sept. 2014.

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

[28] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

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

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

[33] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[34] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

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

[36] 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, Sept. 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.167

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

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

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

[40] 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.

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