PARALLEL STOCHASTIC SUCCESSIVE CONVEX APPROXIMATION METHOD FOR LARGE-SCALE DICTIONARY LEARNING

Alec Koppel[§]*, Aryan Mokhtari*[†]*, and Alejandro Ribeiro*[‡]

[§]Computational and Information Sciences Directorate, U.S. Army Research Laboratory
[†]Simons Institute for the Theory of Computing, University of California Berkeley
[‡]Department of Electrical and Systems Engineering, University of Pennsylvania

ABSTRACT

We consider the problem of dictionary learning over training sets whose sample size and parameter dimension are large-scale, which is formulated as a non-convex stochastic program where the objective decomposes into a smooth non-convex part and a convex sparsity-promoting penalty. We propose a Doubly Stochastic Successive Convex approximation scheme (DSSC) as a new numerical tool to address this task which operates by decomposing the dictionary and sparse codes into blocks and operates on random subsets of blocks at each step. The algorithm belongs to the family of successive convex approximation methods since we replace the original non-convex stochastic objective by a strongly convex sample surrogate function, and solve the resulting convex program, for each randomly selected block in parallel. The method operates on subsets of features (block coordinate methods) and training examples (stochastic approximation) at each step. In contrast to many training schemes for dictionary learning, DSSC attains almost sure convergence to a stationary solution of the problem. We observe the practical benefits of this approach for stable learning and computational speedup when applied to streaming visual data gathered by a field robot.

Index Terms— Dictionary Learning, non-convex optimization, stochastic methods, large-scale optimization, parallel optimization

1. INTRODUCTION

Transformations of data domains have become widely used in the past decades, due to their ability to extract useful information from input signals as a precursor to solving higher-level tasks such as statistical inference or learning-based control. For instance, if the signal dimension is very large, dimensionality reduction is of interest, which may be approached with principal component analysis [2]. If instead one would like to analyze the signal at multiple resolutions, wavelets [3] may be more appropriate. These techniques, which also include nonparametric approaches such as k-nearest neighbor, belong to a family of tools called unsupervised learning [4]. Recently, methods based on signal representations learned from data, rather than those fixed a priori, have gained traction for tasks such as image in-painting or de-noising [5,6], as well as reinforcement learning [7]. Learned feature exctraction is broadly referred to as representation learning, a special case of which is dictionary learning [8, 9], the focus of this work, specifically when the training data has both a large sample size and parameter dimension.

The problem of developing a dictionary representation of a signal when both the training sample size and parameter dimension are huge-scale is a difficult non-convex stochastic optimization problem. In the case where the sample size is small but the parameter dimension is large, one may use block coordinate descent (BCD) [10], or alternating gradient methods [11], but these cannot operate with data subsets at each step. Stochastic methods address the computational bottleneck in the sample size [12] by optimizing over subbatches of data, but are not guaranteed to converge for non-convex problems [5, 13], and cannot address deal with large parameter dimension p. To address the non-convexity, one may apply convex techniques to non-convex settings [14], implement simulated annealing [15], or approximate the non-convex function by a convex one [16, 17]. This later approach, termed successive convex approximation (SCA), has the benefit of gracefully handling non-smooth regularization, while possibly exploiting the convex sub-structure of the non-convex term [17]. In this work, we develop a parallelizable stochastic sub-sampling method based upon BCD and SCA to address each of these challenges and obtain a globally convergent numerical tool for dictionary learning from high-dimensional data.

The work of A. Ribeiro is supported by NSF CAREER CCF-0952867, ONR N00014-12-1-0997, and ASEE SMART. Proofs may be found in [1].

