An Effective Hard Thresholding Method Based on Stochastic Variance Reduction for Nonconvex Sparse Learning

Guannan Liang, Qianqian Tong, Chunjiang Zhu, Jinbo Bi
Department of Computer Science and Engineering,
University of Connecticut, Storrs, CT, USA
{guannan.liang, qianqian.tong, chunjiang.zhu, jinbo.bi}@uconn.edu

Abstract

We propose a hard thresholding method based on stochastically controlled stochastic gradients (SCSG-HT) to solve a family of sparsity-constrained empirical risk minimization problems. The SCSG-HT uses batch gradients where batch size is pre-determined by the desirable precision tolerance rather than full gradients to reduce the variance in stochastic gradients. It also employs the geometric distribution to determine the number of loops per epoch. We prove that, similar to the latest methods based on stochastic gradient descent or stochastic variance reduction methods, SCSG-HT enjoys a linear convergence rate. However, SCSG-HT now has a strong guarantee to recover the optimal sparse estimator. The computational complexity of SCSG-HT is independent of sample size n when n is larger than $\frac{1}{\epsilon}$, which enhances the scalability to massive-scale problems. Empirical results demonstrate that SCSG-HT outperforms several competitors and decreases the objective value the most with the same computational costs.

Introduction

We consider the following sparsity-constrained empirical risk minimization (ERM) problems, which have been widely used in high-dimensional data analyses (Donoho and others 2006; Tropp and Gilbert 2007; Bahmani, Raj, and Boufounos 2013; Jalali, Johnson, and Ravikumar 2011),

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{z=1}^{n} f_z(x) \quad \text{subject to} \quad \|x\|_0 \le k, \qquad (1)$$

where $f(x)$ is a smooth and (non-strongly) convex function, $f_z(x)$ ($z \in [n] := \{1, 2, \dots, n\}$) is an individual loss associated with the z^{th} sample, $\|x\|_0$ denotes the l_0-norm of a vector, i.e., computes the number of nonzero entries in x, and the integer k is the required sparsity parameter. Problem (1) plays an essential role in many statistical learning, machine learning, and signal processing problems.

Due to the non-convexity of the cardinality constraint, finding a solution to Problem (1) is generally NP-hard (Natarajan 1995). The l_0-constrained linear regression problems or those with a quadratic loss function gained significant attention first. Many greedy-based algorithms have been developed to solve the problems such as matching pursuit (Mallat and Zhang 1993), orthogonal matching pursuit (Pati, Rezaiifar, and Krishnaprasad 1993), compressive sampling matching pursuit (Needell and Tropp 2009), hard thresholding pursuit (Foucart 2011), iterative hard thresholding (Blumensath and Davies 2009) and subspace pursuit (Dai and Milenkovic 2009). These algorithms largely fall into the regimes of either matching pursuit methods or iterative hard thresholding (HT) methods.

For an arbitrary loss function (not restricted to quadratic functions), (Bahmani, Raj, and Boufounos 2013) proposed a greedy algorithm called Gradient Support Pursuit (GraSP). However, GradSP requires to find an optimal solution to *argmin* $f(x)$ over the identified support after thresholding, which does not have analytical solutions for an arbitrary loss, and thus could be time-consuming. For the general form of Problem (1), matching pursuit methods encounter the same issue as GraSP. Later, coordinate-wise algorithms and block decomposition algorithms also were developed (Patrascu and Necoara 2015; Yuan, Shen, and Zheng 2019). However, they may either cycle indefinitely, if the minimization step has multiple solutions, or need to solve a subproblem globally using combinatorial search methods at each iteration, which may fail for very large sparsity k. Hence, iterative gradient-based HT methods have gained significant interest and become popular for nonconvex sparse learning.

Iterative HT methods include gradient descent HT (GD-HT) method (Jain, Tewari, and Kar 2014), stochastic gradient descent HT (SG-HT) method (Nguyen, Needell, and Woolf 2017), hybrid stochastic gradient HT (HSG-HT) method (Zhou, Yuan, and Feng 2018), and stochastic variance reduced gradient HT (SVRG-HT) method (Li et al. 2016b). These algorithms update the parameter iterate x^t via gradient descent or its variants, and then apply the HT operator to enforce sparsity of x. The computation can be concisely written as $x^{t+1} = \mathcal{H}_k(x^t - \eta v^t)$, where η is the learning rate, v^t can be the full gradient, stochastic gradient or variance reduced gradient at the t^{th} iteration, and $\mathcal{H}_k(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ denotes the HT operator that preserves the top k elements in x and sets other elements to 0.

Importantly, we point out that the computational complexity of iterative HT methods mainly consists of two parts:

Table 1: Comparisons of different iterative hard thresholding algorithms for Problem (1) where $\kappa_s = \frac{L_s}{\rho_s}$ is the restricted condition number with step size η and $s = 2k + k^*$. The number of IFO, number of hard thresholding and estimation error are calculated based on the analysis of parameter estimation error, i.e., $\|x^t - x^*\|$, between the $k-$sparse iterate x^t at iteration t and the optimal solution x^* to Problem (1). Estimation error function $g(x^*)$ is the residual term, which is determined by $\nabla f(x^*)$.

Algorithm	Reference	Constraint on κ_s	Constraint on ρ_s	# of IFO	# of Hard Thresholding	Estimation Error $g(x^*)$
GD-HT	(Yuan, Li, and Zhang 2014)	No	No	$O(n\kappa_s \log(\frac{1}{\epsilon}))$	$O(\kappa_s \log(\frac{1}{\epsilon}))$	$O(\eta\|\nabla f(x^*)\|)$
SG-HT	(Nguyen, Needell, and Woolf 2017)	$\leq \frac{4}{3}$	No	$O(\kappa_s \log(\frac{1}{\epsilon}))$	$O(\kappa_s \log(\frac{1}{\epsilon}))$	$O(\frac{1}{n}\sum_{z=1}^{n} \|\nabla f_z(x^*)\|)$
HSG-HT	(Zhou, Yuan, and Feng 2018)	No	Yes[1]	$O(\frac{\kappa_s}{\rho_s \epsilon})$	$O(\kappa_s \log(\frac{1}{\epsilon}))$	$O(\|\pi_{\tilde{\mathcal{I}}}(\nabla f (x^*))\|)$
SVRG-HT	(Li et al. 2016b)	No	No	$O((n + \kappa_s) \log(\frac{1}{\epsilon}))$	$O((n + \kappa_s) \log(\frac{1}{\epsilon}))$	$O(\sqrt{s}\|\nabla f(x^*)\|_\infty)$
SCSG-HT	Ours	No	No	$O(min\{n, \frac{1}{\epsilon}\} \kappa_s \log(\frac{1}{\epsilon}))$	$O(min\{n, \frac{1}{\epsilon}\} \kappa_s \log(\frac{1}{\epsilon}))$	$O(\sqrt{\eta}\|\pi_{\tilde{\mathcal{I}}} (\nabla f(x^*))\|)$

