EFFICIENT HIGH-DIMENSIONAL DATA REPRESENTATION LEARNING VIA SEMI-STOCHASTIC BLOCK COORDINATE DESCENT METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

With the increase of data volume and data dimension, sparse representation learning attracts more and more attention. For high-dimensional data, randomized block coordinate descent methods perform well because they do not need to calculate the gradient along the whole dimension. Existing hard thresholding algorithms evaluate gradients followed by a hard thresholding operation to update the model parameter, which leads to slow convergence. To address this issue, we propose a novel hard thresholding algorithm, called Semi-stochastic Block Coordinate Descent Hard Thresholding Pursuit (SBCD-HTP). Moreover, we present its sparse and asynchronous parallel variants. We theoretically analyze the convergence properties of our algorithms, which show that they have a significantly lower hard thresholding complexity than existing algorithms. Our empirical evaluations on real-world datasets and face recognition tasks demonstrate the superior performance of our algorithms for sparsity-constrained optimization problems.

1 INTRODUCTION

In modern high-dimensional data analytics, where the variable dimension can be equal or even larger than the number of samples, sparse representation learning has become a mainstream method to explore the potential real model of the problem and provides statistically reliable results. It has been applied to many diverse domains such as high-dimensional statistics (Bühlmann & Van De Geer, 2011), signal processing (Lai et al., 2013), and computer vision (Wright et al., 2008). Many sparse learning algorithms such as $\ell_1$-norm convex relaxation methods (Negahban et al., 2009; Van de Geer et al., 2008) have been proposed in the past few decades. In order to get a smaller estimation error, $\ell_0$-norm constrained algorithms are more prominent than $\ell_1$-norm convex relaxation algorithms (Zhang et al., 2010). In this paper, we mainly focus on the following sparsity-constrained optimization problem:

$$\min_w F(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w), \quad \text{s.t., } \|w\|_0 \leq s,$$

where $F(w)$ is a finite-sum convex and smooth function, each function $f_i(w)$ is associated with the $i$-th sample, $\|w\|_0$ is the number of nonzero entries in variable $w$, and $s$ represents the sparsity level. The goal of sparse representation learning is to recover $w^*$ based on the given data. Such a formulation encapsulates plentiful important problems, including sparse graphical model learning (Zhou et al., 2019), sparse linear/logistic regression (Blumensath & Davies, 2009; Foucart, 2011; Pati et al., 1993; Bahmani et al., 2013), and low-rank regression (Rohde et al., 2011).

However, Problem (1) is in general NP hard, which is caused by the non-convexity of the sparsity constraint. It makes us to obtain an approximate solution to Problem (1). One of the most widely used methods for solving this problem is the hard thresholding based gradient descent method. In recent years, sparse representation learning is becoming more common for large-scale and high-dimensional data. However, deterministic gradient descent hard thresholding algorithms such as fast gradient descent hard thresholding (FG-HT) (Jain et al., 2014; Yuan et al., 2014) have to compute the gradients of all $n$ component functions, which leads to a huge computing overhead. To address this issue, many stochastic hard thresholding methods have been proposed. For instance, Nguyen et al. (2017) proposed a stochastic gradient descent hard thresholding (SG-HT) algorithm, whose...
gradient and hard thresholding complexities are both \(O(\kappa_2 \log(\frac{1}{\epsilon}))\). However, due to the stochastic sampling, SG-HT can only attain a sub-optimal estimation bound, as shown in Table 1 which is inferior to those of deterministic gradient methods such as FG-HT. Another limitation of SG-HT is that it requires that the restricted condition number \(\kappa_2\) should not be larger than \(4/3\), which makes SG-HT hard for solving high-dimensional representation learning problems.

Recently, many stochastic variance reduced methods (e.g., SAG [Roux et al., 2012], SVRG [Johnson & Zhang, 2013]) and their accelerated variants such as [Defazio, 2016; Allen-Zhu, 2018] have been proposed to accelerate stochastic gradient methods for convex optimization. All these methods enjoy low per-iteration complexities comparable with stochastic gradient descent (SGD), and they also can attain improved convergence rates. By incorporating the variance reduction technique into sparsity-constrained optimization domain, [Li et al., 2016b] proposed a stochastic variance reduced gradient hard thresholding (SVRG-HT) algorithm, which can converge more quickly and obtain a smaller estimation error than SG-HT. Moreover, SVRG-HT allows an arbitrarily large condition number in its theoretical analysis similar to FG-HT ([Yuan et al., 2014]). The gradient oracle and hard thresholding complexities of SVRG-HT are \(O((n + \kappa_2) \log(\frac{1}{\epsilon}))\) and \(O(\kappa_2 \log(\frac{1}{\epsilon}))\), respectively. Nevertheless, it can not leverage the coordinate block to accelerate convergence. [Chen & Gu, 2016] proposed an accelerated stochastic block coordinate gradient descent hard thresholding (ASBCDHT) algorithm. The gradient oracle complexity of ASBCDHT is \(O((n + \frac{\kappa_2 B}{k}) \log(\frac{1}{\epsilon}))\), which is superior to SVRG-HT. Moreover, ASBCDHT also has the hard thresholding complexity, \(O(\kappa_2 \log(\frac{1}{\epsilon}))\). However, the hard thresholding complexity of ASBCDHT still scales linearly with \(\kappa_2\), which is usually expensive for real-world sparse learning problems. For large-scale and high-dimensional data, [Li et al., 2016a] also proposed the asynchronous parallel variant of SVRG-HT (called ASVRG-HT) by utilizing multicore architectures. Although it makes each processor to evaluate a stochastic gradient update on a global parameter stored in a shared memory in an asynchronous and lock-free mode, ASVRG-HT attains the similar gradient and hard thresholding complexities as its general version (i.e., SVRG-HT). Therefore, all the algorithms mentioned above have a linearly \(\kappa_2\)-dependent hard thresholding complexity. This motivates us to address the following key issue:

