Accelerated Stochastic Block Coordinate Gradient Descent for Sparsity Constrained Nonconvex Optimization

Jinghui Chen
Department of Systems and Information Engineering
University of Virginia

Quanquan Gu
Department of Systems and Information Engineering
University of Virginia

Abstract

We propose an accelerated stochastic block coordinate descent algorithm for nonconvex optimization under sparsity constraint in the high dimensional regime. The core of our algorithm is leveraging both stochastic partial gradient and full partial gradient restricted to each coordinate block to accelerate the convergence. We prove that the algorithm converges to the unknown true parameter at a linear rate, up to the statistical error of the underlying model. Experiments on both synthetic and real datasets backup our theory.

1 INTRODUCTION

High-dimensional statistics (Bühlmann and Van De Geer, 2011) deals with models in which the number of parameters d is comparable to or even larger than the sample size n. Since it is usually impossible to obtain a consistent estimator when both d and n increase, various types of statistical models with structural assumptions including sparse vectors, sparse matrices, low-rank matrices have been proposed and widely studied. In such a high dimensional regime, a general approach is solving a regularized optimization problem, which consists of a loss function measuring how well the model fits the data and some penalty function that encourages the assumed structures. For an overview of high dimensional statistics, please refer to Bühlmann and Van De Geer (2011); Negahban et al. (2009).

In this paper, instead of considering regularized estimator, we focus on the following sparsity constrained optimization problem:

$$\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq s, \tag{1.1}$$

where $F(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^{n} f_i(\boldsymbol{\beta})$ is a sum of a finite number of convex and smooth functions, $\|\boldsymbol{\beta}\|_0$ is the number of nonzero elements in $\boldsymbol{\beta}$, and s is a tuning parameter that controls the sparsity of $\boldsymbol{\beta}$. The above problem is common in machine learning and statistics, such as the empirical risk minimization (ERM) and M-estimator, where $F(\boldsymbol{\beta})$ is the empirical loss function averaged over the training sample. For example, by choosing the squared loss $f_i(\boldsymbol{\beta}) = (\langle\boldsymbol{\beta}, \mathbf{x}_i\rangle - y_i)^2/2$, (1.1) becomes a sparsity constrained linear regression problem (Tropp and Gilbert, 2007).

Due to the nonconvexity of the sparsity constraint, the problem in (1.1) is in general NP hard. In order to obtain an approximate solution to (1.1), a variety of algorithms have been proposed. For example, when the objective function $F(\boldsymbol{\beta})$ is chosen to be the square loss function, it can be solved approximately by matching pursuit (Mallat and Zhang, 1993), orthogonal matching pursuit (Tropp and Gilbert, 2007), CoSaMP (Needell and Tropp, 2009), hard thresholding pursuit (Foucart, 2011), iterative hard thresholding (Blumensath and Davies, 2009) and forward backward feature selection algorithm (Zhang, 2011). For general loss functions, there also exists a set of algorithms such as forward feature selection (Shalev-Shwartz et al., 2010; Bahmani et al., 2013), forward backward feature selection algorithm (Jalali et al., 2011; Liu et al., 2013) and iterative gradient hard thresholding (Yuan et al., 2013; Jain et al., 2014). However, all the above algorithms are based on deterministic optimization such as gradient descent algorithm. In each iteration of gradient descent algorithm, it requires the evaluation of the full gradient over the n component functions, which is computationally very expensive, especially when n is large. In order to address this issue, Nguyen et al. (2014) proposed two stochastic iterative greedy algorithms. Yet neither of the algorithms attain linear rate of convergence for the objective function value. Li et al. (2016) proposed a stochastic variance reduced gradient hard thresholding algorithm. Nevertheless, it cannot leverage the coordinate block to accelerate the convergence.

In this paper, by leveraging the advantages of both stochastic gradient descent (Nemirovski et al., 2009; Lan, 2012) and randomized block coordinate descent (Shalev-Shwartz

and Tewari, 2011; Nesterov, 2012; Beck and Tetruashvili, 2013; Richtárik and Takáč, 2014; Lu and Xiao, 2015), we propose a stochastic block coordinate gradient descent algorithm to solve the nonconvex sparsity constrained optimization problem in (1.1). The core of our algorithm is to exploit both stochastic partial gradient and full partial gradient restricted to each coordinate block. In detail, our algorithm consists of two layers of loops. For each iteration of the outer loop, the full gradient is computed once; while in the follow-up inner loop, partial stochastic gradient is computed to adjust the full gradient. We also incorporate mini-batch gradient computation into our algorithm, to further accelerate the convergence. Replacing full gradients with stochastic gradients restricted on coordinate blocks essentially trades the number of iterations with a low computational cost per iteration. We prove that the algorithm is guaranteed converge to the unknown true parameter $\boldsymbol{\beta}^*$ at a linear rate up to statistical error. The gradient complexity[1] of our algorithm is

$$\mathcal{O}\Big(\big(n + \kappa_{\widetilde{s}}|\mathcal{B}|/k\big)\log(1/\epsilon)\Big),$$

where k is the number of coordinate blocks, $|\mathcal{B}|$ is the mini batch size, ϵ is the optimization error for the objective function value, and $\kappa_{\widetilde{s}}$ is the condition number of the Hessian matrix $\nabla^2 F(\boldsymbol{\beta})$ restricted on any $\widetilde{s} \times \widetilde{s}$ principal submatrix. When $k = 1$ and $|\mathcal{B}| = 1$, our algorithm is reduced to accelerated gradient descent for the sparsity constrained nonconvex optimization problem. It improves the gradient complexity for gradient hard thresholding algorithms (Yuan et al., 2013; Jain et al., 2014) from $\mathcal{O}(n\kappa_{\widetilde{s}}\log(1/\epsilon))$ to $\mathcal{O}[(n + \kappa_{\widetilde{s}})\log(1/\epsilon)]$. Furthermore, for both sparse linear regression and sparse generalized linear model estimation, we show that the estimator from our algorithm attains the minimax optimal statistical rate. Experiments on both synthetic and real datasets backup our theory.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. In Section 3, we review two examples of the optimization problems. We present the algorithm in Section 4, and analyze it in Section 5. In addition, we apply our theory to two specific examples and illustrate the corresponding theory for the two examples. We compare the proposed algorithm with existing algorithms in Section 6. Finally, we conclude this paper in Section 7.