[1] The constraint of ρ_s is for asymptotic IFO complexity.

i) the computation of gradients and ii) the operation of hard thresholding, which are represented by the number of IFO calls (see Definition 2) and the number of HT operations, respectively. The number of IFO calls takes more weight than the number of HT operations in complexity analysis for Problem (1) because gradients are more expensive to compute than sorting a vector to take the top k elements.

It has been proved that the sequence of parameter iterates x^t generated by the GD-HT can approximate the optimal sparse solution with an arbitrary precision tolerance. As shown in Table 1, the expected error between the iterate x^t and the optimal solution x^* can be bounded in the order of $\eta\|\nabla f(x^*)\|$. Although $\|\nabla f(x^*)\|$ is a fixed nonzero term due to the sparsity constraint on x^*, η can be chosen small. However, its computational complexity depends on the sample size n which makes the algorithm hard to scale. Stochastic versions of the GD-HT method, such as SG-HT and HSG-HT, and SVRG-HT, have then been developed to improve computational efficiency, but a fixed term based on the norm of $\|\nabla f(x^*)\|$ is introduced to the bound of estimation error. The algorithms no longer guarantee to produce solutions *arbitrarily close* to the optimal solution. There are also other problems in the convergence analysis of these algorithms. For instance, the SG-HT assumes that the condition number κ_s of the objective function is very small ($\leq \frac{4}{3}$), which is hardly satisfied in practice. For a restricted ρ_s-strongly convex objective function f, the computational complexity of HSG-HT is proportional to $\frac{1}{\rho_s}$, so it requires ρ_s to be relatively large. The sample size n comes back to affect the IFO complexity of the SVRG-HT.

We propose to use stochastically controlled stochastic gradients (SCSG) in the HT method. Our **main contributions** are summarized as follows.

- It is the first time that SCSG methods are incorporated into a HT operator which shows clear advantages over the state of the art. It uses batch gradients to approximate the computationally-expensive full gradients in variance reduction. Although it introduces a bias to the updating direction v^t, we have used new theoretical ingredients to control the balance between this bias and batch size to show a strong convergence result.
- As shown in Table 1, the convergence and complexity analysis of SCSG-HT does not need strong assumptions as required by the SG-HT and HSG-HT methods. More importantly, the step size η comes back to the estimation error bound (similar to the full gradient GD-HT) to achieve better parameter accuracy with a linear convergence rate.
- The computational complexity of SCSG-HT is independent of the sample size n when n is larger than $\frac{1}{\epsilon}$ where ϵ is a pre-chosen error tolerance. Extensive experiments also show that it decreases the objective value fastest among all competitors in both situations when a small ϵ is required (high precision regime) and when ϵ can be bigger (medium precision regime).

Related Work

The GD-HT methods have been comprehensively studied in compressed sensing community and sparse learning community (Blumensath and Davies 2009; Foucart 2011; Yuan, Li, and Zhang 2014; Jain, Tewari, and Kar 2014; Garg and Khandekar 2009). At the t^{th} iteration, $x^{t+1} = \mathcal{H}_k(x^t - \eta\nabla f(x^t))$ where $\nabla f(x^t)$ is the full gradient. For GD-HT methods, linear convergence rate to the optimal solution x^* can be guaranteed with an arbitrary estimation accuracy (Yuan, Li, and Zhang 2014; Jain, Tewari, and Kar 2014). Compared with convex relaxation of the cardinality constraint, such as imposing l_1-norm of $\|x\|_1 \leq \delta$ (Tibshirani 1996; Van de Geer and others 2008), solving Problem (1) via GD-HT methods often achieves comparable empirical performance and can be more computationally efficient.

However, despite these desirable properties, GD-HT methods still need to compute the full gradient at each iteration and thus can be difficult to scale with large datasets.

The SG-HT methods have been recently proposed to improve computational complexity utilizing the finite-sum structure of Problem (1) (Nguyen, Needell, and Woolf 2017; Li et al. 2016a; Zhou, Yuan, and Feng 2018). At the t^{th} iteration, $x^{t+1} = \mathcal{H}_k(x^t - \eta \nabla f_z(x^t))$, where $\nabla f_z(x^t)$ is the stochastic gradient computed on sample z. Even though the computational complexity of SG-HT is independent of sample size n, it requires the restricted condition number κ_s to be $\leq \frac{4}{3}$, which is a strong constraint and is hard to satisfy in real-life high-dimensional problems. To overcome this issue, (Zhou, Yuan, and Feng 2018) proposes **HSG-HT** algorithm, which increases mini-batch size over iterations and successfully removes its dependence on the restricted condition number, at the cost that the number of IFO calls for gradient computation is linearly dependent on $\frac{1}{\epsilon}$ instead of $\log(\frac{1}{\epsilon})$ in the SG-HT algorithm, and is also linearly dependent on $\frac{1}{\rho_s}$, where ρ_s can be difficult to control in practice.

The SVRG-HT method has been designed for sparse learning inspired by the success of SVRG methods (Li et al. 2016b). Variance reduction methods have been extensively studied for convex optimization, such as, the SVRG (Johnson and Zhang 2013) and stochastic average gradient (SAGA) (Defazio, Bach, and Lacoste-Julien 2014) methods, are well known for their fast convergence. In nonconvex optimization, variance reduction methods have been proved to converge to first-order stationary points (Reddi et al. 2016; Lei et al. 2017). Benefiting from the variance reduction technique, the SVRG-HT method can converge more stably and efficiently with a higher estimation accuracy than SG-HT methods. Unlike SG-HT methods, the convergence analysis for the SVRG-HT method allows an arbitrary bounded restricted condition number. Although the IFO complexity of SVRG-HT is substantially improved over SG-HT methods, the overall complexity still scales linearly with respect to the sample size n. Therefore, for large-scale datasets, the SVRG-HT method may still suffer.