**Can we design such algorithms that have a \(\kappa_2\)-independent hard thresholding complexity and a lower gradient oracle complexity?**

To answer the above problem, we propose an efficient Semi-stochastic Block Coordinate Descent Hard Thresholding Pursuit (SBCD-HTP) algorithm and its sparse Asynchronous variant (ASBCD-HTP). The oracle complexities and statistical estimation error of the proposed algorithms and other hard thresholding methods are summarized in Table 1. We highlight several theoretical advantages of the proposed algorithms over the state-of-the-art methods as follows:

- SBCD-HTP and ASBCD-HTP substantially improve the restricted condition on the sparsity level \(s\), i.e., \(s = \Omega(\kappa_2 s^*)\), while most related algorithms such as SVRG-HT require \(s = \Omega(\kappa_2 s^*)\).
- The statistical estimation error of both our algorithms is better than those of FG-HT, SVRG-HT and ASBCDHT, as our algorithms have a smaller sparsity level \(s = \Omega(\kappa_2 s^*)\) and thus have a smaller

<table>
<thead>
<tr>
<th>Methods</th>
<th>Required value of (s)</th>
<th>Gradient Complexity</th>
<th>Hard Thresholding Complexity</th>
<th>Statistical Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>FG-HT</td>
<td>(\Omega(\kappa_2 s^*))</td>
<td>(O((n + \kappa_2) \log(\frac{1}{\epsilon})))</td>
<td>(O(\kappa_2 \log(\frac{1}{\epsilon})))</td>
<td>(O(\sqrt{\kappa_2} \norm{\nabla f(x^<em>)} + \norm{\nabla F(x^</em>)}))</td>
</tr>
<tr>
<td>SG-HT</td>
<td>(\Omega(\kappa_2 s^*))</td>
<td>(O((n + \kappa_2) \log(\frac{1}{\epsilon})))</td>
<td>(O(\kappa_2 \log(\frac{1}{\epsilon})))</td>
<td>(O(\sqrt{n} \norm{\nabla f(x^<em>)} + \norm{\nabla F(x^</em>)}))</td>
</tr>
<tr>
<td>SVRG-HT</td>
<td>(\Omega(\kappa_2 s^*))</td>
<td>(O((n + \kappa_2) \log(\frac{1}{\epsilon})))</td>
<td>(O(\kappa_2 \log(\frac{1}{\epsilon})))</td>
<td>(O(\sqrt{n} \norm{\nabla f(x^<em>)} + \norm{\nabla F(x^</em>)}))</td>
</tr>
<tr>
<td>ASBCDHT</td>
<td>(\Omega(\kappa_2 s^*))</td>
<td>(O((n + \kappa_2) \log(\frac{1}{\epsilon})))</td>
<td>(O(\kappa_2 \log(\frac{1}{\epsilon})))</td>
<td>(O(\sqrt{n} \norm{\nabla f(x^<em>)} + \norm{\nabla F(x^</em>)}))</td>
</tr>
<tr>
<td>ASBCD-HTP</td>
<td>(\Omega(\kappa_2 s^*))</td>
<td>(O((n + \kappa_2) \log(\frac{1}{\epsilon})))</td>
<td>(O(\kappa_2 \log(\frac{1}{\epsilon})))</td>
<td>(O(\sqrt{n} \norm{\nabla f(x^<em>)} + \norm{\nabla F(x^</em>)}))</td>
</tr>
</tbody>
</table>

Table 1: Comparison of some properties of the hard thresholding algorithms for solving sparsity-constrained problems. [\(s\)] is the mini-batch size of block coordinate descent methods, and \(k\) is the number of coordinate blocks. Since the sparsity level \(s = \Omega(\kappa_2 s^*)\) required in our algorithms (i.e., SBCD-HTP and its parallel variant, ASBCD-HTP) is much smaller than those of other algorithms (e.g., \(\Omega(\kappa_2 s^*)\)), the condition number \(\kappa_2\) in our algorithms is also smaller than \(\kappa_2\) in other algorithms, where \(\tilde{s} = 2\Omega(\kappa_2 s^*) + s^*\) and \(\tilde{\kappa} = 2\Omega(\kappa_2 s^*) + s^*\).
cardinality, i.e., $\hat{s} = 2s + s^*$. This makes the restricted condition number $\tilde{\kappa}_s$ of SBCD-HTP smaller than other algorithms. Moreover, the statistical error of our algorithms is also smaller than that of SG-HT (i.e., $\frac{1}{n} \sum_{i=1}^{n} \| \nabla f_i(w^*) \|$) due to the large magnitude of $\| \nabla f_i(w^*) \|$ (Zhou et al., 2018b).

