An Accelerated Doubly Stochastic Gradient Method with Faster Explicit Model Identification

Runxue Bao
University of Pittsburgh
USA
runxue.bao@pitt.edu

Bin Gu
Nanjing University of Information Science and Technology & MBZUAI
UAE
jsgubin@gmail.com

Heng Huang
University of Pittsburgh
USA
henghuanghh@gmail.com

ABSTRACT

Sparsity regularized loss minimization problems play an important role in various fields including machine learning, data mining, and modern statistics. Proximal gradient descent method and coordinate descent method are the most popular approaches to solving the minimization problem. Although existing methods can achieve implicit model identification, aka support set identification, in a finite number of iterations, these methods still suffer from huge computational costs and memory burdens in high-dimensional scenarios. The reason is that the support set identification in these methods is implicit and thus cannot explicitly identify the low-complexity structure in practice, namely, they cannot discard useless coefficients of the associated features to achieve algorithmic acceleration via dimension reduction. To address this challenge, we propose a novel accelerated doubly stochastic gradient descent (ADSGD) method for sparsity regularized loss minimization problems, which can reduce the number of block iterations by eliminating inactive coefficients during the optimization process and eventually achieve faster explicit model identification and improve the algorithm efficiency. Theoretically, we first prove that ADSGD can achieve a linear convergence rate and lower overall computational complexity. More importantly, we prove that ADSGD can achieve a linear rate of explicit model identification. Numerically, experimental results on benchmark datasets confirm the efficiency of our proposed method.

CCS CONCEPTS

• **Computing methodologies → Feature selection**; **Regularization**; • **Mathematics of computing → Convex optimization**.

KEYWORDS

sparse learning, model identification, stochastic optimization

ACM Reference Format:
Runxue Bao, Bin Gu, and Heng Huang. 2022. An Accelerated Doubly Stochastic Gradient Method with Faster Explicit Model Identification. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557234

1 INTRODUCTION

Many popular statistical learning models, such as Lasso [27], group Lasso [32], sparse logistic regression [20], Sparse-Group Lasso [26], elastic net [35], sparse Support Vector Machine (SVM) [34], *etc*, have been developed and achieved great success for both regression and classification tasks in machine learning, data mining, and modern statistics. Given design matrix $A \in \Re^{n\times d}$ with n observations and d features, these models can be formulated as regularized loss minimization problems:

$$\min_{x\in\Re^d} \mathcal{P}(x) := \mathcal{F}(x) + \lambda\Omega(x), \tag{1}$$

where $\mathcal{F}(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x)$ is the data-fitting loss, $\Omega(x)$ is the block-separable regularizer that encourages the property of the model parameters, λ is the regularization parameter, and x is the model parameter. Let $\mathcal{G}$ be a partition of the coefficients, we have $\Omega(x) = \sum_{j=1}^{q} \Omega_j(x_{\mathcal{G}_j})$.

Proximal gradient descent (PGD) method was proposed in [5, 16] to solve Problem (1). However, at each iteration, PGD requires gradient evaluation of all the samples, which is computationally expensive. To address this issue, stochastic proximal gradient (SPG) method is proposed in [8], which only relies on the gradient of a sample at each iteration. However, SPG only achieves a sublinear convergence rate due to the gradient variance introduced by random sampling. Further, proximal stochastic variance-reduced gradient methods, such as ProxSVRG [30] and ProxSAGA [7], were proposed to achieve a linear convergence rate for strongly convex functions.

On the other hand, coordinate descent method has received increasing attention due to its efficiency. Randomized block coordinate descent (RBCD) method is proposed in [23, 24], which only updates a single block of coordinate at each iteration. However, it is still expensive because the gradient evaluation at each iteration depends on all the data points. Further, doubly stochastic gradient methods are proposed in [6, 25]. Among them, [6], called stochastic randomized block coordinate descent (SRBCD), computes the partial derivative on one coordinate block with respect to a sample. However, SRBCD can only achieve a sublinear convergence rate due to the variance of stochastic gradients. Further, accelerated mini-batch randomized block coordinate descent (MRBCD) method [29, 33] was proposed to achieve a linear convergence rate by reducing the gradient variance.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557234

Table 1: Comparison between existing methods and our ADSGD method. "Stochasticity" represents whether the method is stochastic on samples or coordinates.

Method	Stochasticity	Sample Scalablity	Low Per-iteration Cost	Model Identification	Identification Rate
PGD-type Method [16, 30]	✓	✓	✗	Implicit	–
Screening [1, 10]	✗	✗	✓	Explicit	–
ADSGD (Ours)	✓	✓	✓	Implicit & Explicit	$O(\log(\frac{1}{\epsilon_j}))$

However, all the existing methods still suffer from huge computational costs and memory usage in the practical high-dimensional scenario. The reason is that these methods require the traversal of all the features and correspondingly the overall complexity increases linearly with feature size. The pleasant surprise is that the non-smooth regularizer usually promotes the model sparsity and thus the solution of Problem (1) has only a few non-zero coefficients, aka the support set. In high-dimensional problems, model identification, aka support set identification, is a key property that can be used to speed up the optimization. If we can find these non-zero features in advance, Problem (1) can be easily solved by restricted to the support set with a significant computational gain without any loss of accuracy.

Model identification can be achieved in two ways: implicit and explicit identification. In terms of the implicit model identification, PGD can identify the support set after a finite number of iterations T [13–15], which means, for some finite $K > 0$, we have $\text{supp}(x^k) = \text{supp}(x^*)$ holds for any $k > K$. Other popular variants of PGD [15, 21] also have such an implicit identification property. However, these results can only show the existence of T and cannot give any useful estimate without the knowledge of θ^* [13, 14], which makes it implicit and impossible to explicitly discard the coefficients that must be zero at the optimum in advance. In terms of the explicit model identification, screening can identify the zero coefficients and discard these features directly [1, 3, 10]. By achieving dimension reduction, the algorithm only needs to solve a sub-problem and save much useless computation. However, first, existing works [1, 3, 10, 18, 22] mainly focus on the existence of identification but fail to show how fast we can achieve explicit model identification. Besides, existing works on explicit model identification [19] are limited to the deterministic setting. Therefore, accelerating the model training by achieving fast explicit model identification is promising and sorely needed for high-dimensional problems in the stochastic setting.

To address the challenges above, in this paper, we propose a novel Accelerated Doubly Stochastic Gradient Descent (ADSGD) method for sparsity regularized loss minimization problems, which can significantly improve the training efficiency without any loss of accuracy. On the one hand, ADSGD computes the partial derivative on one coordinate block with respect to a mini-batch of samples to simultaneously enjoy the stochasticity on both samples and features and thus achieve a low per-iteration cost. On the other hand, ADSGD not only enjoys implicit model identification of the proximal gradient method, but also enjoys explicit model identification by eliminating the inactive features. Specifically, ADSGD has two loops. We first eliminate the inactive blocks at the main loop. Within the inner loop, we only estimate the gradient over a selected active block with a mini-batch of samples. To reduce the gradient variance, we adjust the estimated gradient with the exact gradient over the selected block. Theoretically, by addressing the difficulty of the uncertainty of double stochasticity, we establish the analysis for the convergence of ADSGD. Moreover, by linking the sub-optimality gap and duality gap, we provide theoretical analysis for fast explicit model identification. Finally, based on the results of the convergence and explicit model identification ability, we establish the theoretical analysis of the overall complexity of ADSGD. Empirical results show that ADSGD can achieve a significant computational gain than existing methods. Both the theoretical and the experimental results confirm the superiority of our method. Table 1 summarizes the advantages of our ADSGD over existing methods.

Contributions. We summarize the main contributions of this paper as follows:

- We propose a novel accelerated doubly stochastic gradient descent method for generalized sparsity regularized problems with lower overall complexity and faster explicit model identification rate. To the best of our knowledge, this is the first work of doubly stochastic gradient method to achieve a linear model identification rate.
- We derive rigorous theoretical analysis for our ADSGD method for both strongly and nonstrongly convex functions. For strongly convex function, ADSGD can achieve a linear convergence rate $O(\log(\frac{1}{\epsilon}))$ and reduce the per-iteration cost from $O(d)$ to $O(s)$ where $s \ll d$, which improves existing methods with a lower overall complexity $O(s(n + \frac{T}{\mu})\log(\frac{1}{\epsilon}))$. For nonstrongly convex function, ADSGD can also achieve a lower overall complexity $O(s(n + \frac{T}{\epsilon})\log(\frac{1}{\epsilon}))$.
- We provide the theoretical guarantee of the iteration number T to achieve explicit model identification. We rigorously prove our ADSGD algorithm can achieve the explicit model identification at a linear rate $O(\log(\frac{1}{\epsilon_j}))$.