Notation Let $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{d\times d}$ be a matrix and $\mathbf{x} = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$ be vector. For $0 < q < \infty$, we define the ℓ_0, ℓ_q and ℓ_∞ vector norms as $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbb{1}(x_i \neq 0)$, $\|\mathbf{x}\|_q = \Big(\sum_{i=1}^d |x_i|^q\Big)^{\frac{1}{q}}$ and $\|\mathbf{x}\|_\infty = \max_{1\leq i\leq d}|x_i|$, where $\mathbb{1}(\cdot)$ represents the indicator function. For a vector $\mathbf{x}$, we define $\mathrm{supp}(\mathbf{x})$ as the index set of nonzero entries of $\mathbf{x}$, and $\mathrm{supp}(\mathbf{x}, s)$ as the index set of the top s entries of $\mathbf{x}$ in terms of magnitude. In addition, we denote by $\mathbf{x}_S$ the restriction of $\mathbf{x}$ onto a index set S, such that $[\mathbf{x}_S]_i = x_i$ if $i \in S$, and $[\mathbf{x}_S]_i = 0$ if $i \notin S$. In addition, we denote by $\mathbf{x}_s$ the restriction of $\mathbf{x}$ onto the top s entries in terms of magnitude, i.e., $[\mathbf{x}_s]_i = x_i$ if $i \in \mathrm{supp}(\mathbf{x}, s)$, and $[\mathbf{x}_s]_i = 0$ if $i \notin \mathrm{supp}(\mathbf{x}, s)$. For a set $\mathcal{B}$, we denote its cardinality by $|\mathcal{B}|$. For a matrix $\mathbf{X}$, its i-th row is denoted by $\mathbf{X}_{i*}$ and its j-th column is denoted by $\mathbf{X}_{*j}$.

[1]Gradient complexity is defined to be the iteration complexity times the number of gradient evaluation on each component function

2 RELATED WORK

In this section, we briefly review additional lines of research beyond the sparsity constrained nonconvex optimization, that are relevant to our work.

Gradient descent is computationally expensive at each iteration, hence stochastic gradient descent is often used when the data set is large. At each iteration, only one or a mini-batch of the n component functions f_i is sampled (Nemirovski et al., 2009; Lan, 2012). Due to the variance in estimating the gradient by stochastic sampling, stochastic gradient descent has a sublinear rate of convergence even when $F(\boldsymbol{\beta})$ is strongly convex and smooth. To accelerate stochastic gradient descent, various types of accelerated stochastic gradient descent algorithms (Schmidt et al., 2013; Johnson and Zhang, 2013; Konečnỳ and Richtárik, 2013; Defazio et al., 2014b; Mairal, 2014; Defazio et al., 2014a). The most relevant work to ours is stochastic variance reduced gradient (SVRG) (Johnson and Zhang, 2013) and its variants (Xiao and Zhang, 2014; Konečnỳ et al., 2014a).

In contrast to gradient descent, block coordinate descent (BCD) (Shalev-Shwartz and Tewari, 2011; Nesterov, 2012; Beck and Tetruashvili, 2013; Richtárik and Takáč, 2014; Lu and Xiao, 2015) only computes the full gradient of $F(\boldsymbol{\beta})$ restricted on a randomly selected coordinate block at each iteration. Compared with gradient descent, the per-iteration time complexity of RBCD is much lower. However, such algorithms still compute the partial full gradient based on all the n component functions per iteration.

Stochastic block coordinate gradient descent was proposed recently (Dang and Lan, 2015; Xu and Yin, 2015; Reddi et al., 2014), which integrates the advantages of stochastic gradient descent and block coordinate descent. Such algorithms compute the stochastic partial gradient restricted to one coordinate block with respect to one component function, rather than the full partial derivative with respect to all the component functions. These algorithms essentially employ sampling of both coordinates and data instances at each iteration. However, they can only achieve a sublinear rate of convergence. Recently, randomized block coordinate descent using mini-batches (Zhao et al., 2014; Wang and Banerjee, 2014; Konečnỳ et al., 2014b) are proposed

independently to accelerate the convergence of stochastic block coordinate gradient descent.

Our work departs from the above studies by considering a sparsity constrained nonconvex optimization problem instead of convex optimization. Due to the nonconvex nature of (1.1), our algorithm is no longer guaranteed to converge to the global optimum. Nevertheless, by taking into account the underlying statistical models, we illustrate that proposed algorithm is guaranteed to converge to the unknown true model parameters up to the statistical error.

3 ILLUSTRATIVE EXAMPLES OF SPARSITY CONSTRAINED OPTIMIZATION

In this section, we give two examples of the statistical estimation problems, which fall in the sparsity constrained optimization problem in (1.1). We return to demonstrate the implication of our general algorithm and theory to these examples in Section 5.

Example 3.1 (Sparse Linear Regression)**.** Consider the following linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ denotes a vector of the responses, and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the design matrix, $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is the unknown regression coefficient vector such that $\|\boldsymbol{\beta}^*\|_0 \leq s^*$, and $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is a noise vector. A commonly used estimator for the above sparse linear regression problem is the Lasso estimator (Tibshirani, 1996) with ℓ_1 norm penalty. An alternative estimator is the sparsity constrained estimator

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2n}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq s, \tag{3.2}$$

where s is a tuning parameter, which controls the sparsity of $\boldsymbol{\beta}$. This is indeed an example of the nonconvex optimization problem in (1.1) where $F(\boldsymbol{\beta}) = 1/(2n)\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$, $f_i(\boldsymbol{\beta}) = 1/2(\mathbf{x}_i^\top\boldsymbol{\beta} - y_i)^2$ and $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th row of $\mathbf{X}$. Similar estimator has been studied by Tropp and Gilbert (2007); Zhang (2011); Jain et al. (2014), to mention a few.

Example 3.2 (Sparse Generalized Linear Models)**.** We assume that the observations in each task are generated from generalized linear models

$$\mathbb{P}(y|\mathbf{x}, \boldsymbol{\beta}^*, \sigma) = \exp\left\{\frac{y\langle\boldsymbol{\beta}^*, \mathbf{x}\rangle - \Phi(\boldsymbol{\beta}^{*\top}\mathbf{x})}{c(\sigma)}\right\}, \tag{3.3}$$

where $\Phi(\cdot) : \mathbb{R} \to \mathbb{R}$ is a link function, $y \in \mathbb{R}$ is the response variable, $\mathbf{x} \in \mathbb{R}^d$ is the predictor vector, $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is the parameter such that $\|\boldsymbol{\beta}^*\|_0 \leq s^*$, and $c(\sigma) \in \mathbb{R}$ is fixed and known scale parameter. A special example of generalized linear model is the linear regression model where the noise follows from a Gaussian distribution, which corresponds to $c(\sigma) = \sigma^2$ and $\Phi(t) = t^2$. Logistic regression is another special case of the generalized linear model, where $\Phi(t) = \log(1 + \exp(t)), c(\sigma) = 1$ and $y \in \{0, 1\}$.