Preliminaries

We use lowercase letters, e.g. x, to denote a vector and use $\|\cdot\|$ to denote the l_2-norm of a vector. In this paper, the notations $O(\cdot)$ and $\Omega(\cdot)$ are asymptotic upper bounds and asymptotic lower bounds respectively. The operator $E[\cdot]$ represents taking expectation over all random variables, $[n]$ denotes the integer set $\{1, ..., n\}$, $\nabla f(\cdot)$, $\nabla f_I(\cdot)$ and $\nabla f_z(\cdot)$ are the full gradient, the stochastic gradient over a minibatch $I \subset [n]$ and the stochastic gradient over a training example indexed by $z \in [n]$, respectively, and $\mathbb{I}(\cdot)$ is an indicator function. The notation $supp(x)$ means the support of x or the index set of non-zero elements in x. The support $\mathcal{I}_{t+1}^{(j)} = supp(x^*) \cup supp(x_t^{(j)}) \cup supp(x_{t+1}^{(j)})$, is associated with the $t+1$ iteration at the j^{th} epoch (and $\mathcal{I}$ is used throughout the paper without ambiguity). The projector $\pi_{\mathcal{I}}(x)$ takes only the elements of x indexed in $\mathcal{I}$.

Definition 1. *A random variable N follows a geometric distribution $Geom(\gamma)$, denoted as $N \sim Geom(\gamma)$, if N is a non-negative integer and the probability distribution is*

$$P(N = k) = (1 - \gamma)\gamma^k$$

for any $k = 0, 1, \cdots$. Then, we know $E[N] = \frac{\gamma}{1-\gamma}$.

Definition 2. *(Agarwal and Bottou 2014) (Incremental First-order Oracle (IFO)) An IFO is a subroutine that takes a point $x \in \mathbb{R}^d$ and an index $z \in [n]$ and returns a pair $(f_z(x), \nabla f_z(x))$.*

The Proposed SCSG-HT Method

In this section, we present our new algorithm – SCSG-HT in Algorithm 1 for solving Problem (1). The SCSG algorithm belongs to the SVRG family, and was first proposed in (Lei and Jordan 2017) that showed competitive time complexity in convex optimization, and was later extended to non-convex optimization in (Lei et al. 2017). Similar to the SVRG method, SCSG method has an outer loop, and each outer iteration also includes an inner loop. The main differences between SCSG and the classic SVRG are the following. Before starting the inner loop, the SCSG calculates batch gradient over small batch $I^{(j)}$ with batch size $B^{(j)}$, whereas the SVRG calculates the full gradient. The number of iterations in the inner loop for the SCSG is stochastically determined by the geometric distribution, rather than a fixed number of $O(n)$ (usually n) used in the SVRG.

Algorithm 1 SCSG-HT

```
Require: Number of outer loops $\mathcal{J}$, initial state $\tilde{x}^1$, stepsize $\eta$, batch size $(B^{(j)})_{j=1}^{\mathcal{J}}$ and mini-batch size $(b^{(j)})_{j=1}^{\mathcal{J}}$
 1: for $j = 1, 2, ..\mathcal{J}$ do
 2:     Uniformly sample a batch $I^{(j)} \subset \{1, ..., n\}$, where $|I^{(j)}| = B^{(j)}$
 3:     $\tilde{\mu}^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)})$
 4:     $x_0^{(j)} = \tilde{x}^{(j)}$
 5:     (Option I) Generate $N^{(j)} \sim$ Geom $(B^{(j)}/(B^{(j)} + b^{(j)}))$
 6:     (Option II) $N^{(j)} = \frac{B^{(j)}}{b^{(j)}}$
 7:     for $t = 1, 2, \ldots, N^{(j)}$ do
 8:         Randomly pick $I_t^{(j)} \subset \{1, ..., n\}$, where $|I_t^{(j)}| = b^{(j)}$
 9:         $x_t^{(j)} = \mathcal{H}_k(x_{t-1}^{(j)} - \eta(\nabla f_{I_t^{(j)}}(x_{t-1}^{(j)}) - \nabla f_{I_t^{(j)}}(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)}))$
10:     end for
11:     set $\tilde{x}^{j+1} = x_{N^{(j)}}^{(j)}$
12: end for
```

The outer loop consists of the steps in Lines 1 - 12 and an inner loop in Lines 7 - 10. The batch gradient is computed in each outer iteration (Lines 3) where batch size $B^{(j)}$ is to be pre-determined. Often times, we set all $B^{(j)} = B$. In each iteration of the inner loop, the mini-batch has a size of $b^{(j)}$ in general which we can also set to be a constant b. We provide two choices to set the number of iterations of the inner loop $N^{(j)}$: in option I (Line 5) $N^{(j)}$ is randomly drawn from a geometric distribution, similar to the SCSG; in option II (Line 6) $N^{(j)}$ is the deterministic constant $\frac{B^{(j)}}{b^{(j)}}$, which is the expectation of the geometric distribution $N^{(j)} \sim$ Geom $(B^{(j)}/(B^{(j)}+b^{(j)}))$. Our theoretical analysis is based on option I, which provides more general results. We observe that the variance of $N^{(j)}$ in option I is larger than option II, and option II is more stable in practice.

A variance reduction step (in Line 9) is commonly used in the SVRG family, and it is performed inside the hard thresholding operator $\mathcal{H}_k(\cdot)$ in Algorithm 1. The algorithm starts from randomly initialized $\tilde{x}^1$. The step size η can take a constant or decay over iterations. In our theoretical analysis, we assume that η is a constant. The maximum number of iterations $\mathcal{J}$ is typically chosen to be large, and for convex f, our theoretical analysis provides guidance on choosing a value for $\mathcal{J}$ based on ϵ.