2 PRELIMINARY

2.1 Notations and Background

For norm $\Omega(\cdot)$, $\Omega^D(\cdot)$ is the dual norm and defined as $\Omega^D(u) = \max_{\Omega(z)\leq 1}\langle z, u\rangle$ for any $u \in \Re^d$ if $z \in \Re^d$. Denote θ as the dual solution and Δ_A as the feasible space of θ, the dual formulation $\mathcal{D}(\theta)$ of Problem (1) can be written as:

$$\max_{\theta\in\Delta_A} \mathcal{D}(\theta) := -\frac{1}{n}\sum_{i=1}^{n} f_i^*(-\theta_i). \tag{2}$$

The dual $\mathcal{D}(\theta)$ is strongly concave for Lipschitz gradient continuous $\mathcal{F}(x)$ (see Proposition 3.2 in [11]).

For Problem (1), based on the subdifferential [12, 17], Fermat's conditions (see [4] for Proposition 26.1) holds as:

$$\frac{1}{n}A_j^\top \theta^* \in \lambda \partial \Omega_j(x^*_{\mathcal{G}_j}). \tag{3}$$

The optimality conditions of Problem (1) can be read from (3) as:

$$\frac{1}{n}\Omega_j^D(A_j^\top \theta^*) = \lambda, \quad if\, x^*_{\mathcal{G}_j} \neq 0; \tag{4}$$

$$\frac{1}{n}\Omega_j^D(A_j^\top \theta^*) \leq \lambda, \quad if\, x^*_{\mathcal{G}_j} = 0. \tag{5}$$

2.2 Definitions and Assumptions

One key property of our method is to achieve explicit model identification, which means we can explicitly find the Equicorrelation Set in Definition 1 of the solution.

Definition 1. *(**Equicorrelation Set** (see [28])) Suppose θ^* is the dual optimal, the equicorrelation set is defined as*

$$\mathcal{S}^* := \{j \in \{1, 2, \ldots, q\} : \frac{1}{n}\Omega_j^D(A_j^\top \theta^*) = \lambda\}. \tag{6}$$

Assumption 1. *Given the partition $\{\mathcal{G}_1, \ldots, \mathcal{G}_q\}$, all $\nabla_{\mathcal{G}_j} f_i(x) = [\nabla f_i(x)]_{\mathcal{G}_j}$ are block-wise Lipschitz continuous with constant L_i, which means that for any x and x', there exists a constant $L = \max_i L_i$, we have*

$$\|\nabla_{\mathcal{G}_j} f_i(x) - \nabla_{\mathcal{G}_j} f_i(x')\| \leq L\|x_{\mathcal{G}_j} - x'_{\mathcal{G}_j}\|. \tag{7}$$

Assumption 2. *$\mathcal{F}(x)$ and $\Omega(x)$ are proper, convex and lower-semicontinuous.*

Assumptions 1 and 2 are commonly used in the convergence analysis of the RBCD method [29, 33], which are standard and satisfied by many regularized loss minimization problems. Assumption 1 implies that there exists a constant $T \leq qL$, for any x and x', we have

$$\|\nabla f_i(x) - \nabla f_i(x')\| \leq T\|x - x'\|, \tag{8}$$

i.e., $\nabla f_i(x)$ is Lipschitz continuous with constant T.

3 PROPOSED METHOD

In this section, we will introduce the accelerated doubly stochastic gradient descent (ADSGD) method with discussions.

To achieve the model identification, a naive implementation of the ADSGD method is summarized in Algorithm 1. The ASGD algorithm can reduce the size of the original optimization problem and the variables during the training process. Thus, the latter problem has a smaller size and fewer variables for training, which is a sub-problem of the problem from the previous step, but generates the same optimal solution.

We denote the original problem as $\mathcal{P}_0$ and subsequent subproblem as $\mathcal{P}_{k-1}$ at the k-th iteration in the main loop of ASGD. Moreover, we define the active set at the k-th iteration of the main loop as $\mathcal{S}_{k-1}$. Thus, in the main loop of ASGD, we compute θ_{k-1} with the active set $\mathcal{S}_{k-1}$ from the previous iteration as

$$\theta_{k-1} = \frac{-\nabla \mathcal{F}(\tilde{x}_{k-1})}{\max(1, \Omega^D(A^\top_{\mathcal{S}_{k-1}} \nabla \mathcal{F}(\tilde{x}_{k-1}))/\lambda)}. \tag{9}$$

Then we compute the intermediate duality gap

$$\mathrm{Gap}(\tilde{x}_{k-1}, \theta_{k-1}) := \mathcal{P}_{k-1}(\tilde{x}_{k-1}) - \mathcal{D}_{k-1}(\theta_{k-1}), \tag{10}$$

for the screening test. In step 6, we obtain new active set $\mathcal{S}_k$ from $\mathcal{S}_{k-1}$ by the screening conducted on all $j \in \mathcal{S}_{k-1}$ as

$$\frac{1}{n}\Omega_j^D(A_j^\top \theta_{k-1}) + \frac{1}{n}\Omega_j^D(A_j)r^{k-1} < \lambda \Rightarrow \tilde{x}^*_{\mathcal{G}_j} = 0, \tag{11}$$

where the safe region is chosen as spheres $\mathcal{R} = \mathcal{B}(\theta_{k-1}, r^{k-1})$ [1, 18].

With obtained active set $\mathcal{S}_k$, we update the design matrix A and the related parameter variables $x^0, \tilde{x}$ in step 7. In the inner loop, we conduct all the operations on $\mathcal{S}_k$. To make the algorithm scale well with the sample size, we only randomly sample a mini-batch $\mathcal{I} \in \{1, 2, \ldots, n\}$ of samples at each iteration to evaluate the gradients.

Algorithm 1 The ASGD method

```
Input: $\hat{x}_0$.
1: for $k = 1, 2, \ldots$ do
2:     $\tilde{x}_{k-1} = \hat{x}_{k-1}$.
3:     $x^0_{k-1} = \tilde{x}_{k-1}$.
4:     Compute $\theta_{k-1}$ by (9).
5:     $r^{k-1} = \sqrt{2T\,\mathrm{Gap}(\tilde{x}_{k-1}, \theta_{k-1})}$.
6:     Update $\mathcal{S}_k \subset \mathcal{S}_{k-1}$ by (11).
7:     Update $A_{\mathcal{S}_k}, x^0_k, \tilde{x}_k$ with $\mathcal{S}_k$.
8:     for $t = 1, 2, \ldots, \frac{mq_k}{q}$ do
9:         Randomly pick $\mathcal{I} \subset \{1, 2, \ldots, n\}$.
10:        $x^t_k = \mathrm{prox}_{\eta,\lambda}(x^{t-1}_k - \eta \nabla \mathcal{F}_{\mathcal{I}}(x^{t-1}_k))$.
11:    end for
12:    $\hat{x}_k = \frac{1}{m_k}\sum_{t=1}^{m_k} x^t_k$.
13: end for
Output: Coefficient $\hat{x}_k$.
```

Property 1. *Let $\hat{x}_{\mathcal{G}_j}$ be the j-th block of $\hat{x}$ in Algorithm 1, $\forall j \in \{1, 2, \ldots, q\}$, $\hat{x}_{\mathcal{G}_j}$ discarded by ASGD is guaranteed to be 0 at the optimum.*

Remark 1. *Property 1 shows that ASGD is guaranteed to be safe not only for the current iteration but also for the whole training process. The safety of the screening is the foundation of the analysis of convergence and explicit model identification rate in the following part.*

Remark 2. *Property 1 also shows that discarding inactive variables can either decrease or make no changes to the objective function.*

Doubly Stochastic Gradient Update. Since ASGD is only singly stochastic on samples, the gradient evaluation is still expensive because it depends on all the coordinates at each iteration. Thus, we randomly select a coordinate block j from $\mathcal{S}_k$ to enjoy the stochasticity on features. Specifically, ADSGD only computes the partial derivative $\nabla_{\mathcal{G}_j}\mathcal{F}_{\mathcal{I}}(x^{t-1}_k)$ on one coordinate block with respect to a sample each time, which yields a much lower per-iteration computational cost. The proximal step is computed as:

$$\mathrm{prox}^j_{\eta,\lambda}(x'_{\mathcal{G}_j}) = \underset{x_{\mathcal{G}_j}}{\arg\min}\ \frac{1}{2\eta}\|x'_{\mathcal{G}_j} - x_{\mathcal{G}_j}\|^2 + \lambda\Omega_j(x_{\mathcal{G}_j}). \tag{12}$$

Therefore, ADSGD is doubly stochastic and can scale well with both the sample size and feature size.