Given $\{\mathbf{x}_i, y_i\}_{i=1}^n$, a widely used estimator for $\boldsymbol{\beta}^*$ is the ℓ_1 regularized maximum likelihood estimator (Negahban et al., 2009; Loh and Wainwright, 2013). An alternative estimator is the sparsity constrained maximum likelihood estimator as follows

$$\min_{\boldsymbol{\beta}} -\frac{1}{n}\sum_{i=1}^n \frac{1}{c(\sigma)}\big[y_i\langle\boldsymbol{\beta}, \mathbf{x}_i\rangle - \Phi(\boldsymbol{\beta}^{*\top}\mathbf{x})\big], \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq s. \tag{3.4}$$

The estimator in (3.4) has been investigated by Jalali et al. (2011); Yuan et al. (2013); Li et al. (2016).

For more examples, please refer to Yuan et al. (2013); Jain et al. (2014) and references therein.

4 THE PROPOSED ALGORITHM

In this section, we present an accelerated stochastic block coordinate descent algorithm based on nonconvex optimization for solving the proposed estimator in (1.1). The key motivation of the algorithm is using iterative hard thresholding to ensure cardinality constraint and mixed mini-batch partial gradient to reduce the variance of the stochastic gradient and accelerate the convergence. We display the algorithm in Algorithm 1.

Algorithm 1 Accelerated Stochastic Block Coordinate Gradient Descent with Hard Thresholding (ASBCDHT)

Initialization: $\widetilde{\boldsymbol{\beta}}^{(0)}$ **with** $\|\widetilde{\boldsymbol{\beta}}^{(0)}\|_0 \leq s$
for $\ell = 1, 2, \ldots$ **do**
  $\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(\ell-1)}$
  $\widetilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widetilde{\boldsymbol{\beta}})$
  $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$
  Randomly sample z **uniformly from** $\{0, \ldots, m-1\}$
  for $t = 0, 1, \ldots, z-1$ **do**
    Randomly sample a mini-batch $\mathcal{B}$ **from** $\{1, \ldots, n\}$ **uniformly**
    Randomly sample j **from** $\{1, \ldots, k\}$ **uniformly**
    $[\mathbf{v}]_{\mathcal{G}_j} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \nabla_{\mathcal{G}_j} f_i(\boldsymbol{\beta}^{(t)}) - \nabla_{\mathcal{G}_j} f_i(\widetilde{\boldsymbol{\beta}}) + \widetilde{\boldsymbol{\mu}}_{\mathcal{G}_j}$
    $\boldsymbol{\beta}^{(t+0.5)} = \boldsymbol{\beta}^{(t)} - \eta[\mathbf{v}]_{\mathcal{G}_j}$
    $\boldsymbol{\beta}^{(t+1)} = \mathrm{HT}\big(\boldsymbol{\beta}^{(t+0.5)}, s\big)$
  end for
  Set $\widetilde{\boldsymbol{\beta}}^{(\ell)} = \boldsymbol{\beta}^{(z)}$
end for

Note that in Algorithm 1, we have two layers of loops. In the outer loop, $\widetilde{\boldsymbol{\beta}}^{(r-1)}$ denotes the estimated parameter from previous stage, and $\widetilde{\boldsymbol{\mu}}$ denotes the full gradient computed based on $\widetilde{\boldsymbol{\beta}}^{(r-1)}$.

In the inner loop, Algorithm 1 integrates the advantages of randomized block coordinate descent and stochastic gradient descent together. Let $\{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$ be a partition of all the d coordinates where $\mathcal{G}_j$ is a block of coordinates. In step 7, it uniformly samples a mini batch of component functions. And in step 8, it uniformly samples a coordinate block. The random sampling significantly reduces the computational cost. Based on the mini batch of component functions, and the coordinate block, it calculates the mixed partial gradient $[\mathbf{v}]_{\mathcal{G}_j}$ restricted on the selected coordinate block, which is the combination of the partial stochastic gradient and the partial full gradient (See step 9). Note that similar mixed gradient has been originally introduced in Johnson and Zhang (2013) and later adopted by Xiao and Zhang (2014); Konečnỳ et al. (2014a); Zhao et al. (2014); Konečnỳ et al. (2014b) to reduce the variance introduced by random sampling. More specifically, we can show that the variance of $[\mathbf{v}]_{\mathcal{G}_j}$ i.e., $\mathbb{E}[\|\mathbf{v}_{\widetilde{\mathcal{S}}} - \nabla_{\widetilde{\mathcal{S}}} F(\boldsymbol{\beta}^{(t)})\|_2^2]$ diminishes when $\boldsymbol{\beta}^{(t)}$ approaches the unknown true model parameter vector $\boldsymbol{\beta}^*$. $\boldsymbol{\beta}^{(t+0.5)}$ is the output of coordinate gradient descent step. Since $\boldsymbol{\beta}^{(t+0.5)}$ is not necessarily sparse after the coordinate descent update, in order to make it sparse, we apply a hard thresholding procedure (Yuan et al., 2013; Jain et al., 2014) right after coordinate descent step. The hard thresholding operator is defined as follows:

$$[\mathrm{HT}(\boldsymbol{\beta}, s)]_i = \begin{cases} \beta_i, & \text{if } i \in \mathrm{supp}(\boldsymbol{\beta}, s), \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

The hard thresholding step preserves the entries of $\boldsymbol{\beta}^{(t+0.5)}$ with the top s large magnitudes and sets the rest to zero. This gives rise to $\boldsymbol{\beta}^{(t+1)}$. Recall that s is a tuning parameter that controls the sparsity level.

5 MAIN THEORY AND IMPLICATIONS

In this section, we will present the main theory that characterizes the performance of Algorithm 1, followed which we show the consequences of our theory when it is applied to the two examples in Section 3.

5.1 MAIN THEORETICAL RESULTS

We first layout a set of definition and assumptions, that are essential for our main theory.

Definition 5.1 (Sparse Eigenvalues)**.** Let $\widetilde{s}$ be a positive integer. The largest and smallest s-sparse eigenvalues of the Hessian matrix $\nabla^2 F(\boldsymbol{\beta})$ are

$$\rho_+(\widetilde{s}) = \sup_{\mathbf{v}} \Big\{ \mathbf{v}^\top \nabla^2 F(\boldsymbol{\beta}) \mathbf{v} : \|\mathbf{v}\|_0 \leq \widetilde{s}, \|\mathbf{v}\|_2 = 1, \boldsymbol{\beta} \in \mathbb{R}^d \Big\},$$
$$\rho_-(\widetilde{s}) = \inf_{\mathbf{v}} \Big\{ \mathbf{v}^\top \nabla^2 F(\boldsymbol{\beta}) \mathbf{v} : \|\mathbf{v}\|_0 \leq \widetilde{s}, \|\mathbf{v}\|_2 = 1, \boldsymbol{\beta} \in \mathbb{R}^d \Big\}.$$

Moreover, we define the restricted condition number $\kappa_{\widetilde{s}} = \rho_+(\widetilde{s})/\rho_-(\widetilde{s})$.