Theoretical Analysis

In this section, we present our main theoretical results characterizing the estimation error of parameters x and the error of the objective value. We first show that the sparsity recovery can be guaranteed in Thoerem 1. Then, we present a result of convergence performance in terms of objective value in Corollary 1.3.

Throughout the theoretical analyses, we assume that the objective function $f(x)$ satisfies the following assumption, which is commonly used in related works for Problem (1):

Assumption 1. *Assume that the differentiable function $f(x)$ satisfies:*

(i) for given $s \in \mathbb{N}_+$, restricted ρ_s-strongly convex at sparsity level s, i.e., there exists a constant $\rho_s > 0$ s.t. $\forall x_1, x_2 \in \mathbb{R}^d$ with $\|x_1 - x_2\|_0 \leq s$, we have

$$f(x_1) - f(x_2) - \langle \nabla f(x_2), x_1 - x_2 \rangle \geq \frac{\rho_s}{2}\|x_1 - x_2\|^2;$$

(ii) for given $s \in \mathbb{N}_+$, restricted L_s-strongly smooth at sparsity level s, i.e., there exists a constant $L_s > 0$ s.t. $\forall x_1, x_2 \in \mathbb{R}^d$ with $\|x_1 - x_2\|_0 \leq s$, we have

$$f(x_1) - f(x_2) - \langle \nabla f(x_2), x_1 - x_2 \rangle \leq \frac{L_s}{2}\|x_1 - x_2\|^2;$$

(iii) σ^2-bounded for stochastic gradient variance, i.e., $E[\|\nabla f_z(x) - \nabla f(x)\|^2] \leq \sigma^2, \forall x \in \mathbb{R}^d$, where index $z \in [n]$.

The following theorem is our main result on the parameter estimation accuracy of the SCSG-HT for sparsity-constrained problems. Although this paper is focused on the cardinality constraint, the theoretical analysis can be applied to other sparsity constraints such as on matrix rank. Due to page limit, we include a complete proof of Theorem 1 in Appendix and provide a proof sketch in the next section.

Theorem 1. *Let x^* be the optimal of Problem (1), $k^* = \|x^*\|_0$, and suppose $f(x)$ satisfies Assumption 1. Define $\tilde{\mathcal{I}} = supp(x^*) \cup supp(\mathcal{H}_{2k}(f(x^*)))$, the restricted condition number $\kappa_s = \frac{L_s}{\rho_s} \geq 1$, $\eta \leq \frac{1}{32 L_s \kappa_s}$, $\alpha \leq min\{\frac{b}{B}, \frac{1}{64\kappa_s^2 - 1}\}$. Then we can obtain the following result:*

$$E[\|\tilde{x}^{(j+1)} - x^*\|^2] \leq \theta^{(j+1)} E[\|\tilde{x}^{(0)} - x^*\|^2] \\ + 2\gamma E[\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2] + \gamma \frac{\mathbb{I}(B < n)}{B}\sigma^2, \tag{2}$$

where $0 < \theta = 1 - (\eta(\rho_s/2 - (2\rho_s L_s + 14L_s^2)\eta))/\beta < 1$, $\beta = \frac{\frac{b}{B} - \alpha}{1+\alpha} + \eta(\rho_s - 2\rho_s L_s \eta - 4L_s^2 \eta) > 0$, $\alpha = \frac{2\sqrt{k^*}}{\sqrt{k - k^*}}$ and $\gamma = \frac{4\eta^2}{(1-\theta)\beta}$.

Remark 1. *When $\eta = \frac{1}{32 L_s \kappa_s}$, and the sparsity parameter $k \geq max\{(4(\frac{B}{b})^2 + 1)k^*, (4(64\kappa_s - 1)^2 + 1)k^*\}$, we can further increase θ to $\tilde{\theta} = 1 - \frac{1}{\frac{64\kappa_s^2}{1+\alpha}\frac{b}{B} + \frac{13}{8}}$. It is obvious that $0 < \tilde{\theta} < 1$.*

In Theorem 1, the first term on the right hand side of Ineq.(2) approaches to 0 when j increases. When the size of the outer loop batch B takes n (equivalent to the SVRG), the third term becomes 0. Otherwise, the second and third terms both depend on γ. Based on the formula of γ, it depends on η. Hence, when η takes a small value, the last two terms become small. We assume a constant η here, and leave it as our future work to optimize the decay η in our analysis. We further obtain the following corollary that bounds the number of iterations $\mathcal{J}$ to obtain a sub-optimal solution (i.e. the difference between the solution and x^* is bounded only by the second term of Ineq.(2)).

Corollary 1.1. *Assume that the setup in Theorem 1 holds, $\gamma \leq \frac{1}{2\sigma^2}$ and $B = min\{\frac{1}{\epsilon}, n\}$ for a given accuracy $\epsilon > 0$, then we need at most $\mathcal{J} \leq C_1 \log(\frac{4\|\tilde{x}^{(1)} - x^*\|}{\epsilon})$ outer iterations to obtain*

$$E[\|\tilde{x}^{(\mathcal{J})} - x^*\|^2] \leq \epsilon + g_1(x^*),$$

where $C_1 = -(\log(\theta))^{-1}$, $g_1(x^*) = 2\gamma\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2$.

Corollary 1.1 indicates that under proper conditions, the estimation error of the SCSG-HT to the optimal solution x^* is determined by the second term of Ineq.(2) which we denote as $g_1(x^*)$,

$$g_1(x^*) = \frac{8\eta^2}{(1-\theta)\beta}\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2. \tag{3}$$

By setting Ineq.(2) to be less than a given tolerance ϵ, we can see that the convergence rate is linear or geometric before reaching ϵ. After that, the accuracy cannot be further improved due to $g_1(x^*)$. To further explain this, we can examine it from another angle. Once η is fixed (to a value determined by ϵ), $g_1(x^*)$ produces a fixed residual $O(\epsilon)$ to the estimation error. Then after enough iterations (j), the first term of Ineq.(2) drops below this value $O(\epsilon)$, $g_1(x^*)$ becomes dominant and thus further iterations will not help.

Overall, this result guarantees that the iterates approach to the optimal sparse estimator x^* with an arbitrary precision in a finite number of outer iterations. The computational complexity of the SCSG-HT is characterized in the following corollary.