2. ALGORITHM DEVELOPMENT

We address the task of dictionary learning using sparse representations from possibly infinitely many training examples when the number of predictive parameters p is large. To do so, consider a collection of signals $\{\mathbf{z}_n\} \subset \mathbb{R}^p$ for $n = 1, \ldots, N$. We seek to represent signals $\mathbf{z}_n$ as combinations of a common set of k basis elements $\{\mathbf{d}_l\}_{l=1}^k$ which are unknown and must be learned from data. We group these basis elements into a dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{p\times k}$ and denote the coding of $\mathbf{z}_n$ as $\boldsymbol{\alpha}_n \in \mathbb{R}^k$. The coding $\boldsymbol{\alpha}_n$ are the coefficients of $\mathbf{z}_n$ with respect to dictionary $\mathbf{D}$. The dictionary learning and coding problem calls for finding a coding $\boldsymbol{\alpha}$ and dictionary $\mathbf{D}$ such that the signal $\mathbf{z}$ is close to its dictionary representation $\mathbf{D}\boldsymbol{\alpha}$ for all samples. This goal can be mathematically formulated by introducing a loss function $f(\boldsymbol{\alpha}, \mathbf{D}; \mathbf{z}) = \|\boldsymbol{\alpha}\mathbf{D} - \mathbf{z}\|_2^2$ that depends on the proximity between $\mathbf{D}\boldsymbol{\alpha}$ and the data point $\mathbf{z}$. We focus on the case where the sparsity dimension is greater than the feature dimension, i.e., $k \geq p$, but we want the codes to be sparse, motivated by sparse representation methods. This goal may be incentivized via ℓ_1 regularization, yielding (with $N \to \infty$)

$$(\mathbf{D}^*, \boldsymbol{\alpha}^*) := \underset{\mathbf{D}\in\mathcal{D}, \boldsymbol{\alpha}\in\mathcal{A}}{\operatorname{argmin}} \ \mathbb{E}_{\mathbf{z}}[\|\mathbf{D}\boldsymbol{\alpha} - \mathbf{z}\|_2^2] + \zeta\|\boldsymbol{\alpha}\|_1 \ . \qquad (1)$$

Here $\mathcal{D} := \{\mathbf{D} \in \mathbb{R}^{p\times k} : \|\mathbf{d}_l\| \leq 1, \text{ for all } l\}$ is the set of $p \times k$ matrices with unit column norm that is introduced to eliminate scale ambiguity in the bilinear term $\mathbf{D}\boldsymbol{\alpha}$ in (1) and $\mathcal{A} \subset \mathbb{R}^k$ is a compact set in which sparse codes live. The bilinear term makes (1) non-convex. Moreover, we do not assume the total number of examples

$\mathbf{z}_n$ is finite, yielding the statistical average loss in (1). In this paper, we are interested in developing algorithms that use a subset of samples of random signals $\mathbf{z}$ (training points) – stochastic approximation of the function $F(\mathbf{D}, \boldsymbol{\alpha}) := \mathbb{E}_{\mathbf{z}}[\|\mathbf{D}\boldsymbol{\alpha} - \mathbf{z}\|_2^2]$ – and a subset of coordinates at each iteration. Moreover, we aim to design parallel algorithms. For simplicity, we subsequently stack the dictionary and sparse codes together in one matrix $\mathbf{x} = [\mathbf{D}; \boldsymbol{\alpha}] \in \mathbb{R}^{p\times k+k}$, such that $F(\mathbf{D}, \boldsymbol{\alpha}) = F(\mathbf{x})$ and define the objective we seek to minimize in (1) as $V(\mathbf{x}) := F(\mathbf{x}) + \zeta\|\boldsymbol{\alpha}\|_1$. Moreover, ζ is a regularization parameter which makes $\boldsymbol{\alpha}$ sparser as it increases. Further define the stacked dimension $\tilde{p} := p \times k + k$.

