LARGE-SCALE NONCONVEX STOCHASTIC OPTIMIZATION BY DOUBLY STOCHASTIC SUCCESSIVE CONVEX APPROXIMATION

Aryan Mokhtari[†]*, Alec Koppel*[†]*, Gesualdo Scutari*[⋆]*, and Alejandro Ribeiro*[†]

[†]Department of Electrical and Systems Engineering, University of Pennsylvania
[⋆]School of Industrial Engineering, Purdue University

ABSTRACT

We consider supervised learning problems over training sets in which both the number of training examples and the dimension of the feature vectors are large. We focus on the case where the loss function defining the quality of the parameter we wish to estimate may be non-convex, but also has a convex regularization. We propose a Doubly Stochastic Successive Convex approximation scheme (DSSC) able to handle non-convex regularized expected risk minimization. The method operates by decomposing the decision variable into blocks and operating 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 stochastic convex methods whose *almost sure behavior* is not guaranteed in *non-convex* settings, DSSC attains almost sure convergence to a stationary solution of the problem. Numerical experiments on a non-convex variant of a lasso regression problem show that DSSC performs favorably in this setting.

Index Terms— Non-convex optimization, stochastic methods, large-scale optimization, parallel optimization, lasso

1. INTRODUCTION

Statistical parameter estimation in the case of large datasets is often formulated as an optimization problem with a stochastic objective [1]. Define the parameter vector we wish to estimate as $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ and the random variable $\boldsymbol{\theta} \in \mathbb{R}^q$ defining training examples of a dataset as inputs to the random function $f(\mathbf{x}, \boldsymbol{\theta})$. Further define the average function $F(\mathbf{x}) := \mathbb{E}\left[f(\mathbf{x}, \boldsymbol{\theta})\right]$ as the expectation of the random functions $f(\mathbf{x}, \boldsymbol{\theta})$. We focus on minimizing the global objective $V(\mathbf{x}) := F(\mathbf{x}) + G(\mathbf{x})$ which is the sum of the non-convex smooth average function $F(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x}, \boldsymbol{\theta})\right]$ and the non-smooth convex function $G(\mathbf{x})$,

$$\min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x}) := \min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) + G(\mathbf{x}). \tag{1}$$

Since p is assumed to be large, we decompose the parameter vector into blocks $\mathbf{x} = [\mathbf{x}_1; \cdots; \mathbf{x}_B]$ with $\mathbf{x}_i \in \mathcal{X}_i \subset \mathbb{R}^{p_i}$ where $p_i << p$, and assume that the aggregate data domain admits the Cartesian structure $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_B$. We consider the case where the function $G(\mathbf{x})$ decomposes along the block variables $G(\mathbf{x}) = \sum_{i=1}^B g_i(\mathbf{x}_i)$, where $\mathbf{x}_i$ is the i-th block of the vector $\mathbf{x}$. Problems of this form arise in regression and classification, with statistical models trained on nonlinear transformations of the feature space that appears in, e.g., compressive sensing approaches to texture identification [2, 3] facial recognition [4, 5], and neural networks [6].

The work of Mokhtari, Koppel, and Ribeiro is supported by NSF CAREER CCF-0952867, ONR N00014-12-1-0997, and ASEE SMART. The work of Scutari is supported by by the NSF Grants CIF 1564044, CCF 1632599, and CAREER Award 1555850, and the ONR N00014-16-1-2244.

To solve Problem (1) there are four main challenges to deal with, namely: (a) the non-convexity of the objective $F(\mathbf{x})$; (b) the dependence of $F(\mathbf{x})$ on the expectation $\mathbb{E}\left[f(\mathbf{x}, \boldsymbol{\theta})\right]$ which either does not have a closed form or is not computationally affordable; (c) the presence of the nonsmooth function $G(\mathbf{x})$; and (d) the large dimension of the feature space p. Current works cannot address all the above challenges.