Algorithm 2 The ADSGD method

```
Input: $\hat{x}_0$.
1:  for $k = 1, 2, \dots$ do
2:      $\tilde{x}_{k-1} = \hat{x}_{k-1}$.
3:      $\tilde{\mu}_{k-1} = \nabla \mathcal{F}(\tilde{x}_{k-1})$.
4:      $x^0_{k-1} = \tilde{x}_{k-1}$.
5:      Compute $\theta_{k-1}$ by (9).
6:      $r^{k-1} = \sqrt{2T\,\mathrm{Gap}(\tilde{x}_{k-1}, \theta_{k-1})}$.
7:      Update $\mathcal{S}_k \subset \mathcal{S}_{k-1}$ by (11).
8:      Update $A_{\mathcal{S}_k}, x^0_k, \tilde{x}_k, \tilde{\mu}_k$ with $\mathcal{S}_k$.
9:      for $t = 1, 2, \dots, \frac{mq_k}{q}$ do
10:         Randomly pick $\mathcal{I} \subset \{1, 2, \dots, n\}$.
11:         Randomly pick $j$ from $\mathcal{S}_k$.
12:         $\mu_k = \nabla_{\mathcal{G}_j} \mathcal{F}_{\mathcal{I}}(x^{t-1}_k) - \nabla_{\mathcal{G}_j} \mathcal{F}_{\mathcal{I}}(\tilde{x}_k) + \tilde{\mu}_{\mathcal{G}_j,k}$.
13:         $x^t_{k,\mathcal{G}_j} = \mathrm{prox}^j_{\eta,\lambda}(x^{t-1}_{k,\mathcal{G}_j} - \eta \mu_k)$.
14:     end for
15:     $\hat{x}_k = \frac{1}{m_k} \sum_{t=1}^{m_k} x^t_k$
16: end for
Output: Coefficient $\hat{x}_k$.
```

Variance Reduction on the Selected Blocks. However, the gradient variance introduced by stochastic sampling does not converge to zero. Hence, a decreasing step size is required to ensure the convergence. In that case, even for strongly convex functions, we can only obtain a sublinear convergence rate. Thanks to the full gradient we computed in step 3, we can adjust the partial gradient estimation over the selected block $\mathcal{G}_j$ to reduce the gradient variance with almost no additional computational costs as:

$$\mu_k = \nabla_{\mathcal{G}_j} \mathcal{F}_{\mathcal{I}}(x^{t-1}_k) - \nabla_{\mathcal{G}_j} \mathcal{F}_{\mathcal{I}}(\tilde{x}_k) + \tilde{\mu}_{\mathcal{G}_j,k}. \tag{13}$$

It can guarantee that the variance of stochastic gradients asymptotically goes to zero. Thus, a constant step size can be used to achieve a linear convergence rate if $\mathcal{P}$ is strongly convex.

The algorithmic framework of the ADSGD method is presented in Algorithm 2. ADSGD can identify inactive blocks and thus only active blocks are updated in the inner loop, which can make more progress for training rather than conduct useless updates for inactive blocks. Thus, fewer inner loops are required for each main loop and correspondingly huge computational time is saved.

Suppose we have q_k active blocks for the inner loop at the k-th iteration, we only do $m_k = \frac{mq_k}{q}$ iterations in the inner loop for the current outer iteration, which means the number of the inner loops continues decreasing in our algorithm. The output for each iteration is the average result of the inner loops. As the algorithm converges, the duality gap converges to zero and thus (11) can eliminate more inactive blocks and thus save the computational costs to a large extent.

Remarkably, the key step of the screening in Algorithm 2 is the computation of the intermediate duality gap, which only imposes extra $O(q_k)$ costs to the original algorithm at the k-th iteration. Note the extra complexity of screening is much less than $O(d)$ in practice, which would not affect the complexity analysis of the algorithm. Further, at the k-th iteration, our Algorithm 2 only requires the computation over $\mathcal{S}_k$, which could be much smaller than the original model over the full parameter set. Hence, in high-dimensional regularized problems, the computation costs are promising to be effectively reduced with the constantly decreasing active set $\mathcal{S}_k$.

On the one hand, ADSGD enjoys the implicit model identification of the proximal gradient method. On the other hand, discarding the inactive variables can further speed up the identification rate. Thus, ADSGD is promising to achieve a fast identification rate by simultaneously enjoying the implicit and explicit model identification to yield a lower per-iteration cost.

4 THEORETICAL ANALYSIS

In this section, we first give several useful lemmas and then provide theoretical analysis on the convergence, explicit model identification, and overall complexity. Detailed proof can be found in [2].

4.1 Useful Lemmas

Lemma 1. *Define* $v_{\mathcal{I}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (\nabla f_i(x^{t-1}) - \nabla f_i(\tilde{x})) + \nabla \mathcal{F}(\tilde{x})$, $\overline{x} = \mathrm{prox}_\eta(x^{t-1} - \eta \nabla \mathcal{F}(x^{t-1}))$, $\overline{x}_{\mathcal{I}} = \mathrm{prox}_\eta(x^{t-1} - \eta v_{\mathcal{I}})$ *and* $\mathrm{prox}_\eta(x') = \arg\min_x \frac{1}{2\eta}\|x' - x\|^2 + \Omega(x)$, *we have*

$$\mathbf{E}_{\mathcal{I}}(v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1}))^\top (x^* - \overline{x}_{\mathcal{I}}) \leq \eta \mathbf{E}_{\mathcal{I}} \|v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1})\|^2. \tag{14}$$

Proof. We can first prove

$$\begin{aligned} & \mathbf{E}_{\mathcal{I}}(v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1}))^\top (x^* - \overline{x}_{\mathcal{I}}) \\ = \ & \mathbf{E}_{\mathcal{I}}[(v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1}))^\top (\overline{x} - \overline{x}_{\mathcal{I}})] \end{aligned}$$

and then prove

$$\begin{aligned} & \mathbf{E}_{\mathcal{I}}(v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1}))^\top (x^* - \overline{x}_{\mathcal{I}}) \\ = \ & \eta \mathbf{E}_{\mathcal{I}} \|v_{\mathcal{I}} - \nabla \mathcal{F}(x^{t-1})\|^2. \end{aligned} \tag{15}$$

to obtain Lemma 1 by using Cauchy-Schwarz inequality and the non-expansiveness of the proximal operator. □

Lemma 2. *(See [30]) Define* $v_i = \nabla f_i(x^{t-1}) - \nabla f_i(\tilde{x}) + \nabla \mathcal{F}(\tilde{x})$, *conditioning on* x^{t-1}, *we have*

$$\mathbf{E}_i v_i = \nabla \mathcal{F}(x^{t-1}),$$

and

$$\mathbf{E}_i \|v_i - \nabla \mathcal{F}(x^{t-1})\|_2^2 \leq 4T(\mathcal{P}(x^{t-1}) - \mathcal{P}(x^*) + \mathcal{P}(\tilde{x}) - \mathcal{P}(x^*)).$$

Lemma 3. *(See [33]) Define* $\delta = (\bar{x} - x)/\eta$ *and* $\delta_{\mathcal{G}_j} = (\bar{x}_{\mathcal{G}_j} - x)/\eta$ *where* $\bar{x} = \mathrm{prox}_\eta(x - \eta v)$ *and* $\mathrm{prox}_\eta(x) = \arg\min_{x'} \frac{1}{2\eta}\|x' - x\|^2 + \Omega(x')$, *we have*

$$\mathbf{E}_j \delta_{\mathcal{G}_j} = \delta / q \quad \text{and} \quad \mathbf{E}_j \|\delta_{\mathcal{G}_j}\|^2 = \|\delta\|^2 / q.$$

Moreover, taking $\eta \leq \frac{1}{L}$, *we have*

$$\begin{aligned} & \mathbf{E}_j[(x - x^*)^\top \delta_{\mathcal{G}_j} + \frac{\eta}{2}\|\delta_{\mathcal{G}_j}\|^2] \\ \leq \ & \frac{1}{q}\mathcal{P}(x^*) + \frac{q-1}{q}\mathcal{P}(x) - \mathbf{E}_j \mathcal{P}(\bar{x}_{\mathcal{G}_j}) \\ & + \frac{1}{q}(v - \nabla \mathcal{F}(x))^\top (x^* - \bar{x}), \end{aligned} \tag{16}$$

where $x^* = \arg\min_x \mathcal{P}(x)$.

4.2 Strongly Convex Functions

We establish the theoretical analysis of ADSGD for strongly convex $\mathcal{F}$ here.