Fix a collection of I parallel processors, with $I \leq B$ and assume that I blocks are chosen uniformly at random from the total B blocks. Consider $i \in \{1, \ldots, B\}$ as the index of the block assigned to one of the I processors, i.e., each processor updates *one* block. Further define a training subset $\mathbf{Z}_i^t$ corresponding to block i at time t consisting of L instantaneous functions. $\mathbf{Z}_i^t$ may be thought of as random subsets of rows of the training data matrix. The aggregate set of selected blocks at time t is denoted by $\mathcal{I}^t \subset \{1, \ldots, B\}$, with $|\mathcal{I}^t| = I$. Note that our construction departs from [18]: rather than use the block-wise stochastic gradient to develop an algorithm based on only the linearization of the instantaneous objective, we replace the instantaneous non-convex objective with a convex surrogate. The benefit of this approach is incorporating the additional non-smooth convex term $\zeta\|\alpha_i\|_1$ associated to block i in (1), yielding a scheme which includes parallel proximal stochastic gradient as a special case.

To design a scheme based on SCA, we consider use of instantaneous convex surrogate functions for the original functions at each step. Let $\mathbf{x}_{-i}$ denote the sub-vector obtained from $\mathbf{x}$ by deleting $\mathbf{x}_i$. To derive the update for the block variable $\mathbf{x}_i$ corresponding to the i-th block, at iteration t, consider the objective with $\mathbf{x}_i$ fixed, i.e., $\mathbb{E}\left[f(\mathbf{x}_i, \mathbf{x}_{-i}^t, \mathbf{z}^t)\right] + g(\mathbf{x}_i, \mathbf{x}_{-i}^t)$. In our proposed SCA scheme, we replace the non-convex function $f(\mathbf{x}_i, \mathbf{x}_{-i}^t, \mathbf{z}^t)$, the stochastic approximation of the aforementioned objective, by a proper local convex function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{z}^t)$, which we call the surrogate function corresponding to block i. We define $\tilde{f}_i$ as a proper surrogate function if it satisfies the following conditions [17].

Assumption 1. *Consider $\mathbf{x}_{-i}$ as the concatenation of all coordinates of $\mathbf{x}$ other than those of block i. The surrogate $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \mathbf{z})$ associated with the i-th block of $\mathbf{x}$, i.e., $\mathbf{x}_i$, satisfies the following:*

(a) $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \mathbf{z})$ is differentiable, convex w.r.t. $\mathbf{x}_i$ for all $\mathbf{x}$, $\mathbf{z}$.

(b) $\nabla_{\mathbf{x}_i}\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \mathbf{z}) = \nabla_{\mathbf{x}_i} f(\mathbf{x}, \mathbf{z})$ for all $\mathbf{x}$, $\mathbf{z}$.

(c) $\nabla_{\mathbf{x}_i}\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \mathbf{z})$ is Lipschitz continuous on $\mathcal{X}$ with constant Γ.

The conditions in Assumption 1 for the surrogate functions are mild and there exists a large range of functions satisfying Assumption 1. The most popular choice for the surrogate $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \boldsymbol{\theta})$ is

$$\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{z}) = f(\mathbf{x}^t, \mathbf{z}) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{z})^T(\mathbf{x}_i - \mathbf{x}_i^t) + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^t\|^2, \quad (2)$$

where $\tau_i > 0$. It is easy to show that the surrogate function in (2) satisfies the conditions in Assumption 1 and is strongly convex with constant τ_i. This selection yields a variant of proximal stochastic gradient methods, but many alternatives exist [17, 19].

2.1. Doubly Stochastic Successive Convex approximation method

Computation of the average function $F(\mathbf{x})$ or its gradients $\nabla F(\mathbf{x})$ is prohibitively costly, so we instead devise an algorithm that uses stochastic approximation of the $F(\mathbf{x})$ combined with successive convex approximations. Moreover, to reduce the computation time of the algorithm, we are interested in schemes that, at each iteration, update only a *subset of coordinates* (blocks) of the decision variable $\mathbf{x}$. We introduce DSSC as a doubly stochastic method for non-convex composite optimization that meets these requirements.