In particular, to deal with the nonconvexity of $F(\mathbf{x})$ (Issue (a)) while converging to stationary solutions an effective approach is leveraging Successive Convex Approximation techniques (SCA) [7, 8]: the original nonconvex problem is replaced by a sequence of convexified ones. This approach, which we adopt here, is also well-suited to handle the presence of the non-smooth terms in the objective function (Issue (c)). Additionally, to iteratively solve (1), one must evaluate the gradient of $F(\mathbf{x})$ which in practice is not available (Issue (b)). Stochastic approximation methods operate using a subset of the data to approximate, at each iteration, the gradient of F [9]. First-order methods [9, 10], and quasi-Newton schemes [11–14] use stochastic gradients in lieu of true gradients. Combining the benefits of both approaches, in this paper, we propose using a stochastic (i.e., sample) convex surrogate of the nonconvex objective $F(\mathbf{x})$. Furthermore, we consider the case where the number of features p is large (Issue (d)). Existing SCA schemes, including those developed for stochastic objectives, require solving a p-dimensional optimization problem at each iteration. Block coordinate methods have been proposed to solve optimization problems when the number of realizations of $\boldsymbol{\theta}$ is small and p is large by updating only a subset of coordinates of $\mathbf{x}$ at each step [15–17]. Parallelized versions of block coordinate descent have been developed which motivate our coordinate selection scheme [7, 8, 18–20].

In this paper, we propose a novel Doubly Stochastic Successive Convex approximation scheme (DSSC) able to handle all the mentioned issues (Section 2). The proposed DSSC method uses stochastic approximation of the function F – to resolve issue (b) – for a successive convex approximation of the global cost V – to resolve issues (a) and (c). Moreover, DSSC only operates on a subset of coordinates at each iteration – to resolve issue (d) – which simplifies its parallel implementation. This is the first effort to solve the non-convex problem in (1) in a doubly stochastic manner by using stochastic approximation of functions and updating a random subset of coordinates, which is guaranteed to converge almost surely to a stationary point of the problem (Section 3). Moreover, we apply DSSC to a non-convex variant of a lasso regression problem (Section 4). Finally, we close the paper by concluding remarks (Section 5). Proofs of results in this paper are available in [21].

2. ALGORITHM DEVELOPMENT

One of the most popular applications of the problem in (1) is the non-convex empirical risk minimization problem. In particular consider the training set $\mathcal{T}_N$ which contains N samples points and the loss function associate to each sample is a non-convex function $f_i : \mathcal{X} \to \mathbb{R}$. The goal of empirical risk minimization (ERM) is to minimize the regularized average loss which is defined as

$$\min_{\mathbf{x}\in\mathcal{X}} \frac{1}{N}\sum_{i=1}^N f_i(\mathbf{x}) + G(\mathbf{x}), \tag{2}$$

where $G(\mathbf{x})$ is a convex regularizer added to avoid overfitting. Note that the ERM problem in (2) is an instance of Problem (1) when the random variable $\boldsymbol{\theta} = \theta \in \mathbb{R}$ is chosen uniformly at random from the set $\{1, 2, \ldots, N\}$ and the random function $f(\mathbf{x}, \boldsymbol{\theta})$ is defined as $f(\mathbf{x}, \boldsymbol{\theta}) = f(\mathbf{x}, \theta) = f_\theta$. In this paper, we are interested in developing algorithms that use a subset of N samples (training points)–use stochastic approximation of the function F–and a subset of p coordinates at each iteration. Moreover, we aim to design parallel algorithms.

The proposed algorithm operates by fixing a collection of I parallel processors, with $I \leq B$. We 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 $\boldsymbol{\Theta}_i^t$ corresponding to block i at time t consisting of L instantaneous functions. $\boldsymbol{\Theta}_i^t$ 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 $|\mathcal{I}^t| = I$. These selections are then used to define the block-wise mini-batch stochastic gradient of $F(\mathbf{x})$ with respect to block $\mathbf{x}_i \in \mathbb{R}^{p_i}$ as

$$\nabla_{\mathbf{x}_i} f(\mathbf{x}, \boldsymbol{\Theta}_i^t) = \frac{1}{L} \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_i^t} \nabla_{\mathbf{x}_i} f(\mathbf{x}, \boldsymbol{\theta}). \tag{3}$$