4.2.1 Convergence Results.

Lemma 4. *Suppose $\hat{x}_k$ and $\tilde{x}_k$ are generated from the k-th iteration of the main loop in Algorithm 2 and let $|\mathcal{I}| \geq \frac{T}{L}$ and $\eta < \frac{1}{4L}$, we have*

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}_k(x_k^*) \leq \rho_k[\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)], \tag{17}$$

where $\rho_k = \frac{q_k}{\mu\eta(1-4L\eta)m_k} + \frac{4L\eta(m_k+1)}{(1-4L\eta)m_k}$. We can choose $|\mathcal{I}| = \frac{T}{L}$, $\eta = \frac{1}{16L}$, and $m = \frac{65qL}{\mu}$ to make $\rho_k < \frac{2}{3}$.

Proof. At the k-th iteration of the main loop, all the updates in the inner loop are conducted on sub-problem $\mathcal{P}_k$ with the active set $\mathcal{S}_k$. At the t-th iteration of the inner loop for the k-th main loop, we randomly sample a mini-batch $\mathcal{I}$ and $\mathcal{G}_j \subseteq \mathcal{S}_k$.

Define $v_i = \nabla f_i(x_k^{t-1}) - \nabla f_i(\tilde{x}_k) + \nabla\mathcal{F}(\tilde{x}_k)$, based on Lemma 2, conditioning on x_k^{t-1}, we have

$$\mathbf{E}_i v_i = \nabla\mathcal{F}(x_k^{t-1}),$$

and

$$\mathbf{E}_i\|v_i - \nabla\mathcal{F}(x_k^{t-1})\|_2^2 \leq 4T(\mathcal{P}(x_k^{t-1}) - \mathcal{P}(x_k^*) + \mathcal{P}(\tilde{x}_k) - \mathcal{P}(x_k^*)).$$

Define $\delta = (\bar{x}_k - x_k)/\eta$ and $\delta_{\mathcal{G}_j} = (\bar{x}_{\mathcal{G}_j,k} - x_k)/\eta$ where $\bar{x}_k = \text{prox}_\eta(x_k - \eta v)$, based on Lemma 3, we have

$$\mathbf{E}_j\delta_{\mathcal{G}_j} = \delta/q_k \quad \text{and} \quad \mathbf{E}_j\|\delta_{\mathcal{G}_j}\|^2 = \|\delta\|^2/q_k.$$

Moreover, taking $\eta \leq \frac{1}{L}$, we have

$$\begin{aligned}
& \mathbf{E}_j[(x_k - x_k^*)^\top\delta_{\mathcal{G}_j} + \frac{\eta}{2}\|\delta_{\mathcal{G}_j}\|^2] \\
\leq\ & \frac{1}{q_k}\mathcal{P}_k(x_k^*) + \frac{q_k-1}{q_k}\mathcal{P}_k(x_k) - \mathbf{E}_j\mathcal{P}_k(\bar{x}_{\mathcal{G}_j,k}) && (18)\\
& +\frac{1}{q_k}(v - \nabla\mathcal{F}(x_k))^\top(x_k^* - \bar{x}_k). && (19)
\end{aligned}$$

Define $\delta_{\mathcal{I},\mathcal{G}_j} = (x_k^t - x_k^{t-1})/\eta$ and $\bar{x}_{k,\mathcal{I}} = \text{prox}_\eta(x_k^{t-1} - \eta v_{\mathcal{I}})$, based on Lemma 1, Lemma 2, Lemma 3 and the fact that $\mathbf{E}_{\mathcal{I}}v_{\mathcal{I}}$ is the unbiased estimator of $\nabla\mathcal{F}(x_k^{t-1})$ and

$$\mathbf{E}_{\mathcal{I}}\|v_{\mathcal{I}} - \nabla\mathcal{F}(x_k^{t-1})\|_2^2 = \frac{1}{|\mathcal{I}|}\mathbf{E}_i\|v_i - \nabla\mathcal{F}(x_k^{t-1})\|_2^2, \tag{20}$$

we can prove

$$\begin{aligned}
& \mathbf{E}_{\mathcal{I},j}\|x_k^t - x_k^*\|^2 - \|x_k^{t-1} - x_k^*\|^2 \\
=\ & \mathbf{E}_{\mathcal{I},j}\|x_k^{t-1} + \eta\delta_{\mathcal{I},\mathcal{G}_j} - x_k^*\|^2 - \|x_k^{t-1} - x_k^*\|^2 && (21)\\
=\ & \mathbf{E}_{\mathcal{I}}[2\eta(x_k^{t-1} - x_k^*)^\top\mathbf{E}_j[\delta_{\mathcal{I},\mathcal{G}_j}] + \eta^2\mathbf{E}_j\|\delta_{\mathcal{I},\mathcal{G}_j}\|^2] && (22)\\
\leq\ & \frac{2\eta}{q_k}\mathbf{E}_{\mathcal{I}}[(v_{\mathcal{I}} - \nabla\mathcal{F}(x_k^{t-1}))^\top(x_k^* - \bar{x}_{k,\mathcal{I}})] && (23)\\
& +2\eta\mathbf{E}_{\mathcal{I}}[\frac{1}{q_k}\mathcal{P}_k(x_k^*) + \frac{q_k-1}{q_k}\mathcal{P}_k(x_k^{t-1}) - \mathbf{E}_j\mathcal{P}_k(\bar{x}_{\mathcal{G}_j,k,\mathcal{I}})] \\
\leq\ & -2\eta[\mathbf{E}_{\mathcal{I},j}\mathcal{P}_k(\bar{x}_{\mathcal{G}_j,k,\mathcal{I}}) - \mathcal{P}_k(x_k^*)] && (24)\\
& +2\eta\frac{q_k-1}{q_k}(\mathcal{P}_k(x_k^{t-1}) - \mathcal{P}_k(x_k^*)) \\
& +\frac{8T\eta^2}{q_k|\mathcal{I}|}(\mathcal{P}_k(x_k^{t-1}) - \mathcal{P}_k(x_k^*) + \mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)). && (25)
\end{aligned}$$

Note $x_k^0 = \tilde{x}_k$ and $\hat{x}_k = \frac{1}{m_k}\sum_{t=1}^{m_k}x_k^t$, then based on the inequality obtained by (25), we have

$$\begin{aligned}
& \mathbf{E}\|x_k^{m_k} - x_k^*\|^2 - \|x_k^0 - x_k^*\|^2 + 2\eta\sum_{t=1}^{m_k}(\mathbf{E}\mathcal{P}_k(x_k^t) - \mathcal{P}_k(x_k^*)) \\
\leq\ & \frac{8T\eta^2/|\mathcal{I}| + 2\eta(q_k-1)}{q_k}\sum_{t=1}^{m_k-1}(\mathbf{E}\mathcal{P}_k(x_k^t) - \mathcal{P}_k(x_k^*)) \\
& +\frac{8T\eta^2(m_k+1)}{q_k|\mathcal{I}|}(\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)) && (26)\\
\leq\ & \frac{8T\eta^2/|\mathcal{I}| + 2\eta(q_k-1)}{q_k}\sum_{t=1}^{m_k}(\mathbf{E}\mathcal{P}_k(x_k^t) - \mathcal{P}_k(x_k^*)) \\
& +\frac{8T\eta^2(m_k+1)}{q_k|\mathcal{I}|}(\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)). && (27)
\end{aligned}$$

Rearranging (27), note $x_k^0 = \tilde{x}_k$, we have

$$\begin{aligned}
& 2\eta\left(\frac{1 - 4\eta T/|\mathcal{I}|}{q_k}\right)\sum_{t=1}^{m_k}[\mathbf{E}\mathcal{P}_k(x_k^t) - \mathcal{P}_k(x_k^*)] && (28)\\
\leq\ & \|x_k^0 - x_k^*\|^2 + \frac{8T\eta^2(m_k+1)}{q_k|\mathcal{I}|}(\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)) \\
\leq\ & \frac{2}{\mu}(\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)) + \frac{8T\eta^2(m_k+1)}{q_k|\mathcal{I}|}(\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)).
\end{aligned}$$

where the second inequality is obtained by the strong convexity of $\mathcal{P}$. Based on the convexity of $\mathcal{P}$, we have

$$\mathcal{P}_k(\hat{x}_k) \leq \frac{1}{m_k}\sum_{t=1}^{m_k}\mathcal{P}_k(x_k^t).$$

Thus, by (28), we can obtain

$$\begin{aligned}
& 2\eta\left(\frac{1 - 4\eta T/|\mathcal{I}|}{q_k}\right)m_k\left[\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}_k(x_k^*)\right] \\
\leq\ & \left(\frac{2}{\mu} + \frac{8T\eta^2(m_k+1)}{q_k|\mathcal{I}|}\right)\left[\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)\right]. && (29)
\end{aligned}$$

Defining $\rho_k = \left(\frac{q_k}{\mu\eta(1-4\eta T/|\mathcal{I}|)m_k} + \frac{4\eta T/|\mathcal{I}|(m_k+1)}{(1-4\eta T/|\mathcal{I}|)m_k}\right)$, we have

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}_k(x_k^*) \leq \rho_k\left[\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)\right]. \tag{30}$$

Choosing $|\mathcal{I}| = \frac{T}{L}$, we have

$$\begin{aligned}
& \mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}_k(x_k^*) && (31)\\
\leq\ & \left(\frac{q_k}{\mu\eta(1-4L\eta)m_k} + \frac{4L\eta(m_k+1)}{(1-4L\eta)m_k}\right)\left[\mathcal{P}_k(\tilde{x}_k) - \mathcal{P}_k(x_k^*)\right].
\end{aligned}$$

Considering $m_k = \frac{mq_k}{q}$, we can choose $\eta = \frac{1}{16L}$, and $m = \frac{65qL}{\mu}$ to make $\rho_k < \frac{2}{3}$, which completes the proof.