To do so, define the mini-batch sample surrogate function as $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{Z}_i^t) = \frac{1}{L}\sum_{\mathbf{z}\in\mathbf{Z}_i^t}\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{z})$ for a given a set of realizations $\mathbf{Z}_i^t$, where L is the size of the mini-batch. Further define the mini-batch surrogate function gradient associated with the set $\mathbf{Z}_i^t$: $\nabla\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{Z}_i^t) = \frac{1}{L}\sum_{\mathbf{z}\in\mathbf{Z}_i^t}\nabla\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{z})$. The index i for the training subset $\mathbf{Z}_i^t$ shows that we use distinct sample points to approximate functions for each block, so each processor operates on distinct data in parallel.

The update for coordinate i of the DSSC is based on two steps. First, we convexify the non-convex stochastic composite problem (1) by introducing the strongly convex surrogate $\tilde{f}_i$, and solve the strongly convex sample problem, stated as

$$\hat{\mathbf{x}}_i^{t+1} = \underset{\mathbf{x}_i\in\mathcal{X}_i}{\operatorname{argmin}}\Big\{\rho^t\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{Z}_i^t) + (1-\rho^t)(\mathbf{d}_i^{t-1})^T(\mathbf{x}_i - \mathbf{x}_i^t) + g_i(\mathbf{x}_i) + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^t\|^2\Big\}, \quad (3)$$

where τ_i is a positive constant, $g_i(\mathbf{x}_i) = \zeta\|\alpha_i\|_1$, and ρ^t is a sequence of positive scalars (to be properly chosen) . First, note that proximity term $(\tau_i/2)\|\mathbf{x}_i - \mathbf{x}_i^t\|^2$ makes the loss in (3) strongly convex, so problem (3) has a unique solution, denoted by $\hat{\mathbf{x}}_i^{t+1}$. The linear term $\mathbf{d}_i^t$ in (3) is a time average of stochastic gradients associated to block i, updated as [19]

$$\mathbf{d}_i^t = (1-\rho^t)\mathbf{d}_i^{t-1} + \rho^t\nabla_{\mathbf{x}_i}\tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \mathbf{Z}_i^t). \quad (4)$$

Observe that the update in (3) is similar to a block-wise proximal stochastic gradient step [20], with two key differences: the recursively averaged stochastic gradient $\mathbf{d}_i^{t-1}$ takes place of the stochastic gradient, and the surrogate function $\tilde{f}_i$ is used in lieu of the most recent stochastic gradient. These augmentations of the proximal step allow us to guarantee almost sure convergence in non-convex settings, a property that often eludes first-order stochastic methods (Section 3). The update in (4) shows that instead of approximating the gradient of the function F with its stochastic approximation gradient $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{Z}_i^t)$, which is equivalent to the surrogate function gradient $\nabla_{\mathbf{x}_i}\tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \mathbf{Z}_i^t)$, we use a heavy-ball type average of the observed stochastic gradients for the i-th block. It can be shown that the sequence $\mathbf{d}_i^t$ approaches the exact gradient $\nabla_{\mathbf{x}_i}F(\mathbf{x}^t)$ [19, 21].

The second step in DSSC is computing $\mathbf{x}_i^{t+1}$ as a weighted average of the previous iterate $\mathbf{x}_i^t$ and the solution $\hat{\mathbf{x}}_i^{t+1}$ of (3):

$$\mathbf{x}_i^{t+1} = (1-\gamma^{t+1})\mathbf{x}_i^t + \gamma^{t+1}\hat{\mathbf{x}}_i^{t+1}. \quad (5)$$

γ^t in (5) is an attenuating step-size, to be properly chosen.