Note that our construction departs from [22]: rather than use the block-wise stochastic gradient in (3) to develop an algorithm based on only the linearization of the instantaneous objective, we propose a scheme which replaces the instantaneous non-convex objective with a convex surrogate. The benefit of this approach is incorporating the additional non-smooth convex term $g_i(\mathbf{x}_i)$ associated to block $\mathbf{x}_i$ in (1), yielding a scheme which includes parallel proximal stochastic gradient as a special case. The use of proximal regularization in optimization arises from compressive approaches to, e.g., image processing [5, 23].

To design a scheme based on successive convex approximation, we consider use of instantaneous convex surrogate functions for the original functions at each step. Thus, 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., $\mathbb{E}\left[f(\mathbf{x}_i, \mathbf{x}_{-i}^t, \boldsymbol{\theta}^t)\right] + g(\mathbf{x}_i, \mathbf{x}_{-i}^t)$. In our proposed SCA scheme, we replace the non-convex function $f(\mathbf{x}_i, \mathbf{x}_{-i}^t, \boldsymbol{\theta}^t)$, the stochastic approximation of the aforementioned objective, by a proper local convex function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\theta}^t)$, 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 [8].

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

(a) $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \boldsymbol{\theta})$ is differentiable and convex with respect to $\mathbf{x}_i$ for all $\mathbf{x}$ and $\boldsymbol{\theta}$.

(b) $\nabla_{\mathbf{x}_i} \tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \boldsymbol{\theta})$ is equal to the gradient $\nabla_{\mathbf{x}_i} f(\mathbf{x}, \boldsymbol{\theta})$ for all $\mathbf{x}$ and $\boldsymbol{\theta}$.

(c) $\nabla_{\mathbf{x}_i} \tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \boldsymbol{\theta})$ 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 function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}, \boldsymbol{\theta})$ is

$$\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\theta}^t) = f(\mathbf{x}^t, \boldsymbol{\theta}^t) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta}^t)^T (\mathbf{x}_i - \mathbf{x}_i^t) + \frac{\tau_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^t\|^2, \tag{4}$$

where $\tau_i > 0$. It is easy to show that the surrogate function in (4) satisfies the conditions in Assumption 1 and is strongly convex with constant τ_i. This selection may be used to derive a variant of proximal stochastic gradient methods, but a wide array of alternative choices exist [8, 10].

2.1. Doubly Stochastic Successive Convex approximation method

Since the computation of the average function $F(\mathbf{x})$ or its gradients $\nabla F(\mathbf{x})$ is prohibitively costly, 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 next DSSC as a doubly stochastic method for non-convex composite optimization that meets all these requirements.

To do so, define the mini-batch sample surrogate function as

$$\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\Theta}_i^t) = \frac{1}{L} \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_i^t} \tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\theta}). \tag{5}$$

for a given a set of realizations $\boldsymbol{\Theta}_i^t$. Further define the mini-batch surrogate function gradient associated with the set $\boldsymbol{\Theta}_i^t$:

$$\nabla \tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\Theta}_i^t) = \frac{1}{L} \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_i^t} \nabla \tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\theta}). \tag{6}$$

The index i for the set of realizations $\boldsymbol{\Theta}_i^t$ shows that we use distinct sample points to approximate functions for each block, so each processor operates on a distinct data subset 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

$$\hat{\mathbf{x}}_i^{t+1} = \underset{\mathbf{x}_i \in \mathcal{X}_i}{\operatorname{argmin}} \Big\{ \rho^t \tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \boldsymbol{\Theta}_i^t) + (1 - \rho^t)(\mathbf{d}_i^{t-1})^T (\mathbf{x}_i - \mathbf{x}_i^t) + g_i(\mathbf{x}_i) + \frac{\tau_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^t\|^2 \Big\}, \tag{7}$$