Considering that $C_1 = -(\log(\theta))^{-1}$, where θ is involved with κ_s, we will have the the following result by using the inequality $\log(1+x) \geq \log(2)x$.

Corollary 1.2. *Assume assumptions and setups in the previous corollary hold, the expected number of IFO calls is*

$$\min\{n, \frac{1}{\epsilon}\}(C_1 \log(\frac{2\|\tilde{x}^{(1)} - x^*\|^2}{\epsilon})) = O(\min\{n, \frac{1}{\epsilon}\}\kappa_s \log(\frac{1}{\epsilon})).$$

Based on the above formula, if the target error tolerance ϵ is not too small, then $\frac{1}{\epsilon}$ can be smaller than n especially when large sample is used. In this case, the IFO complexity of the SCSG-HT is no longer dependent on n, an advantage over the GD-HT (Jain, Tewari, and Kar 2014) and SVRG-HT (Li et al. 2016b). Hence, the SCSG-HT can be more scalable than the GD-HT and SVRG-HT with large-scale datasets. Furthermore, the computational complexity of SCSG-HT does not require any constraints on κ_s and ρ_s, which is different from the SG-HT and HSG-HT. More importantly, among all stochastic iterative hard thresholding methods, SCSG-HT is the first one that can effectively diminish the statistical error term $g_1(w^*)$, i.e., making it smaller than a given ϵ. In summary, Theorem 1 and Corollaries 1.1-1.2 guarantee that the SCSG-HT can effectively approximate the optimal estimator x^* with a better computational complexity.

We further investigate the convergence performance in terms of the objective function values $f(x)$ at snapshot $\tilde{x}^{(j+1)}$ in each epoch approaching to the optimal $f(x^*)$.

Corollary 1.3. *Under the same assumptions and setup of parameters in Theorem 1, we can obtain the following convergence result for $f(\tilde{x}^{(j)})$:*

$$E[f(\tilde{x}^{(j+1)}) - f(x^*)] \leq \theta^{j+1}\Delta + g_2(x^*) + g_3(B),$$

where $\Delta = (\frac{1}{2\sqrt{\gamma}} + \frac{L_s}{2})E[\|\tilde{x}^{(0)} - x^*\|^2]$, $g_2(x^*) = (\frac{3}{2}\sqrt{\gamma} + L_s\gamma)E[\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2]$ and $g_3(B) = (\frac{\sqrt{\gamma}}{2} + \frac{L_s\gamma}{2})\frac{\mathbb{I}(B<n)}{B}\sigma^2$.

The convergence of the objective function value to the optimal value is controlled by a linear convergence term $\theta^{j+1}\Delta$, a multiplier of $\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2$ term $g_2(w^*)$, and a batch size B-related term $g_3(B)$. Again, because the last two terms can be controlled by the step size η and batch size B respectively, it results in linear convergence as well for the objective value.

Proof Sketch for Convergence Analysis

Before diving into the detailed proof, we first analyze the bias term $e^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)}) - \nabla f(\tilde{x}^{(j)})$ introduced by the batch variance reduction, which is one of the main differences from the commonly used variance reduction technique and plays an important role in the theoretical analysis. We show that the variance of $e^{(j)}$ can diminish to zero with an increasing batch size B, which gives extra flexibility to adaptively adjust the batch size B based on the target accuracy ϵ.

Lemma 2. *Assume that $v_t^{(j)} = \nabla f_{I_t^{(j)}}(x_t^{(j)}) - \nabla f_{I_t^{(j)}}(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)}$ is the updating direction at the t^{th} iteration of the j^{th} epoch in Algorithm 1, $e^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)}) - \nabla f(\tilde{x}^{(j)})$ is the bias of the updating direction $v_t^{(j)}$, where $E_{I_t^j}[v_t^{(j)}] = \nabla f(x_t^{(j)}) + e^{(j)}$. Then we can bound $E_{I_t^{(j)}}[\|e^{(j)}\|^2]$ as follows:*

$$E[\|\pi_{\mathcal{I}}(e^{(j)})\|^2] \leq 2L_s^2\frac{\mathbb{I}(B<n)}{B}E[\|\tilde{x}^{(j)} - x^*\|^2] + 2\frac{\mathbb{I}(B<n)}{B}\sigma^2.$$

In order to analyze the SCSG-HT algorithm, we further derive the following result:

$$\begin{aligned}
&E_{I_t^{(j)}}[\|\tilde{x}_{t+1}^{(j)} - x^*\|^2] \\
&= E_{I_t^{(j)}}[\|x_t^{(j)} - x^*\|^2] + \eta^2 E_{I_t^{(j)}}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] \\
&- 2\eta\langle\pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^*\rangle - 2\eta\langle\pi_{\mathcal{I}}(e^{(j)}), x_t^{(j)} - x^*\rangle
\end{aligned}$$

where $\tilde{x}_{t+1}^{(j)} = x_t^{(j)} - \eta\pi_{\mathcal{I}}(v_t^{(j)})$ is an intermediate state of the estimator to bridge the analysis between the gradient-based updating step and the hard thresholding step. Then the hard thresholding operation $x_{t+1}^{(j)} = \mathcal{H}_k(\tilde{x}_{t+1}^{(j)})$ immediately follows and we can get $x_{t+1}^{(j)} = \mathcal{H}_k(x_{t-1}^{(j)} - v_t^{(j)})$ due to $\mathcal{I} = supp(x^*) \cup supp(x_t^{(j)}) \cup supp(x_{t+1}^{(j)})$. Next, we will establish connections between the intermediate state $\tilde{x}_{t+1}^{(j)}$ and the sparse estimator $x_{t+1}^{(j)}$. The following lemma can be obtained:

Lemma 3. *(Li et al. 2016a) For $k > k^*$ and for any parameter $x \in \mathbb{R}^d$, we have*

$$\|\mathcal{H}_k(x) - x^*\|^2 \leq (1+\alpha)\|x - x^*\|^2$$

where $k^* = \|x^*\|_0$ and $\alpha = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}}$.