Based on the sparse eigenvalues, we make the following assumptions on $f_i(\boldsymbol{\beta})$ and $F(\boldsymbol{\beta})$ with respect to $\rho_+(\widetilde{s})$ and $\rho_-(\widetilde{s})$ mentioned above.

Assumption 5.2 (Restricted Strong Smoothness)**.** $f_i(\boldsymbol{\beta})$ satisfies restricted strong smoothness condition at sparsity level $\widetilde{s}$ with a constant $\rho_+(\widetilde{s}) > 0$: for all $\boldsymbol{\beta}, \boldsymbol{\beta}'$ such that $\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_0 \leq \widetilde{s}$, we have

$$f_i(\boldsymbol{\beta}) \leq f_i(\boldsymbol{\beta}') + \nabla f_i(\boldsymbol{\beta}')^\top (\boldsymbol{\beta} - \boldsymbol{\beta}') + \frac{\rho_+(\widetilde{s})}{2} \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_2^2.$$

Assumption 5.3 (Restricted Strong Convexity)**.** $F(\boldsymbol{\beta})$ satisfies restricted strong convexity condition at sparsity level $\widetilde{s}$ with a constant $\rho_-(\widetilde{s}) > 0$: for all $\boldsymbol{\beta}, \boldsymbol{\beta}'$ such that $\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_0 \leq \widetilde{s}$, we have

$$F(\boldsymbol{\beta}) \geq F(\boldsymbol{\beta}') + \nabla F(\boldsymbol{\beta}')^\top (\boldsymbol{\beta} - \boldsymbol{\beta}') + \frac{\rho_-(\widetilde{s})}{2} \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_2^2.$$

Assumptions 5.2 and 5.3 indicate that function $f_i(\boldsymbol{\beta})$ is smooth and function $F(\boldsymbol{\beta})$ is strongly convex when restricted on to a sparse subspace. These restricted strong smoothness and strong convexity conditions ensure that the standard convex optimization results for strongly convex and smooth objective functions (Nesterov, 2004) can be applied to our problem settings as well. It is worth noting that we do not require each $f_i(\boldsymbol{\beta})$ to be restricted strongly convex, we only require their summation $F(\boldsymbol{\beta})$ is restricted strongly convex. $f_i(\boldsymbol{\beta})$ typically does not satisfy restricted strong convexity for $\widetilde{s} > 1$. Recall that, in Example 3.1, $f_i(\boldsymbol{\beta}) = 1/2(\mathbf{x}_i^\top \boldsymbol{\beta} - y_i)^2$, which is obviously not restricted strongly convex unless $\widetilde{s} = 1$.

Now we are ready to present our main theorem.

Theorem 5.4. Suppose Assumptions 5.2 and 5.3 hold with $\widetilde{s} = 2s + s^*$. In addition, assume that $0 < \eta \leq 1/(18\rho_+(\widetilde{s}))$ and m, s are chosen such that,

$$\alpha = \frac{2k\tau^{m-1}(\tau - 1)}{\rho_-(\widetilde{s})\eta\gamma(\tau^m - 1)} + \frac{12\eta\rho_+(\widetilde{s})(n - |\mathcal{B}|)}{|\mathcal{B}|(n-1)\gamma} < 1,$$

where $\gamma = 1 - 12\eta\rho_+(\widetilde{s})(n - |\mathcal{B}|)/\big[|\mathcal{B}|(n-1)\big] - 6\eta\rho_+(\widetilde{s})$ and $\tau = 1 + 2\sqrt{s^*}/\sqrt{s - s^*}$. Then the estimator $\widetilde{\boldsymbol{\beta}}^{(\ell)}$ from Algorithm 1 satisfies

$$\mathbb{E}\big[F(\widetilde{\boldsymbol{\beta}}^{(\ell)}) - F(\boldsymbol{\beta}^*)\big] \leq \alpha^\ell \mathbb{E}\big[F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)\big] + \frac{3\eta}{2\gamma(1-\alpha)} \|\nabla_{\widetilde{s}} F(\boldsymbol{\beta}^*)\|_2^2. \tag{5.1}$$

We have the following remarks regarding the above theorem results:

Remark 5.5. Theorem 5.4 implies that in order to achieve linear rate of convergence, the learning rate η need to be set sufficiently small, the sparsity constraint s and the number of inner loop iterations m should be set sufficiently large such that $\alpha \leq 1$. Here we provide an example showing

Table 1: A comparison of gradient complexity for different algorithms.

Algorithms	Gradient Complexity
Nguyen et al. (2014)	$\mathcal{O}\big(1/\epsilon\big)$
Yuan et al. (2013)	$\mathcal{O}\big(n\kappa_{\widetilde{s}} \cdot \log(1/\epsilon)\big)$
Li et al. (2016)	$\mathcal{O}\big([n + \kappa_{\widetilde{s}}] \cdot \log(1/\epsilon)\big)$
Ours	$\mathcal{O}\big((n + \kappa_{\widetilde{s}}\lvert\mathcal{B}\rvert/k) \cdot \log(1/\epsilon)\big)$

this is absolutely achievable. Without loss of generality, we consider the simplified scenario where the batch size $|\mathcal{B}| = 1$ and coordinate block number $k = 1$. As stated in the theorem condition, suppose we choose $\eta = 1/(36\rho_+(\widetilde{s}))$, then we have $\gamma = 1/2$. This simplifies the expression of α to:

$$\alpha = 144\kappa_{\widetilde{s}} \cdot \frac{\tau^m}{\tau^m - 1}\Big(1 - \frac{1}{\tau}\Big) + \frac{2}{3},$$

Therefore, provided that s is chosen to be

$$s \geq \big(1 + 4(1728\kappa_{\widetilde{s}} - 1)^2\big)s^*,\ m \geq \log 2 \cdot (1728\kappa_{\widetilde{s}} - 1),$$

we have,

$$\frac{\tau^m}{\tau^m - 1} \leq 2,\ \Big(1 - \frac{1}{\tau}\Big) \leq \frac{1}{1728\kappa_{\widetilde{s}}},$$

which immediately verifies that $\alpha \leq \frac{5}{6} < 1$.