- Both SBCD-HTP and ASBCD-HTP enjoy a $\kappa_s$-independent hard thresholding complexity, which is significantly better than the state-of-the-art hard thresholding algorithms. That is, the hard thresholding complexity of our algorithms is $\kappa_s$ times lower than that of FG-HT, SG-HT, SVRG-HT and ASBCDHT with $\hat{s} = 2\Omega(\kappa_s s^*) + s^*$. Since both our algorithms have a significantly lower hard thresholding complexity, they are more suitable for handling large-scale sparse representation learning problems, especially for high-dimensional data.

- For sparsity-constrained problems, the gradient oracle complexities of both SBCD-HTP and ASBCD-HTP are much lower than those of the state-of-the-art hard thresholding algorithms. In other words, their gradient oracle complexities are significantly lower than those of ASBCDHT and SVRG-HT, and much lower than that of FG-HT, since they require a smaller sparsity level $s = \Omega(\kappa_s s^*)$ and also have a smaller value of $\kappa_s$ with $\hat{s} = 2s + s^*$ compared with the restricted condition number $\kappa_s$ used in other hard thresholding algorithms (Zhou et al., 2018b).

2 RELATED WORK

In this section, we briefly discuss the relevant research to our work, which beyonds the sparsity-constrained optimization domain. Traditional gradient descent is computational expensive at each iteration, and stochastic gradient descent has a low per-iteration complexity while obtaining a large variance in estimating. Thus, many stochastic variance reduced algorithms (Defazio et al., 2014; Roux et al., 2012; Shalev-Shwartz & Zhang, 2013; Shang et al., 2018; Johnson & Zhang, 2013) and their variants (Schmidt et al., 2017; Konečný et al., 2015; Xiao & Zhang, 2014) have been proposed. The stochastic variance reduced methods are very promising for many machine learning problems including sparse learning problems.

In contrast to gradient descent methods, coordinate descent algorithms have received increasing attention due to their successful applications in high-dimensional problems (Breheny & Huang, 2011; Friedman et al., 2007). Among them, randomized block coordinate descent (RBCD) updates a block of coordinate with respect to the entire training instances. The per-iteration cost is significantly lower than gradient descent, while it still has a relationship with $n$ component functions. Some stochastic block coordinate descent (SBCD) algorithms such as (Dang & Lan, 2015; Xu & Yin, 2015; Konečný et al., 2017) were proposed. Such algorithms compute the stochastic gradient with respect to a random sample restricted to one randomized coordinate block. Therefore, these algorithms randomly sample a block of features and data instances at each iteration. However, they can only achieve a sublinear convergence rate (Zhang & Gu, 2016). Recently, some mini-batch randomized block coordinate descent algorithms such as (Zhao et al., 2014; Wang & Banerjee, 2014) were proposed to accelerate the convergence of stochastic block coordinate gradient descent. Our work studies over the above research by considering a sparsity-constrained optimization problem. The proposed algorithms enjoy a lower computational complexity in both gradient evaluation and hard thresholding computation while obtaining a linear convergence performance. Moreover, its sparse and asynchronous variants are also proposed to deal with sparse high-dimensional data.

3 PRELIMINARIES

Throughout this paper, we use $w^*$ to denote the optimal solution of Problem (1), and its optimal sparsity level is $s^*$ with $\| w^* \|_0 \leq s^*$. $\| w \|$ is the Euclidean norm for a vector $w \in \mathbb{R}^d$ and $\| w \|_\infty$ is the largest absolute entry in $w$. The hard thresholding operation $HT(w, s)$ preserves the $s$ largest entries of $w$ in magnitude for vector $w$ and the rest entries are set to zero. We use supp$(w)$ to denote the support set of $w$, i.e., the indices of its nonzero entries. Given an index set $I$, we define $I^C$ as the complement set of $I$, and $v_T \in \mathbb{R}^d$, where $v_T^j = v_j$ if $j \in I$ and $v_T^j = 0$ if $j \notin I$. Given an integer $n \geq 1$, we define $[n] = \{1, \ldots, n\}$. For a set $B$, we denote its cardinality by $|B|$. Moreover, we use the common notations of $\Omega(\cdot)$ and $O(\cdot)$ to characterize the asymptotics of two real sequences.

Throughout the analysis, we make two important assumptions on the objective function, which are commonly used in the analysis of hard thresholding algorithms (Li et al., 2016b; Shen & Li, 2017; Chen & Gu, 2016; Gao & Huang, 2018). In high-dimensional sparse learning, the per-iteration hard
In this section, we propose a novel Semi-stochastic Block Coordinate Descent Hard Thresholding Pursuit (SBCD-HTP) algorithm, and also theoretically analyze its convergence properties.