Equations (3) and (5) define the updates of a single block $\mathbf{x}_i^{t+1}$. In DSSC we allow for simultaneous parallel updates of different block coordinates of $\mathbf{x}$, which are selected uniformly at random. Note that this scheme is different from cyclic scheme in [10] that allows for one block update per iteration or the greedy rule in [17], but instead resembles the random coordinate method in [18]

The overall DSSC algorithm is summarized in Algorithm 1. The core steps are Step 7, 9, and 10. In Step 7, the processor that operates on i-th block computes the auxiliary variable $\hat{\mathbf{x}}_i^{t+1}$ by solving the minimization in (3), e.g., a block-wise proximal gradient step

Algorithm 1 DSSC at processor operating on block i

1: **Require:** sequences γ^t and ρ^t.
2: **for** $t = 0, 1, 2 \dots$ **do**
3: Read the variable $\mathbf{x}^t$
4: Receive the randomly chosen block $i \in \{1, \dots, B\}$
5: Choose training subset $\mathbf{Z}_i^t$ for block $\mathbf{x}_i$
6: Compute surrogate function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{Z}_i^t)$
7: Compute variable $\hat{\mathbf{x}}_i^{t+1}$ as the solution of (3)
8: Compute surrogate gradient $\nabla \tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \mathbf{Z}_i^t)$
9: Update average gradient $\mathbf{d}_i^t$ associated with block i [cf. (4)]
10: Compute the updated variable $\mathbf{x}_i^{t+1}$ [cf. (5)]
11: **end for**

or Quasi-Newton step with a recursively averaged descent direction. To do this, the processor needs to access to vector $\mathbf{x}^t$ (Step 3), the block that it should work on (Step 4), and a training subset $\mathbf{Z}_i^t$ (Step 5) to compute its corresponding surrogate function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}_{-i}^t, \mathbf{Z}_i^t)$ (Step 6). In Step 9, the processor updates the stochastic average gradient $\mathbf{d}_i^t$ associated to the block i using the surrogate gradient $\nabla \tilde{f}_i(\mathbf{x}_i; \mathbf{x}_{-i}^t, \mathbf{Z}_i^t)$ which is evaluated in Step 8. Finally, the variable $\mathbf{x}_i^{t+1}$ is computed in Step 10 using the weighted average of the previous iterate $\mathbf{x}_i^t$ and the auxiliary variable $\hat{\mathbf{x}}_i^{t+1}$.

3. CONVERGENCE ANALYSIS

In this section, we establish that the iterates defined by (3)-(5) converge to a stationary point of (1). Before deriving the theoretical results, we state some standard technical conditions which are required to hold for their proofs.

Assumption 2. *The sets $\mathcal{X}_i$ are convex and compact.*

Assumption 3. *Let $\mathcal{F}^t$ be the sigma-algebra generated by the collection of past realizations of $\mathbf{x}$ and $\mathbf{z}$ up to iteration t, i.e. $\mathcal{F}^t \supset \{(\boldsymbol{x}^u, \boldsymbol{z}^u)\}_{u \leq t}$. The instantaneous gradients $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{z}^t)$ induce stochastic errors whose conditional variance is finite:*

$$\mathbb{E}\left[\|\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{z}^t) - \nabla_{\mathbf{x}_i} F(\mathbf{x}^t)\|^2 \mid \mathcal{F}^t\right] \leq \sigma^2 < \infty. \quad (6)$$

Assumption 4. *The statistical average objective $F(\mathbf{x})$ has L-Lipschitz continuous gradients, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$*

$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \quad (7)$$

Assumption 2 is customary in non-convex optimization and guarantees bounded iterates of the algorithm. The condition in Assumption 2 implies that the set $\mathcal{X}$ is also convex and compact. Hence, we can assume there exists a positive constant D such that $\|\mathbf{x} - \mathbf{y}\| \leq D$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Note that the instantaneous gradient $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{z}^t)$ is an unbiased estimator of $\nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$, given the information available until t, i.e., $\mathbb{E}\left[\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{z}^t) \mid \mathcal{F}^t\right] = \nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$. Thus, Assumption 3 ensures the variance of the stochastic gradient as an estimator of the true gradient is bounded. Assumption 4 allows for the simplification of higher order terms that appear in Taylor's expansion.