Remark 5.6. Theorem 5.4 illustrates the linear rate of convergence in objective function value gap. From (5.1), in order to ensure that the linear converging term satisfies $\alpha^\ell\big[F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)\big] \leq \epsilon$, the number of stages should satisfy

$$\ell \geq \log_{\alpha^{-1}} \frac{F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)}{\epsilon}.$$

Thus we need $\mathcal{O}(\log(1/\epsilon))$ outer iterations in Algorithm 1. Recall that from Remark 5.5, we have $m = \Omega(\kappa_{\widetilde{s}})$. Since in each outer iteration, we need to compute one full gradient and m mixed mini-batch gradient, the overall gradient complexity is $\mathcal{O}\big((n + \kappa_{\widetilde{s}} \cdot |\mathcal{B}|/k) \cdot \log(1/\epsilon)\big)$, where k is the number of coordinate blocks, $|\mathcal{B}|$ is the batch size, and $\kappa_{\widetilde{s}}$ is the restricted condition number of $\nabla^2 F(\boldsymbol{\beta}^*)$. For the ease of comparison, we summarize the gradient complexity of our algorithm as well as the other state of the art algorithms in Table 1. As we can see, our proposed algorithm improves gradient complexity over previous work. In particular, with $k > 1$ and $|\mathcal{B}| = 1$, the gradient complexity of our algorithm outperforms that of Li et al. (2016). In the special case that $k = 1$ and $|\mathcal{B}| = 1$, our algorithm has the same gradient complexity as Li et al. (2016). In general, our algorithm provides more flexibility than Li et al. (2016) by incorporating mini-batch technique.

Theorem 5.4 immediately implies the following results.

Corollary 5.7. Under the same conditions of Theorem 5.4, the estimator $\widetilde{\boldsymbol{\beta}}^{(\ell)}$ from Algorithm 1 satisfies

$$\mathbb{E}\|\widetilde{\boldsymbol{\beta}}^{(\ell)} - \boldsymbol{\beta}^*\|_2 \leq \underbrace{\alpha^{\ell/2}\sqrt{\frac{2\big[F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)\big]}{\rho_-(\widetilde{s})}}}_{\text{Optimization Error}} + \underbrace{\left(\frac{2}{\rho_-(\widetilde{s})} + \sqrt{\frac{3\eta}{\gamma\rho_-(\widetilde{s})(1-\alpha)}}\right)\sqrt{\widetilde{s}}\|\nabla F(\boldsymbol{\beta}^*)\|_\infty}_{\text{Statistical Error}}. \tag{5.2}$$

We have the following remark regarding the above result.

Remark 5.8. The right hand side of (5.2) consists of two terms. The first term is the optimization error, which goes to zero as ℓ increase, since $\alpha \in (0, 1)$. The second term corresponds to the statistical error, and is proportional to $\sqrt{\widetilde{s}}\|\nabla F(\boldsymbol{\beta}^*)\|_\infty$. Since $\widetilde{s} = s + s^*$ and $s = O(s^*)$, the statistical is actually in the order of $\sqrt{s^*}\|\nabla F(\boldsymbol{\beta}^*)\|_\infty$. Note that the statistical error of the regularized M estimators (Negahban et al., 2009; Loh and Wainwright, 2013) for sparse linear regression and generalized linear models, is also proportional to $\sqrt{s^*}\|\nabla F(\boldsymbol{\beta}^*)\|_\infty$. Theorem 5.7 suggests that our algorithm attains a linear rate of convergence to the true parameter, up to the statistical error. In other words, our algorithm linearly converges to a local optima, which enjoys good statistical property.

5.2 IMPLICATION FOR SPECIFIC STATISTICAL ESTIMATION PROBLEMS

We now turn to the consequences of our algorithm and general theory for specific statistical estimation problems that arise in applications. In particular, we show the theoretical results by applying our theory to the two examples introduced in Section 3.

We begin with a corollary for the problem of sparse linear regression, as introduced in Example 3.1. We assume that the noise vector $\boldsymbol{\epsilon}$ in (3.1) is zero-mean and has sub-Gaussian tails.

Assumption 5.9. $\boldsymbol{\epsilon}$ is a zero mean random vector, and there exists a constant $\sigma > 0$ such that for any fixed $\|\mathbf{v}\|_2 = 1$, we have

$$\mathbb{P}\big(|\mathbf{v}^\top\boldsymbol{\epsilon}| > \delta\big) \leq 2\exp\left(-\frac{\delta^2}{2\sigma^2}\right) \quad \text{for all} \quad \delta > 0.$$

In addition, without loss of generality, we make an additional assumption on the design matrices $\mathbf{X}$ in (3.1).

Assumption 5.10. For all columns in $\mathbf{X} \in \mathbb{R}^{n\times d}$, we have $\|\mathbf{X}_{*j}\|_2 \leq \sqrt{n}$, where $\mathbf{X}_{*j}$ is the j-th column of $\mathbf{X}$.

Note that Assumption 5.10 is often made in the analysis of Lasso estimator (Negahban et al., 2009; Zhang et al., 2009).

Corollary 5.11. Under the same conditions as Corollary 5.7, if Assumptions 5.9 and 5.10 hold, then with probability at least $1 - 1/d$, the estimator $\widetilde{\boldsymbol{\beta}}^{(\ell)}$ from Algorithm 1 for sparse linear regression in (3.2) satisfies

$$\mathbb{E}\|\widetilde{\boldsymbol{\beta}}^{(\ell)} - \boldsymbol{\beta}^*\|_2 \leq \underbrace{\alpha^{\ell/2}\sqrt{\frac{2\big[F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)\big]}{\rho_-(\widetilde{s})}}}_{\text{Optimization Error}} + \underbrace{C\left(\frac{2}{\rho_-(\widetilde{s})} + \sqrt{\frac{3\eta}{\gamma\rho_-(\widetilde{s})(1-\alpha)}}\right)\sigma\sqrt{\frac{s^*\log d}{n}}}_{\text{Statistical Error}}, \quad (5.3)$$

where σ is the variance proxy of the sub-Gaussian random vector $\boldsymbol{\epsilon}$.

Corollary 5.11 suggests that when applying our algorithm to sparse linear regression, it achieves $O(\sqrt{s^*\log d/n})$ statistical error. It matches the minimax optimal rate for sparse linear regression (Raskutti et al., 2011).

We then provide a corollary for the problem of sparse generalized linear model estimation, as introduced in Example 3.2. For generalized linear model, we need the following assumption on its link function $\Phi(t)$, which is introduced in (3.3).

Assumption 5.12. There exists one $\alpha_u > 0$ such that the second derivative of the link function satisfies $\Phi''(t) \leq \alpha_u$ for all $t \in \mathbb{R}$.

Similar assumption has been made in Loh and Wainwright (2013).

Corollary 5.13. Under the same conditions as Corollary 5.7, if Assumptions 5.10 and 5.12 hold, then with probability at least $1-1/d$, the estimator $\widetilde{\boldsymbol{\beta}}^{(\ell)}$ from Algorithm 1 for sparse generalized linear models in (3.4) satisfies

$$\mathbb{E}\|\widetilde{\boldsymbol{\beta}}^{(\ell)} - \boldsymbol{\beta}^*\|_2 \leq \underbrace{\alpha^{\ell/2}\sqrt{\frac{2\big[F(\widetilde{\boldsymbol{\beta}}^{(0)}) - F(\boldsymbol{\beta}^*)\big]}{\rho_-(\widetilde{s})}}}_{\text{Optimization Error}} + \underbrace{C\left(\frac{2}{\rho_-(\widetilde{s})} + \sqrt{\frac{3\eta}{\gamma\rho_-(\widetilde{s})(1-\alpha)}}\right)\alpha_u\sqrt{\frac{s^*\log d}{n}}}_{\text{Statistical Error}}, \quad (5.4)$$

where α_u is an upper bound on the second derivative of the link function $\Phi(t)$.