where τ_i is a positive constant. First, note that proximity term $(\tau_i/2)\|\mathbf{x}_i - \mathbf{x}_i^t\|^2$ makes the loss in (7) strongly convex, so problem (7) has a unique solution, denoted by $\hat{\mathbf{x}}_i^{t+1}$ is unique. The linear term $\mathbf{d}_i^t$ in (7) is a time average of stochastic gradients associated to block i, updated as [10]

$$\mathbf{d}_i^t = (1 - \rho^t)\mathbf{d}_i^{t-1} + \rho^t \nabla_{\mathbf{x}_i} \tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \boldsymbol{\Theta}_i^t) \,, \tag{8}$$

and ρ^t is a sequence of positive scalars (to be properly chosen). Observe that the update in (7) is similar to a block-wise proximal stochastic gradient step [24], 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 (8) shows that instead of approximating the gradient of the function F with its stochastic approximation gradient $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t)$, which is equivalent to the surrogate function gradient $\nabla_{\mathbf{x}_i} \tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \boldsymbol{\Theta}_i^t)$, 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 $\nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$ [10, 25].

The second step in the update of 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 (7):

$$\mathbf{x}_i^{t+1} = (1 - \gamma^{t+1})\mathbf{x}_i^t + \gamma^{t+1} \hat{\mathbf{x}}_i^{t+1}. \tag{9}$$

The parameter γ^t in (9) is an attenuating step-size, to be properly chosen.

Equations (7) and (9) 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 [15] that allows for one block update per iteration or the greedy rule in [8], but instead resembles the random coordinate method in [22]

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 $\mathbf{\Theta}_i^t$ for block $\mathbf{x}_i$
6: Compute surrogate function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t, \mathbf{\Theta}_i^t)$ [cf. (5)]
7: Compute variable $\hat{\mathbf{x}}_i^{t+1}$ as the solution of (7)
8: Compute surrogate gradient $\nabla \tilde{f}_i(\mathbf{x}_i^t; \mathbf{x}^t, \mathbf{\Theta}_i^t)$ [cf. (6)]
9: Update the average gradient $\mathbf{d}_i^t$ associated with block i [cf. (8)]
10: Compute the updated variable $\mathbf{x}_i^{t+1}$ [cf. (9)]
11: **end for**

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 (7), e.g., a block-wise proximal gradient step 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 $\mathbf{\Theta}_i^t$ (Step 5) to compute its corresponding surrogate function $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}_{-i}^t, \mathbf{\Theta}_i^t)$ (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 $\nabla \tilde{f}_i(\mathbf{x}_i; \mathbf{x}_{-i}^t, \mathbf{\Theta}_i^t)$ 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 sequence of iterates defined by Algorithm 1 converges almost surely to a stationary solution of (1). To establish this result, first recall the definition of the set $\mathcal{I}^t \subset \{1, \dots, B\}$ as aggregate set of selected blocks at time t with $|\mathcal{I}^t| = I$. Further, define $\mathcal{S}^t \subset \mathcal{X}$ as the set containing the blocks that are updated at step t. Note that components of the set $\mathcal{S}^t$ are chosen uniformly at random from the set of blocks $\{\mathbf{x}_1, \dots, \mathbf{x}_B\}$. Since the number of updated blocks is equal to the number of processors, the ratio of updated blocks is $r := |\mathcal{I}^t|/B = I/B$. The following conditions are required.

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

Assumption 3. *Let $\mathcal{F}^t$ be the sigma-algebra generated by $(\boldsymbol{x}^t, \boldsymbol{\theta}^t)$ up to iteration t. The instantaneous gradients $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta}^t)$ satisfy the condition*

$$\mathbb{E}\left[\|\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta}^t) - \nabla_{\mathbf{x}_i} F(\mathbf{x}^t)\|^2 \mid \mathcal{F}^t\right] < \infty. \tag{10}$$

Assumption 4. *The sequences γ^t and ρ^t 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) $\lim_{t\to\infty} \gamma^t/\rho^t = 0$.