With all above results, the relation between $x_{t+1}^{(j)}$ and $x_t^{(j)}$ can be established as follows:

$$\begin{aligned}
&E^{(j)}[\|x_{t+1}^{(j)} - x^*\|^2] \leq (1+\alpha)E^{(j)}[\|\tilde{x}_{t+1}^{(j)} - x^*\|^2] \\
&\leq (1+\alpha)E^{(j)}[\|x_t^{(j)} - x^*\|^2] + (1+\alpha)\eta^2 E^{(j)}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] \\
&- 2(1+\alpha)\eta E^{(j)}[\langle\pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^*\rangle] \qquad (4)
\end{aligned}$$

where $E^j[\cdot]$ is the expectation over the j^{th} epoch, in other words, over all randomness generated by $\{I^{(j)}, I_0^{(j)}, I_1^{(j)}, ...\}$.

Until now, all the analyses are still based on iterations in one epoch. Next, we need to use an important property of the geometric distribution that we have used to set the number of inner iterations $N^{(j)}$ to turn previous iteration-based analysis into epoch-based analysis.

Lemma 4. *Let $N \sim Geom(\gamma)$. Then for any sequence $\{D_N\}$, we have*

$$E[D_N - D_{N+1}] = (\frac{1}{\gamma} - 1)(D_0 - E[D_N]).$$

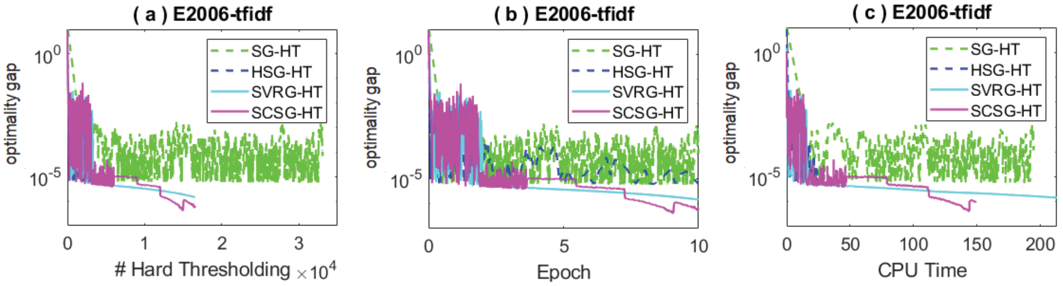

Figure 1: Comparisons among the different HT algorithms for the the sparse linear regression problem. We first run an algorithm long enough (e.g., 100 epochs) to get a high accuracy which gives the lower bound f^*. In view of the $f(x) - f^*$, our SCSG-HT achieves the highest accuracy in 10 epochs in comparison with the SG-HT, HSG-HT and SVRG-HT.

Reorganizing the inequality (4) and taking the expectation over $N^{(j)}$ yield :

$$\begin{aligned} &2(1+\alpha)\eta E[\langle \pi_{\mathcal{I}}(\nabla f(\tilde{x}^{(j+1)})), \tilde{x}^{(j+1)} - x^*\rangle] \\ &\leq (\alpha - \frac{b}{B})E[\|\tilde{x}^{(j+1)} - x^*\|^2] + \frac{b}{B}E[\|\tilde{x}^{(j)} - x^*\|^2] \\ &+ (1+\alpha)\eta^2 E[\|\pi_{\mathcal{I}}(v_{N^{(j)}}^{(j)})\|^2] \end{aligned} \tag{5}$$

We now need to further bound $E[\|\pi_{\mathcal{I}}(v_{N^{(j)}}^{(j)})\|^2]$ in the inequality (5) :

$$\begin{aligned} E_{I_t^{(j)}}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] \leq{}& 4L_s(f(x_0^{(j)}) - f(x^*)) \\ &+ 4L_s(\langle \pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^*\rangle) + 2\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2 \\ &+ 2\|\pi_{\mathcal{I}}(\nabla f(x_t^{(j)}))\|^2 + 2L_s^2\|x_0^{(j)} - x^*\|^2 + 2\|\pi_{\mathcal{I}}(e^{(j)})\|^2. \end{aligned}$$

Then we obtain an important intermediate result, which will be used shortly:

$$\begin{aligned} &2(1-2L_s\eta)\eta E[\langle \pi_{\mathcal{I}}(\nabla f(\tilde{x}^{(j+1)})), \tilde{x}^{(j+1)} - x^*\rangle] \\ &+ (\frac{1}{1+\alpha}\frac{b}{B} - \frac{\alpha}{1+\alpha})E[\|\tilde{x}^{(j+1)} - x^*\|^2] \\ &\leq -4L_s\eta^2 E[f(\tilde{x}^{(j+1)}) - f(x^*)] + 4L_s\eta^2 E[f(\tilde{x}^{(j)}) - f(x^*)] \\ &+ (\frac{1}{1+\alpha}\frac{b}{B} + 2L_s^2\eta^2)E[\|\tilde{x}^{(j)} - x^*\|^2] + 2\eta^2 E[\|\pi_{\mathcal{I}}(e^{(j)})\|^2] \\ &+ 2\eta^2 E[\|\pi_{\mathcal{I}}(\nabla f(\tilde{x}^{(j+1)}))\|^2] + 2\eta^2 E[\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2]. \end{aligned} \tag{6}$$

Guarantee of Sparse Estimator Recovery. Under the condition that f is ρ_s-restricted strongly convex and L_s-restricted strongly smooth, we obtain that:

(i) $\langle \pi_{\mathcal{I}}(\nabla f(\tilde{x}^{(j+1)})), \tilde{x}^{(j+1)} - x^*\rangle \geq \frac{\rho_s}{2}\|\tilde{x}^{(j+1)} - x^*\|^2$;

(ii) $f(\tilde{x}^{(j)}) - f(x^*) \leq \frac{1}{2L_s}\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2 + L_s\|\tilde{x}^{(j)} - x^*\|^2$.