□

Remark 3. *Lemma 4 shows that the overall inner loop of Algorithm 2 can decrease the expected objective function with a factor ρ_k at the k-th iteration.*

Theorem 1. *Suppose $\hat{x}_k$ be generated from the k-th iteration of the main loop in Algorithm 2 and let $|\mathcal{I}| \geq \frac{T}{L}$ and $\eta < \frac{1}{4L}$, we have*

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \rho^k[\mathcal{P}(\hat{x}) - \mathcal{P}(x^*)]. \tag{32}$$

We can choose $|\mathcal{I}| = \frac{T}{L}$, $\eta = \frac{1}{16L}$, *and* $m = \frac{65qL}{\mu}$ *to make* $\rho < \frac{2}{3}$.

Proof. Considering each sub-problem $\mathcal{P}_k$, based on Lemma 4, for each main loop, we have

$$\mathbf{E}\mathcal{P}_k\left(\hat{x}_k\right) - \mathcal{P}_k(x_k^*) \leq \rho_k\left[\mathcal{P}_k\left(\tilde{x}_k\right) - \mathcal{P}_k(x_k^*)\right]. \tag{33}$$

Since the eliminating in the ADSGD algorithm is safe, the optimal solution of all the sub-problems are the same. Thus, we have

$$\mathcal{P}_k(x_k^*) = \mathcal{P}_{k-1}(x_{k-1}^*). \tag{34}$$

Then, the coefficients eliminated at the k-th iteration of the main loop must be zeroes at the optimal, which means the eliminating stage at the k-th iteration of the main loop actually minimizes the sub-problem $\mathcal{P}_{k-1}$ over the eliminated variables. Thus, we have

$$\mathcal{P}_k(\tilde{x}_k) \leq \mathcal{P}_{k-1}(\tilde{x}_{k-1}).$$

Moreover, considering $\mathcal{P}_{k-1}(\hat{x}_{k-1}) = \mathcal{P}_{k-1}(\tilde{x}_{k-1})$, we have

$$\mathcal{P}_k(\tilde{x}_k) \leq \mathcal{P}_{k-1}(\hat{x}_{k-1}). \tag{35}$$

Combining above, we have

$$\begin{aligned}\mathbf{E}\mathcal{P}_k\left(\hat{x}_k\right) - \mathcal{P}_k(x_k^*) &\leq \rho_k\left[\mathcal{P}_k\left(\tilde{x}_k\right) - \mathcal{P}_k(x_k^*)\right] & (36)\\ &\leq \rho_k\left[\mathcal{P}_{k-1}(\tilde{x}_{k-1}) - \mathcal{P}_{k-1}(x_{k-1}^*)\right] & (37)\end{aligned}$$

We can choose $|\mathcal{I}| = \frac{T}{L}$, $\eta = \frac{1}{16L}$, and $m = \frac{65qL}{\mu}$ to make $\forall k, \rho_k < \frac{2}{3}$. Thus, $\exists \rho < \frac{2}{3}$, applying the above inequality recursively, we obtain

$$\mathbf{E}\mathcal{P}_k\left(\hat{x}_k\right) - \mathcal{P}_k(x_k^*) \leq \rho^k\left[\mathcal{P}_0(\hat{x}_{\mathcal{S}_0}) - \mathcal{P}_0(x_{\mathcal{S}_0}^*)\right].$$

From (34), we have

$$\mathcal{P}_k(x_k^*) = \mathcal{P}_{k-1}(x_{k-1}^*) = \cdots = \mathcal{P}_0(x_{\mathcal{S}_0}^*). \tag{38}$$

Note $\mathcal{P}_0 = \mathcal{P}$ and $\mathcal{S}_0$ is the universe set for all the variables blocks, we have

$$\mathbf{E}\mathcal{P}_k\left(\hat{x}_k\right) - \mathcal{P}(x^*) \leq \rho^k\left[\mathcal{P}(\hat{x}) - \mathcal{P}(x^*)\right], \tag{39}$$

which completes the proof. □

Remark 4. *Theorem 1 shows that ADSGD converges linearly to the optimal with the convergence rate* $O(\log_{\frac{1}{\rho}}(\frac{1}{\epsilon}))$.

4.2.2 Explicit Model Identification.

Lemma 5. $\lim_{k\to+\infty}\theta_k = \theta^*$ *if* $\lim_{k\to+\infty}\hat{x}_k = x^*$.

Proof. Define $\alpha_k = \max\left(1, \Omega^D\left(A_{\mathcal{S}_k}^\top \nabla\mathcal{F}(\tilde{x}_k)\right)/\lambda\right)$, We have

$$\begin{aligned}\left\|\theta_k - \theta^*\right\|_2 &= \left\|\frac{\nabla\mathcal{F}\left(\tilde{x}_k\right)}{\alpha_k} - \nabla\mathcal{F}(x^*)\right\|_2 & (40)\\ &\leq \left|1 - \frac{1}{\alpha_k}\right|\|\nabla\mathcal{F}\left(\tilde{x}_k\right)\|_2 + \left\|\nabla\mathcal{F}\left(\tilde{x}_k\right) - \nabla\mathcal{F}(x^*)\right\|_2.\end{aligned}$$

Considering the right term, if $\lim_{k\to+\infty}\hat{x}_k = \hat{x}^*$, we have

$$\alpha_k \to \max\left(1, \Omega^D\left(A_{\mathcal{S}^*}^\top \nabla\mathcal{F}(x^*)\right)/\lambda\right) = 1,$$

and

$$\left\|\nabla\mathcal{F}\left(\tilde{x}_k\right) - \nabla\mathcal{F}(x^*)\right\|_2 \to 0.$$

Thus, the right term converges to zero, which completes the proof. □

Remark 5. *Lemma 5 shows the convergence of the dual solution is guaranteed by the convergence of the primal solution.*

Lemma 6. $\exists u \in \partial\Omega^D(A_{\mathcal{S}_k}^\top\theta_k)$, *for all* $s \in [0,1]$, *we have*

$$\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \geq s\,\mathrm{Gap}(\hat{x}_k, \theta_k) - \frac{Ts^2}{2}\|A_{\mathcal{S}_k}(u - \hat{x}_k)\|^2. \tag{41}$$

Proof. From the smoothness of $\mathcal{F}$, we have:

$$\begin{aligned}&\mathcal{F}(\hat{x}_k) - \mathcal{F}(\hat{x}_k + s(u - \hat{x}_k))\\ \geq\ & -s\langle\nabla\mathcal{F}(\hat{x}_k), A_{\mathcal{S}_k}(u - \hat{x}_k)\rangle - \frac{s^2T}{2}\|A_{\mathcal{S}_k}(u - \hat{x}_k)\|^2. & (42)\end{aligned}$$

By the convexity of Ω, we have:

$$\Omega(\hat{x}_k) - \Omega(\hat{x}_k + s(u - \hat{x}_k)) \geq s(\Omega(\hat{x}_k) - \Omega(u)). \tag{43}$$

Moreover, we have

$$\begin{aligned}&\Omega(\hat{x}_k) - \Omega(u) - \langle\nabla\mathcal{F}(\hat{x}_k), A_{\mathcal{S}_k}(u - \hat{x}_k)\rangle\\ =\ &\Omega(\hat{x}_k) + \Omega^D(A_{\mathcal{S}_k}^\top\theta_k) + \langle\nabla\mathcal{F}(\hat{x}_k), A_{\mathcal{S}_k}\hat{x}_k\rangle & (44)\\ =\ &\Omega(\hat{x}_k) + \Omega^D(A_{\mathcal{S}_k}^\top\theta_k) + \mathcal{F}(\hat{x}_k) + \mathcal{F}^*(-\theta_k) & (45)\\ =\ &\mathrm{Gap}(\hat{x}_k, \theta_k), & (46)\end{aligned}$$

where the first equality comes from

$$\Omega(u) = \langle u, A_{\mathcal{S}_k}^\top\theta_k\rangle - \Omega^D(A_{\mathcal{S}_k}^\top\theta_k),$$

and the third equality comes from

$$\mathcal{F}(\hat{x}_k) = \langle\nabla\mathcal{F}(\hat{x}_k), A_{\mathcal{S}_k}\hat{x}_k\rangle - \mathcal{F}^*(-\theta_k).$$