Assumption 2 is customary in non-convex optimization and guarantees bounded iterates of the algorithm. Note that the instantaneous gradients $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta}^t)$ is an unbiased estimator of the gradient $\nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$, given the information available until t, i.e., $\mathbb{E}\left[\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta}^t) \mid \mathcal{F}^t\right] = \nabla_{\mathbf{x}_i} F(\mathbf{x}^t)$. Thus, Assumption 3 ensures the variance of the estimator is bounded. The first two conditions in Assumption 4 are required to show that the noise of stochastic approximation is asymptotically null. The last condition in Assumption 4 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.

We turn to analyzing the sequence of iterates generated by (7) - (9) and show that the algorithm converges to a stationary solution of (1). To do so, we first present two lemmas which establish decrement-like properties on the data-dependent term $F(\mathbf{x})$ and the complexity-reducing penalty $G(\mathbf{x})$. The following lemma concerns the former risk term $F(\mathbf{x})$.

Lemma 1. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (7) - (9) *with non-negative scalar parameters $\rho^t, \gamma^t > 0$. Further, define $\tau = \max_i \tau_i$. If the condition in Assumptions 1 and 2 are satisfied, the following decrement property holds for the expected risk $F(\mathbf{x})$ defined by* (1)

$$\begin{aligned} F(\mathbf{x}^{t+1}) \leq F(\mathbf{x}^t) &+ \gamma^{t+1} \sum_{i \in \mathcal{S}^t} g_i(\mathbf{x}_i^t) - g_i(\hat{\mathbf{x}}_i^{t+1}) \\ &- \gamma^{t+1} \left(\frac{\tau}{2} - \frac{\Gamma \gamma^{t+1}}{2} - \rho^t \Gamma\right) \|\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t\|_{\mathcal{S}^t}^2 \\ &+ \gamma^{t+1} \left(\nabla F(\mathbf{x}^t) - \mathbf{d}^t\right)_{\mathcal{S}^t}^T (\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t)_{\mathcal{S}^t}. \end{aligned} \tag{11}$$

Note that in this section we use the notation $(\mathbf{a})_{\mathcal{S}^t}$ as the concatenation of the blocks $\mathbf{a}$ that belong to the set $\mathcal{S}^t$. The result in Lemma 1 captures the decrement in terms of the function $F(\mathbf{x})$. In the following lemma, we turn our attention to the non-smooth convex term $G(\mathbf{x})$ in (1).

Lemma 2. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (7) - (9) *with averaging rate $\rho^t > 0$ and algorithm step-size $\gamma^t > 0$. If the condition in Assumptions 1 and 2 are satisfied, we obtain*

$$G(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^t) + \gamma^{t+1} \sum_{i \in \mathcal{S}^t} g_i(\hat{\mathbf{x}}_i^{t+1}) - g_i(\mathbf{x}_i^t). \tag{12}$$

The result in Lemma 2 shows an upper bound for the decrement $G(\mathbf{x}^{t+1}) - G(\mathbf{x}^t)$. Subsequently, we use the results in Lemmas 1-2 to prove an upper bound for the expected decrement of the function V.

Lemma 3. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (7) - (9) *with non-negative scalar parameters $\rho^t, \gamma^t > 0$. If the condition in Assumptions 1 and 2 are satisfied, for all steps t we have*

$$\begin{aligned} \mathbb{E}\left[V(\mathbf{x}^{t+1}) \mid \mathcal{F}^t\right] \leq V(\mathbf{x}^t) &- r\gamma^{t+1} \left[\frac{\tau}{2} - \frac{\Gamma \gamma^{t+1}}{2} - \rho^t \Gamma\right] \|\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t\|^2 \\ &+ r\gamma^{t+1} \left(\nabla F(\mathbf{x}^t) - \mathbf{d}^t\right)^T (\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t), \end{aligned} \tag{13}$$

where $\mathcal{F}^t$ is the filtration measuring the random variables up to step t and the expectation is taken with respect to the randomized block selection $\mathcal{S}^t$.