4.1 Our SBCD-HTP Algorithm

The proposed SBCD-HTP algorithm is summarized in Algorithm 1. In each outer-loop, we select a snapshot point \( \tilde{w} \) and compute its full gradient \( \nabla F(\tilde{w}) \). In each inner-loop, we uniformly and randomly select a mini-batch samples \( B \) and a block of coordinates \( G_{j_t} \), where \( j_t \in \{1, 2, \ldots, k\} \). \( \{G_1, \ldots, G_k\} \) is a partition of all the \( d \) coordinates and is divided uniformly at random. We usually set \( k = 10 \) in our experiments. Note that, different from ASBCDHT, in line 5 we set \( S \) to be a union set of the support set of snapshot point \( \tilde{w} \) and \( G_{j_t} \). In this way, the coordinate of the support set \( S \) which includes the possibly optimal coordinates can be updated, and thus this can reduce the error caused by randomness. Moreover, SBCD-HTP enjoys the dual advantages of randomized block coordinate descent and semi-stochastic gradient descent to optimize the objective as shown in line 6. Note
that we perform the hard thresholding operation in only the outer-loop, which is different from all existing hard thresholding algorithms. Performing hard thresholding operations with high frequency or prematurely in inner-loop will make the gradient information lost, which is the main reason for slow convergence and low accuracy of existing hard thresholding sparsity-constrained algorithms. Our hard thresholding algorithm not only guarantees the sparsity of model parameters, but also uses the full dimensional information without the interference of hard thresholding operations to update model parameter \( w^{t+1} \). Our theoretical analysis and experimental results show that SBCD-HTP has a faster convergence rate and more accurate results.

### 4.2 Convergence Analysis

We first analyze the convergence behavior of SBCD-HTP. The main result is summarized in the following theorem, whose proof is provided in Section [C]

**Theorem 1** Suppose \( F(w) \) is \( \bar{s} \)-strongly convex and each function \( f_i(w) \) is \( \rho_{\bar{s}}^+ \)-strongly smooth with parameter \( \bar{s} = 2s + s^* \). Let \( \kappa_{\bar{s}} = \frac{\rho_{\bar{s}}^+}{\rho_{\bar{s}}^-} \) and the sparsity level \( s \geq \Omega(\kappa_{\bar{s}}s^*) \). In addition, assume that the learning rate \( \eta \leq \frac{1}{4\rho_{\bar{s}}^-} \), the number of inner-loops \( m \geq 1800\kappa_{\bar{s}} \), and the number of blocks \( k = 10 \), then the convergence rate \( \gamma = \frac{k}{2m\rho_{\bar{s}}^- (1-\omega)} + \frac{8\rho_{\bar{s}}^+\eta(n-|B|)}{|B|(1-\omega)(n-1)} \leq \frac{1}{2} \). For the sparsity-constrained problem \( (\mathcal{I}) \), the output of Algorithm \( \mathcal{I} \) satisfies

\[
\mathbb{E}[F(\tilde{w}^r) - F(w^*)] \leq \left( \frac{1}{2} \right)^r \mathbb{E}[F(\tilde{w}^0) - F(w^*)] + \frac{\eta}{1-\omega} \|\nabla F(w^*)\|^2,
\]

(4)

where \( \tilde{I} = \text{supp}(HT(\nabla F(w^*), 2s)) \cup \text{supp}(w^*) \) and \( \omega = 8\rho_{\bar{s}}^+\eta(1+\frac{|B|}{|B|(n-1)}) \).

**Corollary 1** Suppose the conditions in Theorem [I] hold. To achieve \( \left( \frac{1}{2} \right)^r \mathbb{E}[F(\tilde{w}^0) - F(w^*)] \leq \epsilon \), the gradient oracle complexity of SBCD-HTP is \( \mathcal{O}((n + \frac{\kappa_{\bar{s}}|B|}{k}) \log(\frac{1}{\epsilon})) \), and its hard thresholding complexity is \( \mathcal{O}(\log(\frac{1}{r})) \).

Table [I] summarizes the properties of some hard thresholding methods. Compared with the gradient oracle complexities of FG-HT [Yuan et al. 2014] and SVRG-HT [Li et al. 2016b] (i.e., \( \mathcal{O}(n\kappa_{\bar{s}} \log(\frac{1}{\epsilon})) \) and \( \mathcal{O}(n + \kappa_{\bar{s}}) \log(\frac{1}{\epsilon})) \), respectively, SBCD-HTP has a much lower cost than them. For ASBCDHT [Chen & Gu 2016], whose gradient oracle complexity is \( \mathcal{O}((n + \frac{\kappa_{\bar{s}}|B|}{k}) \log(\frac{1}{\epsilon})) \), the gradient oracle complexity of SBCD-HTP is \( \mathcal{O}((n + \frac{\kappa_{\bar{s}}|B|}{k}) \log(\frac{1}{\epsilon})) \) and is much lower than ASBCDHT, as \( \kappa_{\bar{s}} \) of SBCD-HTP is usually comparable to or even smaller than those of others (note that \( s \) in \( s = 2s + s^* \) is required to be smaller than those as shown below). Moreover, SBCD-HTP allows \( s = \Omega(\kappa_{\bar{s}}s^*) \), which is considerably superior to the condition of \( s = \Omega(\kappa_{\bar{s}}s^*) \) required in other hard thresholding algorithms such as [Zhou et al. 2018b]. In particular, with the same parameter settings of \( |B| \) and \( k \), the gradient oracle complexity of our algorithm outperforms ASBCDHT. SBCD-HTP also attains a lower hard thresholding complexity than the state-of-the-art hard thresholding methods. That is, the hard thresholding complexity of SBCD-HTP is \( \mathcal{O}(\log(\frac{1}{r})) \), which is \( \kappa_{\bar{s}} \)-independent and is significantly lower than those of other methods. This is a significant improvement on hard thresholding complexity. Theorem [I] immediately implies the following results.