Therefore, for any $\hat{x}_k$ and u, we have:

$$\begin{aligned}&\mathcal{P}(\hat{x}_k) - \mathcal{P}(x^*) & (47)\\ \geq\ &\mathcal{P}(\hat{x}_k) - \mathcal{P}(\hat{x}_k + s(u - \hat{x}_k)) & (48)\\ =\ &\Omega(\hat{x}_k) - \Omega(\hat{x}_k + s(u - \hat{x}_k)) + \mathcal{F}(\hat{x}_k) - \mathcal{F}(\hat{x}_k + s(u - \hat{x}_k))\\ \geq\ &s\,\mathrm{Gap}(\hat{x}_k, \theta_k) - \frac{s^2}{2}T\|A_{\mathcal{S}_k}(u - \hat{x}_k)\|^2, & (49)\end{aligned}$$

where the second equality comes from (42) and (43), which completes the proof.

□

Remark 6. *The difficulty to analyze the model identification rate of ADSGD is that the screening is conducted on the duality gap while the convergence of the algorithm is analyzed based on the sub-optimality gap. Note the sub-optimality gap at the* k*-th iteration of the main loop is computed as* $\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*)$*, Lemma 6 links the sub-optimality gap and the duality gap at the* k*-th iteration.*

Theorem 2. *Define* $\Delta_j \triangleq \frac{n\lambda - \Omega_j^D(A_j^\top\theta^*)}{2\Omega_j^D(A_j)}$, *denote* σ_A^2 *as the spectral norm of A, suppose* Ω *has a bounded support within a ball of radius M, given any* $\gamma \in (0,1)$, *any block that* $j \notin \mathcal{S}^*$ *are correctly identified by ADSGD at iteration* $\log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j})$ *with at least probability* $1-\gamma$ *where* ρ *is from Theorem 1 and* $\epsilon_j = \frac{1}{32}\frac{\Delta_j^4\gamma}{T^3\sigma_A^2M^2(\mathcal{P}(\hat{x}) - \mathcal{P}(x^*))}$.

Proof. Based on the screening condition, any variable block j can be identified at the k-th iteration when

$$\frac{1}{n}\Omega_j^D(A_j^\top\theta^*) \leq \frac{1}{n}\Omega_j^D(A_j^\top\theta^*) + \frac{2}{n}\Omega_j^D(A_j)r^k < \lambda. \tag{50}$$

Thus, we have that variable block j can be identified when

$$r^k < \frac{\lambda - \frac{1}{n}\Omega_j^D(A_j^\top\theta^*)}{\frac{2}{n}\Omega_j^D(A_j)} \triangleq \Delta_j.$$

Considering $\tilde{x}_k = \hat{x}_k$ and $r^k = \sqrt{2T\,\mathrm{Gap}(\tilde{x}_k, \theta_k)}$, we have that any variable block j can be identified at the k-th iteration when

$$\mathrm{Gap}(\hat{x}_k, \theta_k) < \frac{\Delta_j^2}{2T}. \tag{51}$$

For $u_k \in \partial\Omega^D(A_{\mathcal{S}_k}^\top \theta_k)$, considering Ω has a bounded support within a ball of radius M, we have

$$\|A(u_k - \hat{x}_k)\| \leq 2\sigma_A M.$$

Based on Lemma 6, we have

$$\mathrm{Gap}(\hat{x}_k, \theta_k) \leq \frac{1}{s}(\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*)) + 2T\sigma_A^2 M^2 s.$$

Minimizing the right term over s, we have

$$\mathrm{Gap}(\hat{x}_k, \theta_k) \leq \sqrt{8T\sigma_A^2 M^2(\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*))}. \tag{52}$$

Thus, we can make

$$\sqrt{8T\sigma_A^2 M^2(\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*))} \leq \frac{\Delta_j^2}{2T}, \tag{53}$$

to ensure the screening condition (51) holds. Since (53) can be reformulated as

$$\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}, \tag{54}$$

if $\exists k \in \mathcal{N}^+$, we have (54) hold, we can ensure the screening condition hold at the k-iteration.

From Theorem 1, we have

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \rho^k[\mathcal{P}(\hat{x}) - \mathcal{P}(x^*)]. \tag{55}$$

Thus, if we let $(\mathcal{P}(\hat{x}) - \mathcal{P}(x^*))\rho^k \leq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}$, we have

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}. \tag{56}$$

By Markov inequality and Theorem 1, we have

$$\begin{aligned} & \mathbf{P}\left(\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \geq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}\right) \\ \leq\ & \frac{32T^3\sigma_A^2 M^2}{\Delta_j^4}(\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*)) && (57) \\ \leq\ & \frac{32T^3\sigma_A^2 M^2}{\Delta_j^4}\rho^k(\mathcal{P}(\hat{x}_k) - \mathcal{P}(x^*)). && (58) \end{aligned}$$

Denote $\epsilon_j = \frac{1}{32}\frac{\Delta_j^4\gamma}{T^3\sigma_A^2 M^2(\mathcal{P}(\hat{x}) - \mathcal{P}(x^*))}$, if we choose $k \geq \log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j})$, we have

$$\begin{aligned} & \mathbf{P}\left(\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \geq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}\right) \\ \leq\ & \frac{32T^3\sigma_A^2 M^2}{\Delta_j^4}\epsilon_j(\mathcal{P}(\hat{x}_k) - \mathcal{P}(x^*)) = \gamma. && (59) \end{aligned}$$

Thus, for $k \geq \log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j})$, we have

$$\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \frac{1}{32}\frac{\Delta_j^4}{T^3\sigma_A^2 M^2}, \tag{60}$$

with at least probability $1-\gamma$, which means any variable that $j \notin \mathcal{S}^*$ are correctly detected and successfully eliminated by the ADSGD algorithm at iteration $\log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j})$ with at least probability $1-\gamma$, which completes the proof. □

Remark 7. *Theorem 2 shows that the equicorrelation set $\mathcal{S}^*$ can be identified by ADSGD at a linear rate $O(\log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j}))$ with at least probability $1-\gamma$. We use the Lipschitzing trick in [9] to restrict the function Ω within a bounded support.*

4.2.3 Overall Complexity.

Corollary 1. *Suppose the size of the active features in set $\mathcal{S}_k$ is d_k and d^* is the size of the active features in $\mathcal{S}^*$, given any $\gamma \in (0,1)$, let $K_m = O(\log_{\frac{1}{\rho}}(\frac{1}{\epsilon_j}))$, we have d_k is decreasing and d_{K_m} equals to d^* with at least probability $1-\gamma$. Define $s = \frac{1}{K_c}\sum_{k=1}^{K_c} d_k$ where $K_c = O(\log_{\frac{1}{\rho}}(\frac{1}{\epsilon}))$, the overall complexity of ADSGD is $O((n+\frac{T}{\mu})s\log(\frac{1}{\epsilon}))$.*

Proof. The first part of Corollary 1 is the direct result of Theorem 2. For the second part, Theorem 1 shows that the ADSGD method converges to the optimal with the convergence rate $O(\log\frac{1}{\epsilon})$. For each main loop, the algorithm runs m_k inner loops. Thus, since $m = \frac{65qL}{\mu}$ and $m_k = \frac{mq_k}{q}$, the ADSGD algorithm takes $O((1+\frac{q_k L}{\mu})\log\frac{1}{\epsilon})$ iterations to achieve ϵ error.

For the computational complexity, considering the k-th iteration of the main loop, the algorithm is solving the sub-problem $\mathcal{P}_k$ and the complexity of the outer loop is $O(nd_k)$. Within the inner loop, the complexity of each iteration is $\frac{d_k|\mathcal{I}|}{q_k}$. Thus, the complexity of the inner loop is $m_k\frac{d_k|\mathcal{I}|}{q_k}$. Let $|\mathcal{I}| = \frac{T}{L}$, define $s = \frac{1}{K_c}\sum_{k=1}^{K_c} d_k$ where $K_c = O(\log_{\frac{1}{\rho}}(\frac{1}{\epsilon}))$, the overall complexity for the ADSGD algorithm is $O((n+\frac{T}{\mu})s\log\frac{1}{\epsilon})$, which completes the proof.