Next we stipulate that diminishing step-sizes and momentum parameters are needed in order to establish convergence.

Parameter selection. *The algorithm learning rate $\gamma^t \in [0, 1]$ and gradient momentum parameter $\rho^t \in [0, 1]$ are chosen such that*

(i) $\lim_{t\to\infty} \gamma^t = 0$, $\sum_{t=0}^{\infty} \gamma^t = \infty$, $\sum_{t=0}^{\infty} (\gamma^t)^2 < \infty$,

(ii) $\lim_{t\to\infty} \rho^t = 0$, $\sum_{t=0}^{\infty} \rho^t = \infty$, $\sum_{t=0}^{\infty} (\rho^t)^2 < \infty$,

(iii) $\sum_{t=0}^{\infty} (\gamma^t)^2 / \rho^t < \infty$.

The first two parameter conditions make the noise of stochastic approximation asymptotically null. The last condition is required to show that the sequence of stochastic average gradients $\mathbf{d}_i^t$ converges to $\nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$ almost surely by making use of gradient consistency.

Under these conditions, we establish almost sure convergence when diminishing step-size and averaging parameters are used.

Theorem 1. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (3)-(5) *with gradient averaging rate $\rho^t \in [0, 1]$ and algorithm step-size $\gamma^t \in [0, 1]$ satisfying the conditions* (i) - (iii)*. Under Assumptions 1-4, the DSSC algorithm converges to a stationary solution of* (1)*.*

Theorem 1 establishes that Algorithm 1 is guaranteed with probability 1 to attain the minimizer of the composite objective defined by the regularized expected risk minimization problem with a non-convex objective (1). While similar results hold for problems of this type [17, 22], they do not apply to problems defined by training sets where both feature dimension and sample size are huge-scale, while the objective function is non-convex. Therefore, Theorem 1 is one of the most generic convergence guarantees for non-convex large-scale learning problems, and is achieved by a method whose complexity is easily tunable by the number of processors and mini-batch size. Next, we consider specific choices of ρ^t and γ^t to derive an upper-bound of the convergence rate.

Theorem 2. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (3)-(5)*. Further assume that the parameters ρ^t and γ^t are chosen as*

$$\rho^t = \mathcal{O}\left(\frac{1}{t^{\frac{1}{2}+\epsilon}}\right) \quad \text{and} \quad \gamma^t = \mathcal{O}\left(\frac{1}{t^{\frac{3}{4}+\frac{3}{2}\epsilon}}\right), \quad (8)$$

where $\epsilon > 0$ is an arbitrary small constant. Under Assumptions 1-4, the following mean convergence rate holds

$$\mathbb{E}\left[\|\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t\|^2\right] = o\left(\frac{1}{t^{\frac{1}{4}-\frac{3}{2}\epsilon}}\right) \quad (9)$$

Theorem 2 establishes that Algorithm 1 converges at least at a rate of $o(t^{-1/4+(3/2)\epsilon})$ to a stationary solution of (1). The convergence results in this section establish that DSSC successfully solves non-convex stochastic programs with large parameter dimensions. In the next sections, we observe that these theoretical results translate well into practice for dictionary learning.

4. ROBOTIC DICTIONARY LEARNING

We conducted experiments using a Clearpath Husky robot (see Figure 1) at Camp Lejeune, North Carolina, a cluttered urban setting surrounded by forest. We collected images that were sequentially observed by the platform – see Figures 1b and 1c for examples. The images collected from the path traversed by the robot belong to distinct classes of {grass, pavement, building, sky}. Each image converted to a 320×240 pixel grayscale image, whose entries are pixel intensities (elements of the unit interval), and partitioned into $610,528$ overlapping 24-by-24 sub-patches. Dictionaries are trained using these sub-patches as observations. We consider the overdetermined case where the number of atoms $k = 120$ is greater than the patch dimension and the coefficient vector α we seek is sparse. This is a canonical setting for visual feature extraction [6].