Combining the Ineq.(6) with (i) and (ii), we obtain the final result stated in Theorem 1:

$$\begin{aligned} E[\|\tilde{x}^{(j+1)} - x^*\|^2] \leq{}& \theta^{j+1}E[\|\tilde{x}^{(1)} - x^*\|^2] \\ &+ 2\gamma E[\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2] + \gamma\frac{\mathbb{I}(B<n)}{B}\sigma^2. \end{aligned}$$

Guarantee of Convergence. We first use the L_s-restricted strongly smooth condition to establish epoch-based convergence of $f(\tilde{x}^{(j+1)}) - f(x^*)$. Then, we take expectation, substitute the upper bound of $E[\|\tilde{x}^{(j+1)} - x^*\|^2]$ to arrive the convergence result in Corollary 1.3.

Experiments

We compare the proposed algorithm SCSG-HT with the state-of-the-art stochastic sparsity-constraint methods: SG-HT, HSG-HT and SVRG-HT, in our experiments to demonstrate the improved performance and advantage of the SCSG-HT. Following the convention in the stochastic optimization and sparse learning literature, we use the number of IFO per epoch and the number of HT operations to measure the computational complexity. This can make the computational complexity independent of actual implementation of an algorithm. For a comprehensive comparison, we have also included the actual algorithm run time. Five benchmark datasets are used for evaluations: E2006-tfidf, rcv1, real-sim, mnist and news20, all of which can be downloaded from the LibSVM website[1]. In the experiments, parameters B, b and η are determined by the following criteria. Based on Corollary 1.1, $B = \min\{1/\epsilon, n\}$. For a moderate ϵ, e.g. 10^{-3} used on large datasets with a large value of n, B can take the value of $1/\epsilon$. If the data are homogeneous, B can be even smaller. For the inner loop batch size b, it can be 1 for small datasets, or a large value, e.g. 10, for large datasets. The stepsize η for each algorithm is set by a grid search from $\{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. All the algorithms are initialized with $x^{(0)} = 0$.

High Precision Regime. We first conduct experiments on the linear regression problem on the E2006-tfidf dataset with dimension 3308 × 150360,

$$\underset{x}{min}\{f(x) = \frac{1}{n}\sum_{i=1}^{n}\|y_i - z_i^T x\|^2\} \quad \text{subject to} \quad \|x\|_0 \leq k,$$

to check the performance of the proposed SCSG-HT method for achieving high precision solution. In the experiments, we set the sparsity parameter $k = 200$. In Figure 1 (a), HSG-HT uses the smallest number of IFO to achieve the accuracy

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

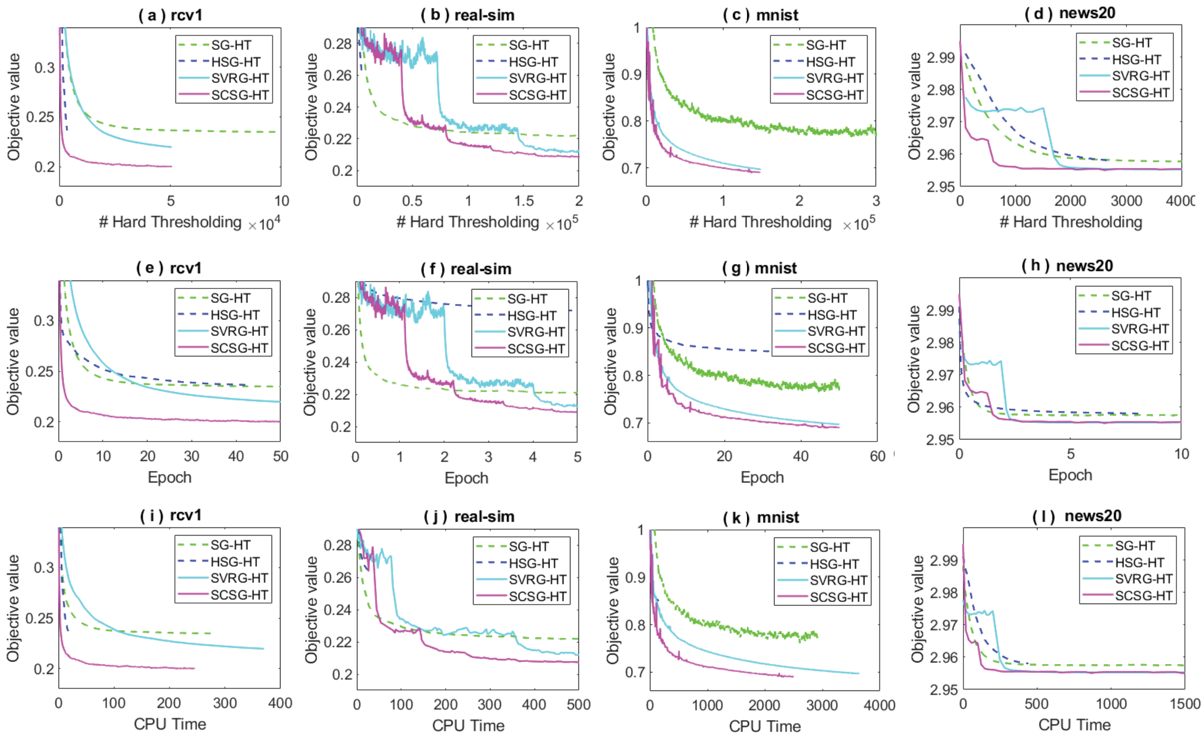

Figure 2: Comparisons of the different HT algorithms on the sparse logistic regression problem and the sparse multi-class softmax regression problem. The first row is for the number of HT operations, the second row is for the number of epochs, and the last row is for the CPU time. Each column in the figure is for a different benchmark dataset.

10^{-5}, but it is not computationally efficient in later stage due to the growth of batch size. In Figure 1 (b), we observe that the variance of $f(x)$ in SCSG-HT is larger than that of SVRG-HT from epoch 2 to 4, because the batch size B in SCSG-HT is smaller than n. However, we find that SCSG-HT can quickly drop function value later on and find a better sparse parameter vector x, which has a smaller optimality gap. This may be due to the extra stochasticity introduced by the biased variance-reduced gradients, which helps our algorithm to jump out of bad local minimal. Overall, SCSG-HT achieved the best performance in terms of the number of IFO calls to achieve a high accuracy, $\epsilon = 10^{-8}$.