Corollary 5.13 demonstrates that when applying our algorithm to sparse generalized linear models, it achieves $O(\sqrt{s^*\log d/n})$ statistical error rate. It is also minimax rate-optimal.

6 EXPERIMENTS

In this section, we apply Algorithm 1 to the two examples discussed in Section 3, and present numerical results on both synthetic and large-scale real datasets to verify the performance of the proposed algorithm, and compare it with state-of-the-art sparsity cardinality constraint methods.

6.1 BASELINE METHODS

We compare our algorithm with several state-of-the-art baseline methods: (1) gradient descent with hard thresholding (GraHTP) by Yuan et al. (2013) ; (2) stochastic variance reduced gradient with hard thresholding by Li et al. (2016) (SVRGHT); (3) Our proposed accelerated stochastic block coordinate gradient descent with hard thresholding (ASBCDHT) with batch size $|\mathcal{B}| = 1$; and (4) Our proposed accelerated stochastic block coordinate descent with hard thresholding (ASBCDHT) with batch size $|\mathcal{B}| = 10$. Since our algorithm involves coordinate block, we set the block number as $k = 10$, where each block has (almost) the same number of coordinates. In addition, SVRGHT and ASBCDHT are based on mixed (partial) gradients, hence we need to specify the number of iterations for the inner loop. We simply choose $m = n$ since our theory demonstrates that m should be chosen as $\Omega(\kappa_{\widetilde{s}})$.

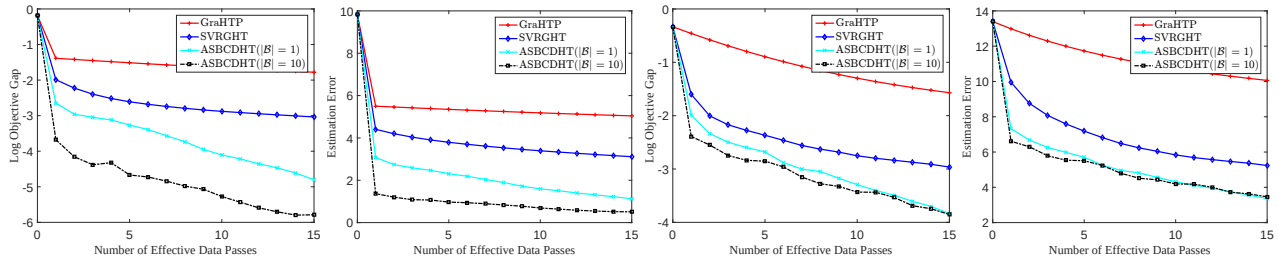
In order to fairly compare the above algorithms, we notice that at each iteration of GradHTP and SVRGHT, the gradient is updated with respect to all coordinates. When in our algorithm, at each iteration of Algorithm 1 the gradient is updated with respect to only a sampled coordinate block among all coordinates, so the computational cost is lower than that of gradient descent per iteration. Therefore, comparing algorithms that update the gradient with respect to different numbers of coordinates per iteration should be based on the same number of entire data passes (the least possible iterations for passing through the entire data set with respect to all coordinates).

6.2 SPARSE LINEAR REGRESSION

We first investigate the sparsity constrained linear regression problem in (3.2).

6.2.1 Synthetic Data

We generate an $n \times d$ design matrix $\mathbf{X}$ with rows drawn independently from a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$, where each element of $\boldsymbol{\Sigma}$ is defined by $\Sigma_{ij} = 0.6^{|i-j|}$. The true regression coefficient vector $\boldsymbol{\beta}^*$ has s^* nonzero entries that are drawn independently from the standard normal distribution. The response vector is generated by $\mathbf{y} = \mathbf{X}^\top\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$, where each entry of $\boldsymbol{\epsilon}$ follows a normal distribution with zero mean and variance $\sigma^2 = 0.01$. In this part, we test our proposed algorithm



(a) (b) (c) (d)

Figure 1: Comparison of different algorithms for sparsity constrained sparse linear regression on the two synthetic datasets: (1) $n = 1000, d = 2000, s^* = 100$ (shown in (a) and (b)); (2) $n = 5000, d = 10000, s^* = 500$ (shown in (c) and (d)). (a) and (c) show the logarithm of the function value gap for the two datasets. (b) and (d) demonstrate the estimation error for the two datasets.

Table 2: Regression on E2006-TFIDF: MSE comparison of algorithms for the same entire effective data passes over 10 replications. The boldfaced results denote the lowest MSE among all the algorithms for the same entire effective data passes.

Method	#Data Passes=3	#Data Passes=6	#Data Passes=9	#Data Passes=12	#Data Passes=15
GraHTP	1.3388	1.1204	1.0522	1.0190	0.9970
SVRGHT	0.8809±0.0949	0.8150±0.0718	0.7819±0.0612	0.7574±0.0539	0.7385±0.0483
ASBCDHT($\|\mathcal{B}\|$ =1)	0.7039±0.1037	0.6835±0.0789	0.6709±0.0677	0.6607±0.0600	0.6518±0.0533
ASBCDHT($\|\mathcal{B}\|$ =10)	**0.7003**±0.1118	**0.6769**±0.0806	**0.6627**±0.0626	**0.6519**±0.0519	**0.6426**±0.0415

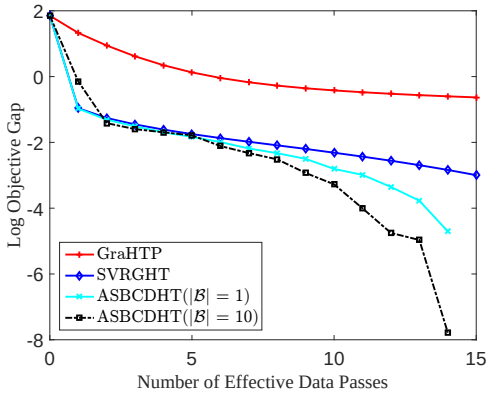


Figure 2: Comparison of different algorithms in terms of the logarithm of objective function value gap on E2006-TFIDF dataset.

along with with the state-of-the-art algorithms in two different settings: (1) $n = 1000, d = 2000, s^* = 100$; and (2) $n = 5000, d = 10000, s^* = 500$. Each experiment is repeated for 10 times. For each algorithm, we plot the logarithm of the objective function value gap and the estimation error $\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\|_2$ for comparison. The sparsity parameter s is set to $s = 1.2s^*$ for all the algorithms according to the theory. The step size η of different algorithms is tuned by cross validation.