□

Remark 8. *Please note the difference between the number of features as d_k and the number of blocks as q_k at the k-th iteration. In the high-dimensional setting, we have $d^* \ll d$ and $s \ll d$. Thus, Corollary 1 shows that ADSGD can simultaneously achieve linear convergence rate and low per-iteration cost, which improves Prox-SVRG and MRBCD with the overall complexity $O((n+\frac{T}{\mu})d\log(\frac{1}{\epsilon}))$ at a large extent in practice.*

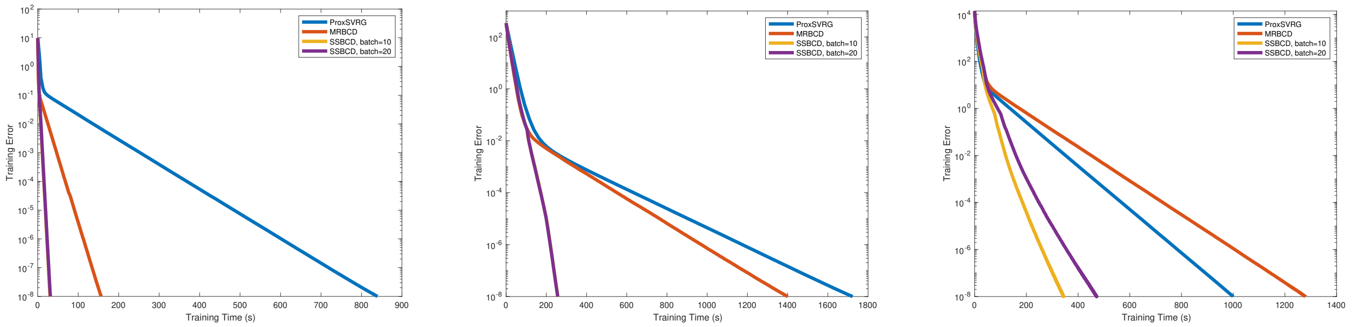
4.3 Nonstrongly Convex Functions

For nonstrongly convex $\mathcal{F}$, we can use a perturbation approach to establishing the convergence analysis here.

Suppose x^0 is the initial input and μ_p is a positive parameter, adding a perturbation term $\mu_p\|x - x^0\|^2$ to Problem (1), we have:

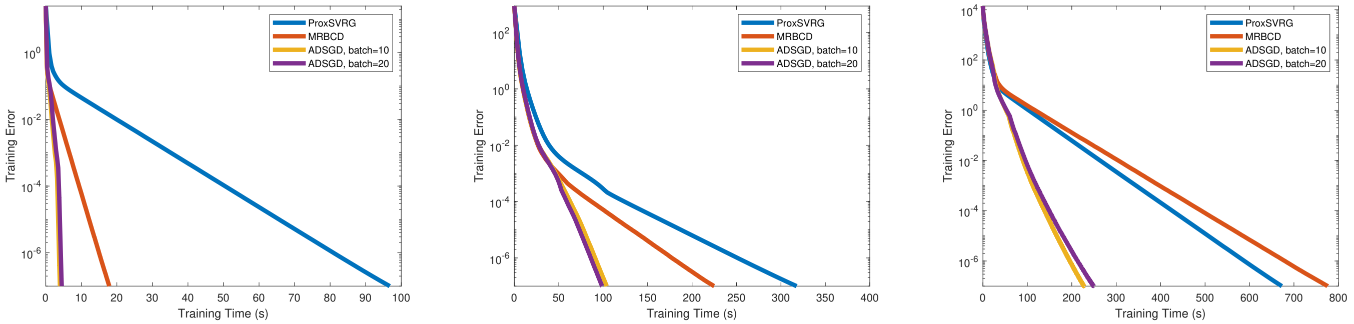
$$\min_{x\in\mathfrak{R}^d} \mathcal{F}(x) + \mu_p\|x - x^0\|^2 + \Omega(x). \tag{61}$$

If we solve (61) with ADSGD, since we can treat $\mathcal{F}(x)+\mu_p\|x-x^0\|^2$ as the data-fitting loss, we know the loss is μ_p-strongly convex, we can obtain the convergence result for (61) as $O((n+\frac{T}{\mu_p})s\log(\frac{1}{\epsilon}))$. Suppose $\hat{x}_k$ be generated from the k-th iteration of the main loop in



(a) PlantGo (b) Protein (c) Real-sim

Figure 1: Convergence results of different algorithms for Lasso on different datasets.



(a) PlantGo (b) Protein (c) Real-sim

Figure 2: Convergence results of different algorithms for Lasso on different datasets.

Algorithm 2 where $k = O((n + \frac{T}{\mu_p})s \log \frac{2}{\epsilon})$ and $\vec{x}^*$ is the optimum solution of (61), let $|\mathcal{I}| \geq \frac{T}{L}$ and $\eta < \frac{1}{4L}$, we obtain:

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) + C_p - \mathcal{P}(\vec{x}^*) - \mu_p\|\vec{x}^* - x^0\|^2 \leq \frac{\epsilon}{2}, \tag{62}$$

where C_p is the expectation of the perturbation term, which is always positive. Thus, we have

$$\begin{aligned} \mathbf{E}\mathcal{P}_k(\hat{x}_k) &\leq \frac{\epsilon}{2} + \mathcal{P}(\vec{x}^*) + \mu_p\|\vec{x}^* - x^0\|^2 && (63)\\ &\leq \frac{\epsilon}{2} + \mathcal{P}(x^*) + \mu_p\|x^* - x^0\|^2 && (64) \end{aligned}$$

where the second inequality is obtained because $\vec{x}^*$ is the optimum solution of (61).

If we set $\mu_p = \frac{\epsilon}{2\|x^* - x^0\|^2}$, we have

$$\mathbf{E}\mathcal{P}_k(\hat{x}_k) - \mathcal{P}(x^*) \leq \epsilon.$$

Since $2\|x^* - x^0\|^2$ is a constant, the overall complexity of Algorithm 1 for nonstrongly convex function is $k = O((n + \frac{T}{\epsilon})s \log(\frac{1}{\epsilon}))$, which also improves ProxSVRG and MRBCD with the overall complexity $O((n + \frac{T}{\epsilon})d \log(\frac{1}{\epsilon}))$ for nonstrongly convex function.

5 EXPERIMENTS

5.1 Experimental Setup

Design of Experiments: We perform extensive experiments on real-world datasets for two popular sparsity regularized models Lasso shown as

$$\min_{x \in \Re^d} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - x_i^\top x)^2 + \lambda\|x\|_1, \tag{65}$$

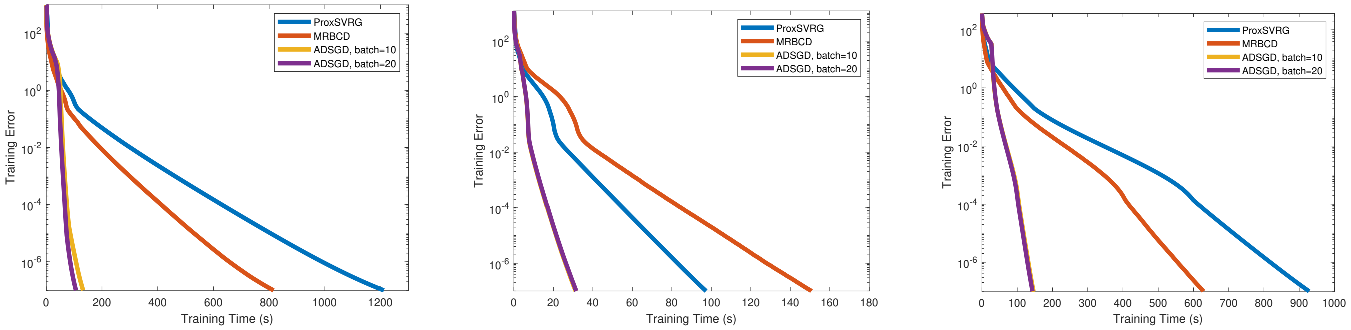
and sparse logistic regression as

$$\min_{x \in \Re^d} \frac{1}{n} \sum_{i=1}^{n} (-y_i a_i^\top x + \log(1 + \exp(a_i^\top x))) + \lambda\|x\|_1. \tag{66}$$

respectively to demonstrate the superiority of our ADSGD *w.r.t.* the efficiency.

To validate the efficiency of ADSGD, we compare the convergence results of ADSGD w.r.t the running time with competitive algorithms ProxSVRG [30] and MRBCD [29, 33] under different setups. We do not include the results of ASGD because the naive implementation is very slow. The batch size of ADSGD is chosen as 10 and 20 respectively.