**Corollary 2** Suppose the conditions in Theorem [I] hold. The output of Algorithm \( \mathcal{I} \) satisfies

\[
\mathbb{E}[\|\tilde{w}^r - w^*\|] \leq \sqrt{\frac{2\left(\frac{1}{2}\right)^r \mathbb{E}[F(\tilde{w}^0) - F(w^*)]}{\rho_{\bar{s}}^-}} + \left( \frac{2}{\rho_{\bar{s}}^-} + \frac{2\eta}{1-\omega} \right) \sqrt{\kappa_{\bar{s}} \|\nabla F(w^*)\|\infty}.
\]

(5)

The right-hand side of Eq. (5) consists of two terms. The first term is the optimization error, which approaches zero with the increase of \( r \). The second term corresponds to the statistical error, which is proportional to \( \sqrt{\kappa_{\bar{s}} \|\nabla F(w^*)\|\infty} \). One can also observe that the statistical error bound is better
Algorithm 2: Asynchronous Sparse SBCD-HTP (ASBCD-HTP)

**Input:** The number of outer-loops $R$, number of inner-loops $m$, step size $\eta$, sparsity level $s$.

**Initialize:** $\bar{w}^0$.

1: for $r = 0, 1, \ldots, R - 1$ do
2:   $w^0 = \bar{w} = \bar{w}^r$, $\tilde{G} = \text{supp}(\bar{w})$, and $\nabla F(\bar{w}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\bar{w})$; \text{//computed in parallel}
3:   $t = 0$; \text{//inner loop counter}
4:   compute the while loop in parallel
5:   while $t < m$ do
6:     Randomly sample $i_t$ from $[n]$ uniformly, and randomly sample $j_t$ from $[k]$ uniformly;
7:     $T_{i_t} := \text{support of sample } i_t$, and $t = t + 1$; \text{//atomic increase counter $t$}
8:     $S = \tilde{G} \cup \tilde{G}_{j_t}$, $[\bar{w}]_{T_{i_t}} := \text{inconsistent read of shared variable } [\bar{w}]_{T_{i_t}}$;
9:     $\nabla_S g([\bar{w}]_{T_{i_t}}) = \nabla_S f_{i_t}([\bar{w}]_{T_{i_t}}) - \nabla_S f_{i_t}(\bar{w}) + D_{i_t} \nabla_S F(\bar{w})$;
10: $[\bar{w}]^r := [\bar{w}]^r \cap T_{i_t} = [\bar{w}]^r \cap T_{i_t} - \eta \nabla_S g([\bar{w}]_{T_{i_t}})$; \text{//atomic write}
11: end while
12: $\bar{w}^{r+1} = HT(\bar{w}^r, s)$;
13: end for

**Output:** $\bar{w}^R$.

than those of FG-HT and ASBCD-HT, since $\hat{s}$ in the statistical error $O(\sqrt{\hat{s}}\|\nabla F(w^*)\|_\infty)$ has a smaller cardinality (i.e., $2\Omega(\kappa\hat{s}^2s^*) + s^2$), while existing algorithms require a larger sparsity level $s^* = \Omega(\kappa^2\hat{s}^2s^*)$ and have a larger cardinality. Therefore, compared with other existing algorithms, SBCD-HTP enjoys a smaller statistical error, especially for the problems with a large restricted condition number. It is usually better than the error bound $O(\sqrt{\hat{s}}\|\nabla F(w^*)\|_\infty + \|\nabla x F(w^*)\|)$ with $\hat{s} = 2\Omega(\kappa^2\hat{s}^2s^*) + s^2$ in SVRG-HT. Moreover, the error bound of SBCD-HTP is much smaller than that of SG-HT (Nguyen et al. 2017) (i.e., $O(\frac{1}{\sqrt{\hat{s}}} \sum_{i=1}^{n} ||\nabla f_i(\hat{s}^*)||)$), since the magnitude of the individual gradient norm $||\nabla f_i(\hat{s}^*)||$ can still be relatively large (Zhou et al. 2016).