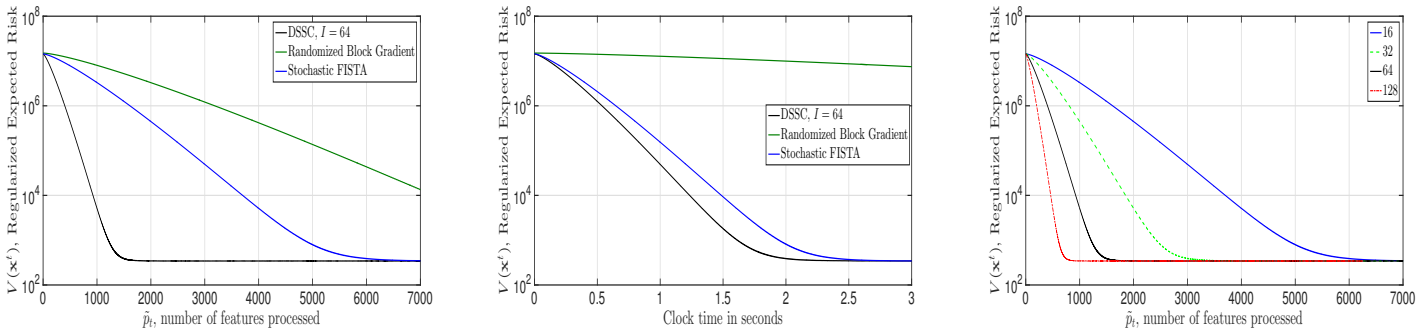
The result in Lemma 3 provides an upper bound for the decrement $\mathbb{E}\left[V(\mathbf{x}^{t+1}) \mid \mathcal{F}^t\right] - V(\mathbf{x}^t)$. We use the result in (13) along with boundedness of the funciton V over the set $\mathcal{X}$ to show that the limit infimum of the sequence $\|\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t\|$ almost surely converges to zero. Moreover, we show that every limit point of the sequence $\mathbf{x}^t$ is a stationary point of (1). These intermediate steps are used in the proof of the following theorem.

Theorem 1. *Consider the sequence $\{\mathbf{x}^t\}$ generated by* (7) - (9)*. If the conditions in Assumptions 1-4 are satisfied, then for every limit point of the sequence $\{\mathbf{x}^t\}$ generated by the DSSC algorithm, there exists a subsequence that converges to a stationary point of* (1) *almost surely.*

Theorem 1 establishes that a subsequence of the iterates defined by (7)-(9) converge to a stationary solution of (1). This is one of the first convergence guarantees for a method which alleviates the bottleneck in the feature dimension p and sample size N simultaneously for non-convex composite optimization problems. Moreover, almost sure convergence of stochastic methods in non-convex settings often remains elusive, and has been consistently, clandestinely avoided.

4. NUMERICAL ANALYSIS

In this section, we study the performance of the doubly stochastic successive convex approximation methods developed in Section 2. We consider a *perturbed* linear regression problem defined by an *indefinite* observation matrix with an ℓ_1 regularization, which is a simple non-convex problem. The setting is the following: observations $\mathbf{z}_n \in \mathbb{R}^q$ are collected as noisy linear transformations $\mathbf{z}_n = \mathbf{H}_n \mathbf{x} + \mathbf{w}_n$ of an unknown signal $\mathbf{x} \in \mathbb{R}^p$,



(a) Objective $V(\mathbf{x}^t)$ vs. feature $\tilde{p}_t$ (b) Objective $V(\mathbf{x}^t)$ vs. clock time (s) (c) Objective $V(\mathbf{x}^t)$ vs. features $\tilde{p}_t$

Fig. 1: DSSC applied to a large-scale non-convex lasso problem defined by an indefinite observation matrix. We observe that DSSC converges faster in objective evaluation $V(\mathbf{x}^t)$ versus the number of features processed (Fig. 1a) and clock time (Fig. 1b). Moreover, the method demonstrates the benefits of parallel processing: when the number of processors I is increased, we obtain accelerated convergence (Fig. 1c).

and $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 I_q)$ is a Gaussian random variable. For a finite set of samples N, a locally optimal estimator $\mathbf{x}^*$ is the λ-regularized least squares estimate

$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{H}_n\mathbf{x} - \mathbf{z}_n\|^2 - c\|\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1 \ . \tag{14}$$

where the ℓ_1 regularization is introduced to sparsify the regressor, and admits a block decomposition. (14) is non-convex due to the presence of the term $-c\|\mathbf{x}\|^2$, provided $c > \lambda_{\max}(\mathbf{H}^T\mathbf{H})$. (14) is a special case of (1): $F(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{H}_n\mathbf{x} - \mathbf{z}_n\|^2 - c\|\mathbf{x}\|^2$ and $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$.