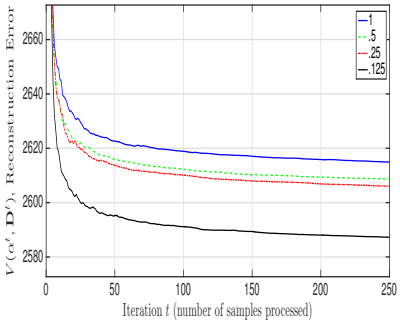
We run Algorithm 1 on the problem (1) for a variety of coordinate selection strategies off-platform in order to assess which strategy yields fastest convergence. In particular, we vary the proportion

(a) Clearpath Husky Robot.

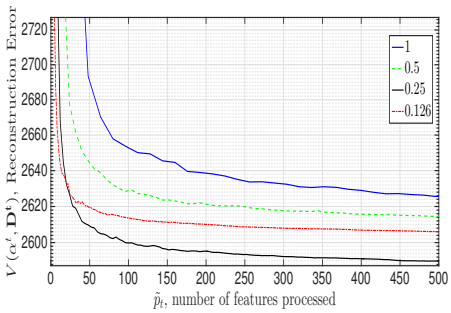
(b) Sample taken by platform in urban area.

(c) Sample taken by platform near forest.

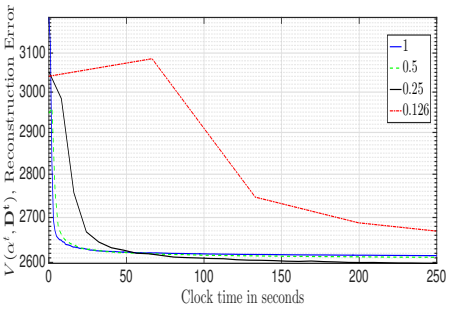
Fig. 1: The platform on which we conduct experiments is a prototypical ground vehicle robot, a Clearpath Husky, which is equipped with a high-fidelity camera. As the robot drives around the environment, we collect images, which are used to build the IRA data set.



(a) Objective $V(\mathbf{x}^t)$ vs. iteration t



(b) Objective $V(\mathbf{x}^t)$ vs. feature $\tilde{p}_t$



(c) Objective $V(\mathbf{x}^t)$ vs. clock time (s)

Fig. 2: DSSC applied to training an unsupervised dictionary and sparse code when the number of processors I is fixed as $I = 16$, and the number of blocks are $B = 16$, $B = 32$, $B = 64$, and $B = 128$, (i.e., $r = 1$, $r = .5$, $r = .25$, and $r = .128$, respectively). Convergence in terms of the regularized reconstruction error (1) relative to the number of features processed improves with *fewer* blocks updated per iteration (Figures 2a and 2b). This trend is inverted when we instead consider clock time (Figure 2c) because of the limitations of simulated, rather than implemented, parallel architectures.

of updated coordinates $r = I/B$ per iteration, which corresponds to varying the number of parallel processors I operating on a single block at each step. We fix B, the number of blocks and vary r as $r = 1/8$, $r = 1/4$, $r = 1/2$, and $r = 1$, which corresponds to updating all blocks at each step.

Results The quantitative results of this empirical experiment on (down-sampled) high-fidelity robotic visual data for the dictionary learning problem (1) are given in Figure 2, where we plot the objective (1) $V(\boldsymbol{\alpha}^t, \mathbf{D}^t)$ with respect to different quantifiers of the amount of information processed. In Figure 2a, we plot the convergence with iteration t. Observe that learning accelerates with fewer blocks (smaller r) relative to iteration count.