5 Asynchronous Variant of SBCD-HTP

In this section, we propose the serial sparse and asynchronous parallel variants of SBCD-HTP for high-dimensional sparse data sets. Our Asynchronous Sparse Semi-stochastic Block Coordinate Descent Hard Thresholding Pursuit (ASBCD-HTP) algorithm is summarized in Algorithm 2. The main difference between ASBCD-HTP and SBCD-HTP is sparse approximate gradients as in (Maia et al. 2017) and (Zhou et al. 2018). In order to perform fully sparse updates, we use a diagonal matrix $D$ to re-weight the dense fully gradient $\nabla F(\bar{w})$. The entries of $D$ are the inverse probabilities of the corresponding coordinates belonging to a uniformly sampled support $T_{i_t}$ of sample $i_t$. Let $P_{i_t}$ be the projection matrix for the support $T_{i_t}$, we can get $D_{i_t} := P_{i_t} D$. Then we can find that $\mathbb{E}_i[D_{i_t} \nabla F(\bar{w})] = \nabla F(\bar{w})$. In this setting, we can compute the full gradient $\nabla F(\bar{w})$ also in a parallel way. In order to reduce thread interference and fully utilize the sparsity of datasets, we restrict the sparse gradient $\nabla_S g([\bar{w}]_{T_{i_t}})$ to the intersection set of $S$ and $T_{i_t}$, and obtain $\nabla_S g([\bar{w}]_{T_{i_t}})$. Specifically, each thread of ASBCD-HTP computes sparse approximate gradient independently and then atomically write this gradient to the shared variable $\bar{w}^r$. Note that Sparse variant of SBCD-HTP (S$^2$BCD-HTP) can be viewed as the single-thread version of ASBCD-HTP, which has linear convergence and the same oracle complexity as SBCD-HTP. The detailed description and analysis of S$^2$BCD-HTP is listed in Section D. Next we give the main theoretical result of ASBCD-HTP.

**Theorem 2** Suppose $F(w)$ is $\rho^-\kappa_-$-strongly convex and each function $f_i(w)$ is $\rho_i^+\kappa_i^+$-smooth with parameter $\hat{s} = 2s + s^*$. Let $\kappa_\hat{s} = \frac{s^*}{\kappa_\hat{s}}$ and the sparsity level $s \geq \Omega(\kappa_\hat{s}s^*)$. We assume that the step size $\eta = \frac{1}{8\rho_\kappa^+\kappa_\hat{s}^2}$, the number of inner-loops $m \geq 120\kappa_\hat{s}$, the number of blocks $k = 10$ and $\tau \leq \min\left\{\frac{3}{5\Delta}, 2\kappa_\hat{s}, \sqrt{\frac{2s^2}{\Delta}}\right\}$ (i.e., the linear speedup condition). Then for the sparsity-constrained problem (1), the gradient oracle and hard thresholding complexities of Algorithm 2 are

\[
O\left((n + \kappa_\hat{s})k \log(1/\epsilon)\right) \quad \text{and} \quad O\left((\log(1/\epsilon))\right),
\]

where $\tau$ denotes the maximum number of concurrent threads (Mania et al. 2017) and $\Delta = \max_{j=1\ldots d} \rho_j$, which is an indicator to measure the sparsity of datasets (Lebeld et al. 2017).
In this section, we evaluate the performance of SBCD-HTP, S^2BCD-HTP and ASBCD-HTP for solving sparse linear regression and sparse logistic regression on real-world datasets. All the dense algorithms were implemented in MATLAB, while the sparse and asynchronous algorithms were implemented in C++ and executed through MATLAB interface for a fair comparison. We also test the performance for face recognition tasks. The detailed information is described in Section F. Figures 1 and 2 show that for both sparse linear regression and sparse logistic regression problems, our algorithm converges significantly faster than the baseline algorithms in terms of both effective passes and CPU time. The main reason is that we use all the information of gradients, while the related algorithms use less information caused by too frequent and premature hard thresholding operation in each inner-loop.

6 EXPERIMENTS

In this section, we evaluate the performance of SBCD-HTP, S^2BCD-HTP and ASBCD-HTP for solving sparse linear regression and sparse logistic regression on real-world datasets. All the dense algorithms were implemented in MATLAB, while the sparse and asynchronous algorithms were implemented in C++ and executed through MATLAB interface for a fair comparison. We also test the performance for face recognition tasks. The detailed information is described in Section F. We set the sparsity level $s = 200$ for all real-world datasets. For the number of inner-loops, we set $m = 2n$ for SBCD-HTP and ASBCDHT, $m = n$ for SVRG-HT as suggested in (Li et al., 2016b; Chen & Gu, 2016) and 1000 (sparse linear regression)/3000 (sparse logistic regression) for FNHTP. We also set the batch-size of FNHTP to $s$, and that of ASBCDHT and SBCD-HTP to 5. For the block coordinate descent methods, we set the number of coordinate blocks $k$ to 10. All the algorithms are tuned to their best performance.

We report the performance of all the algorithms in terms of both effective passes and CPU time. More experimental results are presented in Section F. Figures 1 and 2 show that for both sparse linear regression and sparse logistic regression problems, our algorithm converges significantly faster than the baseline algorithms in terms of both effective passes and CPU time. The main reason is that we use all the information of gradients, while the related algorithms use less information caused by too frequent and premature hard thresholding operation in each inner-loop.
sparse representation learning because it can be easily extended to other sparse learning algorithms.