We run the DSSC method on lasso problem instances where $q = 1$, $p = 1024$, and $N = 10^4$ samples are given. The observation matrices $\mathbf{H}_n \in \mathbb{R}^{q\times p}$, when stacked over all n (an $N \times p$ matrix), are generated from a matrix normal distribution whose mean is a tri-diagonal matrix. The main diagonal is selected uniformly at random from the fractions $\pm\{1, \ldots, p\}/p$, while the super and sub-diagonals are all set to $-1/2$. Observe that this results in $\mathbf{H}_n$ being *indefinite*. Moreover, we set $c = \lambda_{\max}(\mathbf{H}^T\mathbf{H}) + 2$, so that (14) is non-convex. Moreover, the true signal has entries chosen uniformly at random from the fractions $\mathbf{x} \in \{1, \ldots, p\}/p$. Additionally, the noise variance perturbing the observations is set to $\sigma^2 = 10^{-2}$. Initially, we fix the number of processors $I = 16$ is fixed and each processor is in charge of 1 block, and the number of blocks is $B = 64$.

We select the instantaneous convex surrogate for the first term in (14) to be the linearization of the instantaneous non-convex function $\|\mathbf{H}_n\mathbf{x} - \mathbf{z}_n\|^2 - c\|\mathbf{x}\|^2$ around the current block, while omitting the intercept term, i.e. $\tilde{f}_i(\mathbf{x}_i; \mathbf{x}^t_{-i}, \boldsymbol{\theta}^t) = (\mathbf{x}_i - \mathbf{x}^t_i)^T\nabla_{\mathbf{x}_i} f_i(\mathbf{x}^t; \boldsymbol{\theta}^t)$, which results in a parallelized stochastic proximal conditional gradient scheme [26], with a random subset of decision variable coordinates updated at each step. We compare the algorithm proposed in Section 2 for solving non-convex lasso problems to two alternatives: (1) a randomized alternating proximal gradient scheme [27] that uses all samples (mini-batch size $L = N$) to update a random single block with a deterministic gradient at each step (Randomized Block Gradient); and (2) a parallelized stochastic implementation of FISTA [28], i.e., one in which all blocks are updated at each step using stochastic gradients.

These methods process a different amount of information per iteration index t, and thus to obtain a fair comparison, we consider the number of coordinates of the decision variable (features) processed. This may be calculated as $\tilde{p}_t = prtL$, where $r = I/B$ is the proportion of $\mathbf{x}$ updated at each step, p is the length of $\mathbf{x}$, L is the size of the mini-batch, and t is the iteration index. Also considered here due to its unbiased capability to measure algorithm performance is clock time in seconds.

We consider the performance of DSSC when using a diminishing step-size $\rho^t = 5t^{-1/5000}$ and momentum parameter $\gamma^t = 2t^{-1/1000}$. These selections are also used in Stochastic FISTA and Randomized Block Gradient. Further, we set $\tau_i = 2$ for all i, $\lambda = .1$, and for DSSC/FISTA we use mini-batch size $L = 10$. The results of this numerical comparison are given in Figure 1. In particular, we respectively plot the objective $V(\mathbf{x}^t)$ defined by (14) versus feature $\tilde{p}_t$ (Fig. 1a) and clock time (Fig. 1b). Observe that large objective values are a consequence of the ill-conditioning of the problem. Nonetheless, we observe that DSSC converges to stationarity at the fastest rate in terms of features processed as well as clock time. Specifically, to reach approximate stationarity, DSSC requires $\tilde{p}_t = 1658$ features (2 seconds) whereas FISTA requires $\tilde{p}_t > 6000$ features (2.5 seconds) to reach stationarity. We find deterministic Randomized Block Gradient is not comparable.