In Figure 1, we compare the logarithm of the function value gap and the estimation error in the above two datasets for all algorithms. Figure 1 (a) and (c) demonstrate that the optimization error decreases to zero at a linear rate while 1 (b) and (d) show that the estimation error of the estimator converges to certain level after some number of effective data passes. This is consistent with our theory in Corollary 5.11 that the estimation error of our algorithm consists of two terms: the optimization error that goes to zero, and the statistical error that depends on the problem parameters (d, n, s^* and so on). From Figure 1, it is obvious that our proposed algorithm outperforms other state-of-the-art algorithms in estimation error after the same number of effective data passes. Also note that when the data size is relatively small, our algorithm with batch size equals 10 performs better than the case when batch size equals 1, while this advantage decays as the data size grows. This is probably because the mini-batch sampling is more advantageous when the data are relatively small.

6.2.2 E2006-TFIDF Data

We use E2006-TFIDF dataset to test the sparsity constrained linear regression, which predicts risk from financial reports from thousands of publicly traded U.S. companies (Kogan et al., 2009). It contains $16,087$ training instances, $3,308$ testing instances and we randomly sample $50,000$ features for this experiment. In this section, we choose $s = 2000$ and compare all algorithms using mean square error (MSE) for 15 entire effective data passes over 10 replications. The step size η is chosen by cross validation on the training data.

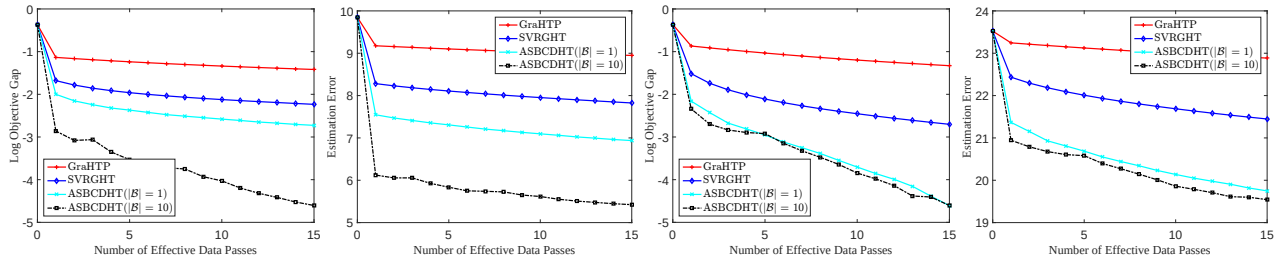


Figure 3: Comparison of different algorithms for sparsity constrained sparse logistic regression on the two synthetic datasets: (1) $n = 1000, d = 2000, s^* = 100$ (shown in (a) and (b)); (2) $n = 5000, d = 10000, s^* = 500$ (shown in (c) and (d)). (a) and (c) show the logarithm of the function value gap for the two datasets. (b) and (d) are the estimation error for the two datasets.

Table 3: Classification on RCV1: classification error comparison of algorithms for the same entire effective data passes over 10 replications. The boldfaced results denote the lowest classification error among all the algorithms for the same entire effective data passes.

Method	#Data Passes=3	#Data Passes=6	#Data Passes=9	#Data Passes=12	#Data Passes=15
GraHTP	0.0758	0.0748	0.0739	0.0733	0.0727
SVRGHT	0.0848±0.0043	0.0763±0.0034	0.0708±0.0029	0.0677±0.0025	0.0671±0.0020
ASBCDHT($\lvert\mathcal{B}\rvert$ =1)	0.0662±0.0044	0.0648±0.0025	0.0644±0.0019	0.0642±0.0021	0.0639±0.0021
ASBCDHT($\lvert\mathcal{B}\rvert$ =10)	**0.0550**±0.0043	**0.0542**±0.0034	**0.0539**±0.0029	**0.0534**±0.0025	**0.0527**±0.0020

Figure 2 illustrates the logarithm of the objective function value gap for all the baseline algorithms and ours. We can see that our algorithm converges faster than the other baselines and our algorithm converges to much smaller objective function value than the other algorithms. In addition, Table 2 shows the mean value as well as the standard error of MSE for all the algorithms with respect to the number of effective data passes. Since there is no randomness in GraHTP, its standard error is zero. We can see that our algorithm attains much smaller mean square error than the other baseline algorithms for the same entire effective data passes. In particular, when the number of data passes equals $3, 6, 9$ and 12, our ASBCDHT with batch size $\mathcal{B} = 10$ achieves the lowest MSE; and when the number of data passes equals 15, our ASBCDHT with batch size $\mathcal{B} = 1$ achieves the best performance.

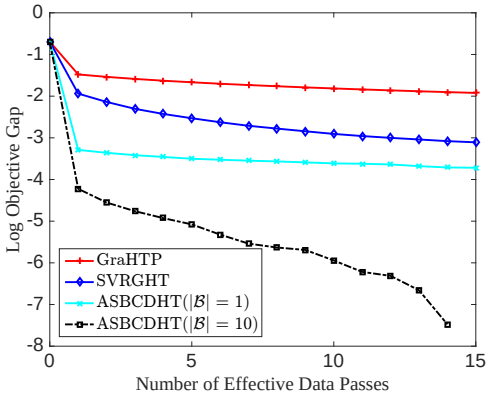


Figure 4: Comparison of different algorithms in terms of the logarithm of the objective function value gap on RCV1 dataset.

6.3 SPARSE LOGISTIC REGRESSION

We then evaluate the sparsity constrained generalized linear model, by considering a particular instance of sparse generalized linear model, i.e., sparse logistic regression. Its estimator is given by

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\left[-y_i\cdot\mathbf{x}_i^\top\boldsymbol{\beta}+\log\left(1+\exp(\mathbf{x}_i^\top\boldsymbol{\beta})\right)\right]$$
$$\text{subject to} \quad \|\boldsymbol{\beta}\|_0 \le s,$$

where $y_i \in \{0, 1\}$. Similar estimator has been studied by Yuan et al. (2013).

6.3.1 Synthetic Data

We generate an $n \times d$ design matrix $\mathbf{X}$ with rows drawn independently from a multivariate normal distribution $N(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ is a $d \times d$ identity matrix. The true regression coefficient vector $\boldsymbol{\beta}^*$ has s^* nonzero entries that are drawn independently from the standard normal distribution. Each response variable is generated from the logistic distribution

$$y_i = \begin{cases} 1, & \text{with probability } 1/(1+\exp(\mathbf{x}_i^\top\boldsymbol{\beta}^*)), \\ 0, & \text{with probability } 1-1/(1+\exp(\mathbf{x}_i^\top\boldsymbol{\beta}^*)). \end{cases}$$

In this part, we test our proposed algorithm along with the baseline algorithms in two different datasets: (1) $n = 1000, d = 2000, s^* = 100$; and (2) $n = 5000, d = 10000, s^* = 500$. Each experiment is repeated for 10 times and for all algorithms we plot the logarithm of the objective function value gap and the estimation error $\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\|_2$ for comparison. The sparsity parameter s is again set to $s = 1.2s^*$. And the step size η is chosen by cross validation.