In this paper, we proposed an efficient semi-stochastic block coordinate descent hard thresholding pursuit (SBCD-HTP) method for sparse representation learning. We proved that SBCD-HTP attains the gradient oracle complexity of $O\left((n + \frac{\alpha}{\nu} |\mathcal{B}|) \log(\frac{1}{\epsilon})\right)$, and the hard thresholding complexity of $O(\log(\frac{1}{\epsilon}))$, respectively, which is an improvement over the existing algorithms. Moreover, we also presented the sparse and asynchronous parallel variants of SBCD-HTP. As far as we know, our algorithms are the first to use the full information of gradient to update. This is an improvement on sparse representation learning because it can be easily extended to other sparse learning algorithms.

6.2 Asynchronous Algorithms

We ran the experiments of all the asynchronous algorithms on a PC with an Intel Xeon(R) Gold 5120 CPU and 64GB RAM. We compare our sparse and asynchronous algorithms with several state-of-the-art algorithms, including ASG-HT, ASVRG-HT and $\hat{\mathcal{S}}^2$BCD-HTP (asynchronous variant of ABSCDHT). For the algorithms, we set the number of threads to 20.

We only report the performance of all the algorithms in terms of CPU time since the performance in terms of effective passes is similar to the dense case. More experimental results are reported in Section F. Figure 3 shows that ASBCD-HTP and $\hat{\mathcal{S}}^2$BCD-HTP significantly outperform than other algorithms. Moreover, ASBCD-HTP has a faster convergence performance than $\hat{\mathcal{S}}^2$BCD-HTP because of asynchronous parallel acceleration. There are three main reasons for the advantages of our ASBCD-HTP: a) We get the set $\hat{\mathcal{S}}$ from the support set of snapshot point $\hat{\omega}$ and $\mathcal{G}_{\hat{\omega}}$. In this way, the coordinate of the support set which appears as the possible optimal solution can be updated, and thus reducing the error caused by randomness. b) Because we put the hard thresholding operation in each outer-loop, the computation cost per thread of our algorithm at each inner-loop is less, while each thread of the baseline algorithms has a hard thresholding operation, which is expensive especially for high-dimensional datasets. c) Our algorithm updates $\hat{\omega}$ with all information of gradients, while the baseline algorithms update $\hat{\omega}$ with less information caused by too frequent and premature hard thresholding operation in each inner-loop. We also evaluate the improvement of asynchronous parallel by running the same passes with different numbers of threads. We calculate the speed-up ratio based on the running time of a single thread. From Figure 4, we can see that our ASBCD-HTP is accelerated by a nearly linear ratio.

7 Conclusion

In this paper, we proposed an efficient semi-stochastic block coordinate descent hard thresholding pursuit (SBCD-HTP) method for sparse representation learning. We proved that SBCD-HTP attains the gradient oracle complexity of $O\left((n + \frac{\alpha}{\nu} |\mathcal{B}|) \log(\frac{1}{\epsilon})\right)$, and the hard thresholding complexity of $O(\log(\frac{1}{\epsilon}))$, respectively, which is an improvement over the existing algorithms. Moreover, we also presented the sparse and asynchronous parallel variants of SBCD-HTP. As far as we know, our algorithms are the first to use the full information of gradient to update. This is an improvement on sparse representation learning because it can be easily extended to other sparse learning algorithms.
REFERENCES


APPENDIX

A KEY LEMMAS

Lemma 1 (Jain et al. [2014]) For any index set $I$, any $w \in \mathbb{R}^I$ and $HT(\cdot, s) : \mathbb{R}^d \to \mathbb{R}^d$ be the hard thresholding operator, which keeps the largest $s$ entries (in magnitude) and sets the other entries equal to zero. Then for any $w^* \in \mathbb{R}^I$ such that $\|w^*\|_0 \leq s^*$, we have

$$\|HT(w, s) - w\|^2 \leq \frac{|I| - s}{|I| - s^*} \|w - w^*\|^2.$$  

Lemma 2 (Li et al. [2016b]) Let $w^* \in \mathbb{R}^d$ be the optimal sparse vector such that $\|w^*\|_0 \leq s^*$ and $HT(\cdot, s) : \mathbb{R}^d \to \mathbb{R}^d$ be the hard thresholding operator, which keeps the largest $s$ entries (in magnitude) and sets the other entries equal to zero. Given $s > s^*$ for any vector $w \in \mathbb{R}^d$, we have

$$\|HT(w, s) - w^*\|^2 \leq (1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}) \|w - w^*\|^2.$$  

Lemma 3 (Nesterov [2004]) For any given vector $w, w' \in \mathbb{R}^d$, suppose $F(\cdot)$ also satisfies RSC and RSS conditions, then the following inequality holds

$$\|\nabla F(w) - \nabla F(w')\|^2 \leq 2\rho_F^+[F(w) - F(w') + \langle \nabla F(w'), w - w' \rangle].$$  

Lemma 4 Suppose $F(w)$ is $\rho_F^-$-strongly convex and each function $f_i(w)$ is $\rho_i^+$-strongly smooth with parameter $\bar{s} = 2s + s^*$. $\mathcal{I} = \text{supp}(w^*) \cup \text{supp}(\bar{w}) \cup \text{supp}((\bar{w})^+)$, $|\mathcal{B}|$ is the batch size. For any $w' \in \mathbb{R}^d$ and the sample $i_t \in \mathcal{B}$, denote $\nu_i = (\nabla_{\mathcal{I}} f_i(w) - \nabla_{\mathcal{I}} f_i(w^*)) - (\nabla_{\mathcal{I}} F(w) - \nabla_{\mathcal{I}} F(w^*))$, we have

$$\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nu_i \right\|^2 = \frac{n - |\mathcal{B}|}{|\mathcal{B}|(n - 1)} \mathbb{E} \|\nabla_{\mathcal{I}} f_i(w') - \nabla_{\mathcal{I}} f_i(w^*) + \nabla_{\mathcal{I}} F(w^*) - \nabla_{\mathcal{I}} F(w')\|^2.$$  

Proof. See Section B.1 for the proof of Lemma 4.  