Medium Precision Regime. In this set of experiments, we apply HT methods to the logistic regression problem as follows on the rcv1 and real-sim datasets with dimension $20,242 \times 47,236$ and $72,309 \times 20,958$ respectively.

$$\min_{x}\{f(x) = \frac{1}{n}\sum_{i=1}^{n}(\log(1 + exp(y_i z_i^T x)) + \frac{\lambda}{2}\|x\|^2)\}$$

$$\text{subject to} \quad \|x\|_0 \le k,$$

where $z_i \in \mathbb{R}^d$ and y_i is the corresponding label. For both databsets, the regularizer $\lambda = 10^{-5}$ and the sparsity parameter $k = 1000$. We then test HT algorithms on the multi-class softmax regression problem as follows on the mnist and news20 datasets with dimension $60,000 \times 780$ and $15,935 \times 62,061$, respectively. For the mnist dataset, we set $\lambda = 10^{-5}$ and $k = 200$, and for the news20 dataset, $\lambda = 0.01$ and $k = 2000$.

$$\min_{x}\{f(x) = \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{c}(-\mathbb{I}(y_i = j)\log(\frac{\exp(z_i^T x_j)}{\sum_{l=1}^{c}\exp(z_i^T x_l)})$$

$$+ \frac{\lambda}{2}\|x_j\|^2))\}, \quad \text{subject to} \quad \|x_j\|_0 \le k, \quad \forall j \in \{1, 2, ..., l\}.$$

Figure 2 presents the learning curves of the empirical loss versus the number of HT operations, the number of epochs and the CPU time. For all four datasets, SCSG-HT has consistently achieved the lowest objective value with respect to the same number of HT operations, the number of epoches and the CPU time. In general, it also consistently uses the smallest number of HT operations to achieve the target accuracy as shown in the first row in Figure 2. It is consistently the fastest algorithm to achieve the target accuracy, as shown in the second and third rows in Figure 2. For the logitstic loss function on the real-sim dataset in the second column in Figure 2, SG-HT performs the best at the beginning stage, but it fails to continue to reduce the loss function after reaching the value 0.22. HSG-HT is not IFO-efficient since its batch size increases to a very large number, resulting in worse performance than SG-HT. Although SCSG-HT may get trapped at a bad support for a while, it can eventually find the right support and achieve the best function value.

Conclusion

We have proposed a stochastic gradient-based hard thresholding algorithm, which we name as SCSG-HT. We use a batch variance reduction technique to replace computationally-expensive full gradient and a geometric distribution technique to choose the number of iterations in the inner loop. The proposed SCSG-HT significantly improves HT algorithms in terms of computational performance. Without constraints on the restricted condition number κ_s and restricted strongly convex number ρ_s of the objective function, we are able to show that the SCSG-HT enjoys linear convergence and its computational complexities are sample-size-independent for large-scale sparsity-constrained problems, where sample size n is commonly larger than $\frac{1}{\epsilon}$. Empirically, we have compared SCSG-HT with several representative greedy iterative HT algorithms. Overall, our SCSG-HT method outperforms these strong competitors in both theoretical results and empirical evaluations. In our future work, we will study the feasibility of using varied batch size $B^{(j)}$, e.g., an increasing sequence in $[n]$, or the decaying step size η, and see how they will affect the computational performance of SCSG-HT in practice.

Acknowledgments

This work was funded by NSF grants CCF-1514357, DBI-1356655, and IIS-1718738 to Jinbo Bi, who was also supported by NIH grants K02-DA043063 and R01-DA037349.

References

Agarwal, A., and Bottou, L. 2014. A lower bound for the optimization of finite sums. *arXiv preprint arXiv:1410.0723*.

Bahmani, S.; Raj, B.; and Boufounos, P. T. 2013. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research* 14(Mar):807–841.

Blumensath, T., and Davies, M. E. 2009. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis* 27(3):265–274.

Dai, W., and Milenkovic, O. 2009. Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory* 55(5):2230–2249.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, 1646–1654.

Donoho, D. L., et al. 2006. Compressed sensing. *IEEE Transactions on information theory* 52(4):1289–1306.

Foucart, S. 2011. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis* 49(6):2543–2563.

Garg, R., and Khandekar, R. 2009. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *ICML*, volume 9, 337–344.

Jain, P.; Tewari, A.; and Kar, P. 2014. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, 685–693.

Jalali, A.; Johnson, C. C.; and Ravikumar, P. K. 2011. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems*, 1935–1943.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, 315–323.

Lei, L., and Jordan, M. 2017. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, 148–156.

Lei, L.; Ju, C.; Chen, J.; and Jordan, M. I. 2017. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, 2348–2358.

Li, X.; Arora, R.; Liu, H.; Haupt, J.; and Zhao, T. 2016a. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*.

Li, X.; Zhao, T.; Arora, R.; Liu, H.; and Haupt, J. 2016b. Stochastic variance reduced optimization for nonconvex sparse learning. In *International Conference on Machine Learning*, 917–925.

Mallat, S. G., and Zhang, Z. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing* 41(12):3397–3415.

Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM journal on computing* 24(2):227–234.

Needell, D., and Tropp, J. A. 2009. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis* 26(3):301–321.

Nguyen, N.; Needell, D.; and Woolf, T. 2017. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory* 63(11):6869–6895.

Pati, Y. C.; Rezaiifar, R.; and Krishnaprasad, P. S. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, 40–44. IEEE.

Patrascu, A., and Necoara, I. 2015. Random coordinate descent methods for l_0 regularized convex optimization. *IEEE Transactions on Automatic Control* 60(7):1811–1824.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, 314–323.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288.

Tropp, J. A., and Gilbert, A. C. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory* 53(12):4655–4666.

Van de Geer, S. A., et al. 2008. High-dimensional generalized linear models and the lasso. *The Annals of Statistics* 36(2):614–645.

Yuan, X.; Li, P.; and Zhang, T. 2014. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning*, 127–135.

Yuan, G.; Shen, L.; and Zheng, W.-S. 2019. A block decomposition algorithm for sparse optimization. *arXiv preprint arXiv:1905.11031*.

Zhou, P.; Yuan, X.; and Feng, J. 2018. Efficient stochastic gradient hard thresholding. In *Advances in Neural Information Processing Systems*, 1988–1997.