Datasets: Table 2 summarizes the benchmark datasets used in our experiments. Protein, Real-sim, Gisette, Mnist, and Rcv1.binary datasets are from the LIBSVM repository, which is available at



(a) Gisette (b) Mnist (c) Rcv1.binary

Figure 3: Convergence results of different algorithms for sparse logistic regression on different datasets.

Table 2: The descriptions of the datasets.

Dataset	Samples	Features
PlantGO	978	3091
Protein	17766	357
Real-sim	72309	20958
Gisette	6000	5000
Mnist	60000	780
Rcv1.binary	20242	47236

https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. PlantGO is from [31], which is available at http://www.uco.es/kdis/mllresources/. Note Mnist is the binary version of the original ones by classifying the first half of the classes versus the left ones.

Implementation Details: All the algorithms are implemented in MATLAB. We compare the average running CPU time of different algorithms. The experiments are evaluated on a 2.30 GHz machine. For the convergence results of Lasso and sparse logistic regression, we present the results with $\lambda_1 = \lambda_{\max}/2$ and $\lambda_2 = \lambda_{\max}/4$. Notably, $\lambda_{\max}$ is a parameter that, for all $\lambda \geq \lambda_{\max}$, x^* must be 0. Specifically, we have $\lambda_{\max} = \frac{1}{n}\|A^\top y\|_\infty$ for Lasso and $\lambda_{\max} = \frac{1}{n}\|A^\top G(0)\|_\infty$ for sparse logistic regression where $G(\theta) \triangleq \frac{e^\theta}{1+e^\theta} - y$. Please note, for each setting, all the compared algorithms share the same hyperparameters for a fair comparison. We set the mini-batch size as 10 for the compared algorithms. The coordinate block number is set as $q = 10$. Other hyperparameters include the initial inner loop number m and step size η, which are selected to achieve the best performance. For Lasso, we perform the experiments on PlantGO, Protein, and Real-sim. For sparse logistic regression, we perform the experiments on Gisette, Mnist, and Rcv1.binary.

5.2 Experimental Results

5.2.1 Lasso Regression. Figures 1(a)-(c) provide the results of the convergence results for Lasso on three datasets with $\lambda = \lambda_1$. Figures 2(a)-(c) provide the convergence results with $\lambda = \lambda_2$. The results confirm that ADSGD always converges much faster than MRBCD and ProxSVRG under different setups, even when $n \gg d$ for Protein. This is because, as the variables are discarded, the optimization process is mainly conducted on a sub-problem with a much smaller size and thus requires fewer inner loops. Meanwhile, the screening step imposes almost no additional costs on the algorithm. Thus, ADSGD can achieve a lower overall complexity, compared to MRBCD and ProxSVRG conducted on the full model.

5.2.2 Sparse Logistic Regression. Figures 3(a)-(c) provide the convergence results for sparse logistic regression on three datasets with $\lambda = \lambda_1$. The results also show that ADSGD spends much less running time than MRBCD and ProxSVRG for all the datasets, even when $n \gg d$ for Mnist dataset. This is because our method solves the models with a smaller size and the screening step imposes almost no additional costs for the algorithm.

6 CONCLUSION

In this paper, we proposed an accelerated doubly stochastic gradient descent for sparsity regularized minimization problem with linear predictors, which can save much useless computation by constantly identifying the inactive variables without any loss of accuracy. Theoretically, we proved that our ADSGD method can achieve lower overall computational complexity and linear rate of explicit model identification. Extensive experiments on six benchmark datasets for popular regularized models demonstrated the efficiency of our method.

ACKNOWLEDGMENTS

This work was partially supported by NSF IIS 1845666, 1852606, 1838627, 1837956, 1956002, 2217003.

REFERENCES

[1] Runxue Bao, Bin Gu, and Heng Huang. 2020. Fast OSCAR and OWL regression via safe screening rules. In *International Conference on Machine Learning*. PMLR, 653–663.

[2] Runxue Bao, Bin Gu, and Heng Huang. 2022. An Accelerated Doubly Stochastic Gradient Method with Faster Explicit Model Identification. *arXiv preprint arXiv:2208.06058* (2022).

[3] Runxue Bao, Xidong Wu, Wenhan Xian, and Heng Huang. 2022. Doubly Sparse Asynchronous Learning for Stochastic Composite Optimization. In *IJCAI*.

[4] Heinz H Bauschke, Patrick L Combettes, et al. 2011. *Convex analysis and monotone operator theory in Hilbert spaces*. Vol. 408. Springer.

[5] Patrick L Combettes and Valérie R Wajs. 2005. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation* 4, 4 (2005), 1168–1200.

[6] Cong D Dang and Guanghui Lan. 2015. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization* 25, 2 (2015), 856–881.

[7] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*. 1646–1654.

[8] John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research* 10 (2009), 2899–2934.

[9] Celestine Dünner, Simone Forte, Martin Takác, and Martin Jaggi. 2016. Primal-dual rates and certificates. In *International Conference on Machine Learning*. PMLR, 783–792.

[10] Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. 2015. Mind the duality gap: safer rules for the Lasso. In *International Conference on Machine Learning*. 333–342.

[11] Tyler Johnson and Carlos Guestrin. 2015. Blitz: A principled meta-algorithm for scaling sparse optimization. In *International Conference on Machine Learning*. 1171–1179.

[12] A Ya Kruger. 2003. On fréchet subdifferentials. *Journal of Mathematical Sciences* 116, 3 (2003), 3325–3358.

[13] Adrian S Lewis. 2002. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization* 13, 3 (2002), 702–725.

[14] Adrian S Lewis and Stephen J Wright. 2016. A proximal method for composite minimization. *Mathematical Programming* 158, 1 (2016), 501–546.

[15] Jingwei Liang, Jalal Fadili, and Gabriel Peyré. 2017. Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. *SIAM Journal on Optimization* 27, 1 (2017), 408–437.

[16] Pierre-Louis Lions and Bertrand Mercier. 1979. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* 16, 6 (1979), 964–979.

[17] Boris S Mordukhovich, Nguyen Mau Nam, and ND Yen. 2006. Fréchet subdifferential calculus and optimality conditions in nondifferentiable programming. *Optimization* 55, 5-6 (2006), 685–708.

[18] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. 2017. Gap safe screening rules for sparsity enforcing penalties. *The Journal of Machine Learning Research* 18, 1 (2017), 4671–4703.

[19] Eugene Ndiaye, Olivier Fercoq, and Joseph Salmon. 2020. Screening rules and its complexity for active set identification. *arXiv preprint arXiv:2009.02709* (2020).

[20] Andrew Y Ng. 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*. 78.

[21] Clarice Poon, Jingwei Liang, and Carola Schoenlieb. 2018. Local convergence properties of SAGA/Prox-SVRG and acceleration. In *International Conference on Machine Learning*. PMLR, 4124–4132.

[22] Alain Rakotomamonjy, Gilles Gasso, and Joseph Salmon. 2019. Screening rules for Lasso with non-convex Sparse Regularizers. In *International Conference on Machine Learning*. 5341–5350.

[23] Peter Richtárik and Martin Takáč. 2014. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming* 144, 1-2 (2014), 1–38.

[24] Shai Shalev-Shwartz and Ambuj Tewari. 2011. Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research* 12 (2011), 1865–1892.

[25] Zebang Shen, Hui Qian, Tongzhou Mu, and Chao Zhang. 2017. Accelerated doubly stochastic gradient algorithm for large-scale empirical risk minimization. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2715–2721.

[26] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2013. A sparse-group lasso. *Journal of computational and graphical statistics* 22, 2 (2013), 231–245.

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

[28] Ryan J Tibshirani et al. 2013. The lasso problem and uniqueness. *Electronic Journal of statistics* 7 (2013), 1456–1490.

[29] Huahua Wang and Arindam Banerjee. 2014. Randomized block coordinate descent for online and stochastic optimization. *arXiv preprint arXiv:1407.0107* (2014).

[30] Lin Xiao and Tong Zhang. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* 24, 4 (2014), 2057–2075.

[31] Jianhua Xu, Jiali Liu, Jing Yin, and Chengyu Sun. 2016. A multi-label feature extraction algorithm via maximizing featurxue variance and feature-label dependence simultaneously. *Knowledge-Based Systems* 98 (2016), 172–184.

[32] Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68, 1 (2006), 49–67.

[33] Tuo Zhao, Mo Yu, Yiming Wang, Raman Arora, and Han Liu. 2014. Accelerated mini-batch randomized block coordinate descent method. *Advances in neural information processing systems* 27 (2014), 3329–3337.

[34] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 2003. 1-norm support vector machines. In *Advances in neural information processing systems*. Citeseer, None.

[35] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* 67, 2 (2005), 301–320.