We now consider the case where the number of blocks is fixed $B = 64$, but the number of processors I varies. The goal of this numerical experiment is to study the benefits of parallel processing (which is equivalent to studying the benefits of updating random subset of blocks at each step rather than all blocks). We maintain the same aforementioned parameter selections (step-size, mini-batch size, regularization). Results are presented in Figure 1c, where we plot the objective (14) versus the number of features processed $\tilde{p}_t$. We observe that increasing the number of processors, or updating *fewer* blocks per step, yields faster convergence in terms of $\tilde{p}_t$. That is, for the benchmark $V(\mathbf{x}^t) \leq 360$, we require $\tilde{p}_t = 813$, $\tilde{p}_t = 1592$, $\tilde{p}_t = 3160$, and $\tilde{p}_t = 6380$, respectively, for $I = 128$, $I = 64$, $I = 32$, and $I = 16$ processors. The accelerated convergence of stochastic methods which use *less information* per step has been empirically observed for parallel Quasi-Newton schemes previously in [29], and comes from the advantages of Gauss-Seidel style block selection schemes in block coordinate methods as compared with Jacobi schemes. Specifically, we note that for certain problems settings, cyclic block updates converge twice as fast as parallel schemes. We interpret DSSC with more processors as comparable to Gauss-Seidel style updates, whereas stochastic FISTA is a Jacobi scheme. We note that the magnitude of this gain is dependent on the condition number of the Hessian of the expected objective $F(\mathbf{x})$, and the difficulty of computing proximal operators associated with $G(\mathbf{x})$.

5. CONCLUSIONS

We proposed the first effort towards solving non-convex regularized expected risk minimization associated with training sets whose sample size and feature dimension are large. We adopted an approach that synthesizes the benefits of successive convex approximation, stochastic approximation, and block coordinate methods. In particular, we decompose the decision variables into blocks, replace the non-convex average objective with a strongly convex block-wise sample surrogate function that we minimize at each step. We established almost sure convergence in infimum to stationary solutions of the original nonconvex stochastic minimization. Moreover, we demonstrated favorable numerical properties on (nonconvex) a lasso problem with an indefinite observation matrix.

6. REFERENCES

[1] V. Vapnik, *The nature of statistical learning theory*, 2nd ed. springer, 1999.

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

[3] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, "On optimization methods for deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 265–272.

[4] R. D. Nowak, S. J. Wright *et al.*, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of selected topics in signal processing*, vol. 1, no. 4, pp. 586–597, 2007.

[5] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] 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.

[8] 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.

[9] 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

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

[11] A. Bordes, L. Bottou, and P. Gallinari, "SGD-QN: Careful quasi-Newton stochastic gradient descent," *The Journal of Machine Learning Research*, vol. 10, pp. 1737–1754, 2009.

[12] N. N. Schraudolph, J. Yu, and S. Günter, "A stochastic quasi-Newton method for online convex optimization," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 436–443.

[13] A. Mokhtari and A. Ribeiro, "RES: Regularized stochastic BFGS algorithm," *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6089–6104, 2014.

[14] ——, "Global convergence of online limited memory BFGS," *Journal of Machine Learning Research*, vol. 16, pp. 3151–3181, 2015.

[15] 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.

[16] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.

[17] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[18] C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin, "Feature clustering for accelerating parallel coordinate descent," in *Advances in Neural Information Processing Systems*, 2012, pp. 28–36.

[19] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, pp. 1–52, 2015.

[20] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2011, pp. 693–701.

[21] 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*, 2016. [Online]. Available: https://fling.seas.upenn.edu/~akoppel/assets/papers/dssc_report.pdf

[22] 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.

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[24] 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.

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

[26] B. T. Polyak, *Introduction to optimization*. Optimization Software New York, 1987.

[27] S. Ma, "Alternating proximal gradient method for convex minimization," *Journal of Scientific Computing*, pp. 1–27.

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

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