This trend is corroborated in Figure 2b: we plot the sparsity-regularized reconstruction error versus the number of features processed. The convergence rate increases with *decreasing* r here, both when the rate is measured in terms of samples and features processed. This trend is in contrast to non-proximal block stochastic gradient methods (RAPSA) [22], where this relationship may not hold. This distinguishing trend of DSSC relative to RAPSA is due to the presence of the proximal term and the non-smoothness of the problem (1), which makes successive convex approximation inherit more of the benefits of Gauss-Seidel style updates in block coordinate methods as compared with the smooth convex setting of [22].

Figure 2c displays the convergence path of DSSC on the problem (1) in terms of clock time for our numerical simulation. We observe that the favorable trends seen in Figures 2a and 2b are inverted in 2c, suggesting that to recover the improved convergence of DSSC, parallel computing architecture *implementations* are required, and simulations of parallel computing cannot match physical reality. The takeaway is that learning rates on autonomous systems may be optimized by equipping them with GPUs tailored towards parallel data processing.

5. CONCLUSION

In this work, we developed a new numerical optimization tool for learning dictionary-based signal representations from training data when the number of samples N is huge-scale, as is the parameter dimension p. This method alleviates the complexity bottleneck in both N and p by operating on subsets of both samples (via stochastic approximation) and parameters (using block coordinate methods). Moreover, dictionary learning defines a non-convex problem, which we circumvented through use of successive convex approximation. We then proved that the resulting algorithm converges to a stationary point of the problem and established its learning rate. Empirical validation on an online dictionary learning from visual data collected by a ground robot demonstrated that reliably and stable learning occurs in practice, despite the non-convexity.

6. REFERENCES

[1] A. Mokhtari, A. Koppel, G. Scutari, and A. Ribeiro, "Large-scale nonconvex stochastic optimization by doubly stochastic successive convex approximation," *University of Pennsylvania Technical Report*, 2017. [Online]. Available: https://koppel.bitballoon.com/assets/papers/dssc_report.pdf

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

[5] 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

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

[7] H. van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, "Stable reinforcement learning with autoencoders for tactile and visual data," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 3928–3934.

[8] F. Sheikholesalmi and G. B. Giannakis, "Online subspace learning and nonlinear classification of big data with misses," in *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*. IEEE, 2015, pp. 1–6.

[9] Y. Shen and G. B. Giannakis, "Online dictionary learning from large-scale binary data," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1808–1812.

[10] 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.

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

[12] 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

[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. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756008

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge, U.K: Cambridge Univ. Press, 2004.

[15] D. Bertsimas, J. Tsitsiklis *et al.*, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[16] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *Signal Processing, IEEE Transactions on*, vol. 62, no. 3, pp. 641–656, 2014.

[17] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *Signal Processing, IEEE Transactions on*, vol. 63, no. 7, pp. 1874–1889, 2015.

[18] A. Mokhtari, A. Koppel, and A. Ribeiro, "Doubly random parallel stochastic methods for large scale learning," in *2016 American Control Conference (ACC)*, July 2016, pp. 4847–4852.

[19] Y. Yang, G. Scutari, D. P. Palomar, and M. Pesavento, "A parallel decomposition method for nonconvex stochastic multiagent optimization problems," *IEEE Transactions on Signal Processing*, vol. 64, no. 11, pp. 2949–2964, 2015.

[20] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[21] A. Ruszczyński, "Feasible direction methods for stochastic programming problems," *Mathematical Programming*, vol. 19, no. 1, pp. 220–229, 1980.

[22] A. Mokhtari, A. Koppel, and A. Ribeiro, "A class of parallel doubly stochastic algorithms for large-scale learning," *arXiv preprint arXiv:1606.04991*, 2016.

[23] M. Wang, E. X. Fang, and H. Liu, "Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions," *Mathematical Programming*, vol. 161, no. 1-2, pp. 419–449, 2017.