Figure 3 illustrates the logarithm of the function value gap and the estimation error $\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\|_2$. The four sub-figures in Figure 3 demonstrate the similar trends as in Figure 1. Our proposed algorithm outperforms the other baseline algorithms by a large margin.

6.3.2 RCV1 Data

In order to evaluate the sparsity constrained logistic regression, we use RCV1 dataset, which is a Reuters Corpus Volume I data set for text categorization research (Lewis et al., 2004). Reuters Corpus Volume I (RCV1) is an archive of over 800,000 manually categorized newswire stories made available by Reuters, Ltd. for research purposes. This dataset contains $20,242$ training instances, $677,399$ testing instances and $47,236$ features. We use the whole training set and a subset of the test set, which contains $20,000$ testing instances for our experiment. In detail, we choose $s = 500$ and compare all algorithms in terms of their classification error on the test set for 15 entire effective data passes over 10 replications. The step size η is chosen by cross validation on the training set.

Table 3 demonstrates the classification results for the four algorithms including ours. It is obvious that our proposed algorithm achieves the lowest test error on RCV1 dataset on all periods of effective data passes and beats the other state-of-the-art baseline algorithms. Figure 4 further illustrates the logarithm of the objective function value gap for both the baseline algorithms and ours. This clearly demonstrates the superiority of our algorithm.

7 CONCLUSIONS

We proposed an accelerated stochastic block coordinate descent algorithm for sparsity constrained nonconvex optimization problems. We show that the algorithm enjoys a linear rate of convergence to the unknown true parameter up to the statistical error. Experiments on both synthetic and real datasets verify the effectiveness of our algorithm.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Research was sponsored by Quanquan Gu's startup funding at Department of Systems and Information Engineering, University of Virginia.

References

BAHMANI, S., RAJ, B. and BOUFOUNOS, P. T. (2013). Greedy sparsity-constrained optimization. *The Journal of Machine Learning Research* **14** 807–841.

BECK, A. and TETRUASHVILI, L. (2013). On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization* **23** 2037–2060.

BLUMENSATH, T. and DAVIES, M. E. (2009). Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis* **27** 265–274.

BÜHLMANN, P. and VAN DE GEER, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

DANG, C. D. and LAN, G. (2015). Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization* **25** 856–881.

DEFAZIO, A., BACH, F. and LACOSTE-JULIEN, S. (2014a). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*.

DEFAZIO, A. J., CAETANO, T. S. and DOMKE, J. (2014b). Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conference on Machine Learning*.

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

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*.

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*.

JOHNSON, R. and ZHANG, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*.

KOGAN, S., LEVIN, D., ROUTLEDGE, B. R., SAGI, J. S. and SMITH, N. A. (2009). Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

KONEČNÝ, J., LIU, J., RICHTÁRIK, P. and TAKÁČ, M. (2014a). ms2gd: Mini-batch semi-stochastic gradient descent in the proximal setting. *arXiv:1410.4744* .

KONEČNÝ, J., QU, Z. and RICHTÁRIK, P. (2014b). Semi-stochastic coordinate descent. *arXiv:1412.6293* .

KONEČNÝ, J. and RICHTÁRIK, P. (2013). Semi-stochastic gradient descent methods. *arXiv:1312.1666* .

LAN, G. (2012). An optimal method for stochastic composite optimization. *Mathematical Programming* **133** 365–397.

LEWIS, D. D., YANG, Y., ROSE, T. G. and LI, F. (2004). Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research* **5** 361–397.

LI, X., ZHAO, T., ARORA, R., LIU, H. and HAUPT, J. (2016). Stochastic variance reduced optimization for nonconvex sparse learning. Tech. rep.

LIU, J., FUJIMAKI, R. and YE, J. (2013). Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. *arXiv preprint arXiv:1401.0086* .

LOH, P.-L. and WAINWRIGHT, M. J. (2013). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*.

LU, Z. and XIAO, L. (2015). On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming* **152** 615–642.

MAIRAL, J. (2014). Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv:1402.4419* .

MALLAT, S. G. and ZHANG, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on* **41** 3397–3415.

NEEDELL, D. and TROPP, J. A. (2009). Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis* **26** 301–321.

NEGAHBAN, S., YU, B., WAINWRIGHT, M. J. and RAVIKUMAR, P. K. (2009). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*.

NEMIROVSKI, A., JUDITSKY, A., LAN, G. and SHAPIRO, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* **19** 1574–1609.

NESTEROV, Y. (2004). *Introductory lectures on convex optimization: A Basic Course*. Springer Science & Business Media.

NESTEROV, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* **22** 341–362.

NGUYEN, N., NEEDELL, D. and WOOLF, T. (2014). Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *arXiv preprint arXiv:1407.0088* .

RASKUTTI, G., WAINWRIGHT, M. J. and YU, B. (2011). Minimax rates of estimation for high-dimensional linear regression over-balls. *Information Theory, IEEE Transactions on* **57** 6976–6994.

REDDI, S., HEFNY, A., DOWNEY, C., DUBEY, A. and SRA, S. (2014). Large-scale randomized-coordinate descent methods with non-separable linear constraints. *arXiv preprint arXiv:1409.2617* .

RICHTÁRIK, P. and TAKÁČ, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming* **144** 1–38.

SCHMIDT, M., ROUX, N. L. and BACH, F. (2013). Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388* .

SHALEV-SHWARTZ, S., SREBRO, N. and ZHANG, T. (2010). Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization* **20** 2807–2832.

SHALEV-SHWARTZ, S. and TEWARI, A. (2011). Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research* **12** 1865–1892.

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

TROPP, J. A. and GILBERT, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on* **53** 4655–4666.

WANG, H. and BANERJEE, A. (2014). Randomized block coordinate descent for online and stochastic optimization. *arXiv preprint arXiv:1407.0107* .

XIAO, L. and ZHANG, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* **24** 2057–2075.

XU, Y. and YIN, W. (2015). Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization* **25** 1686–1716.

YUAN, X.-T., LI, P. and ZHANG, T. (2013). Gradient hard thresholding pursuit for sparsity-constrained optimization. *arXiv preprint arXiv:1311.5750* .

ZHANG, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *Information Theory, IEEE Transactions on* **57** 4689–4708.

ZHANG, T. ET AL. (2009). Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics* **37** 2109–2144.

ZHAO, T., YU, M., WANG, Y., ARORA, R. and LIU, H. (2014). Accelerated mini-batch randomized block coordinate descent method. In *Advances in Neural Information Processing Systems*.