Lemma 5 Suppose $F(w)$ is $\rho_F^-$-strongly convex and each function $f_i(w)$ is $\rho_i^+$-strongly smooth with parameter $\bar{s} = 2s + s^*$. Let $w^* \in \mathbb{R}^d$ be the optimal sparse vector with $\|w^*\|_0 \leq s^*$. $\mathcal{I} = \text{supp}(w^*) \cup \text{supp}(\bar{w}^+) \cup \text{supp}((\bar{w})^+)$, $\delta > 1$ is a uniform constant factor, and $|\mathcal{B}|$ is the batch size. For any $w', \bar{w} \in \mathbb{R}^d$ and the sample $i_t$, denote $\nabla_S g(w^t) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_S f_i(w') + \nabla_S F(\bar{w})$, then we can bound $\mathbb{E}\|\nabla_S g(w^t)\|^2$ as follows:

$$\mathbb{E}\|\nabla_S g(w^t)\|^2 \leq \frac{1}{\kappa} (16\delta \rho_F^+ (1 + \frac{n - |\mathcal{B}|}{|\mathcal{B}|(n - 1)}) \mathbb{E}|F(w') - F(w^*)| + 4\mathbb{E}\|\nabla_{\mathcal{I}} F(w^*)\|^2 + 16\delta \rho_F^+ \frac{n - |\mathcal{B}|}{|\mathcal{B}|(n - 1)} \mathbb{E}|F(\bar{w}) - F(w^*)|).$$  

Proof. See Section B.2 for the detailed proof of Lemma 5.  

Lemma 6 Suppose $F(w)$ is $\rho_F^-$-strongly convex and each function $f_i(w)$ is $\rho_i^+$-strongly smooth with parameter $\bar{s} = 2s + s^*$. Let $w^* \in \mathbb{R}^d$ be the optimal sparse vector with $\|w^*\|_0 \leq s^*$. $\mathcal{I} = \text{supp}(w^*) \cup \text{supp}(\bar{w}^+) \cup \text{supp}((\bar{w})^+)$, $\delta > 1$ is a uniform constant factor, and $D_m = \max_{j=1, \ldots, d} \frac{1}{p_j}$. For any $w', \bar{w} \in \mathbb{R}^d$ and the sample $i_t$, denote $\nabla_S g(w^t) = \nabla_S f_i(w') + D_i\nabla_S F(\bar{w})$, we can bound $\mathbb{E}\|\nabla_S g(w^t)\|^2$ as follows:

$$\mathbb{E}\|\nabla_S g(w^t)\|^2 \leq \frac{1}{\kappa} (12\delta \rho_F^+ |F(w') - F(w^*)| + (3 + 12(D_m^2 - 1)) \mathbb{E}\|\nabla_{\mathcal{I}} F(w^*)\|^2 + (24\delta \rho_F^+ + 48\rho_F^+ (D_m^2 - 1)) |F(\bar{w}) - F(w^*)|),$$  

where $p_j$ is the probability that dimension $j$ belonging to the support set of randomly sampling sample $i_t$.  

12
Proof. See Section B.3 for the proof of Lemma 6 □

This lemma is different from other variance reduced results for sparsity-constrained problem (Mania et al., 2017; Johnson & Zhang, 2013; Chen & Gu, 2016; Shang et al., 2018). Since this variance reduced approximate gradients introduce an approximate full gradient to make fully sparse update in cardinality constraint problem.

B PROOFS OF KEY LEMMAS

In this section, we prove three key lemmas.

B.1 PROOF OF LEMMA 4

Proof. Let \( \nu_i = (\nabla_I f_i(w^t) - \nabla_I f_i(w^*)) - (\nabla_I F(w^t) - \nabla_I F(w^*)) \). We have

\[
E \left[ \left\| \frac{1}{|B|} \sum_{i \in B} \nu_i \right\|^2 \right] = \frac{1}{|B|^2} E \sum_{i, i' \in B} \nu_i^T \nu_i' + \frac{1}{|B|} E \| \nu_i \|^2
\]

\[
= \frac{|B| - 1}{|B| n(n-1)} \sum_{i \neq i'} \nu_i^T \nu_i' + \frac{1}{|B|} E \| \nu_i \|^2 \tag{11}
\]

This proof is similar to that of (Zheng & Kwok, 2016). □

B.2 PROOF OF LEMMA 5

Proof. It is obvious that the stochastic variance reduced gradient satisfies

\[
E[\nabla g(w^t)] = E \left[ \frac{1}{|B|} \sum_{i \in B} (\nabla f_i(w^t) - \nabla f_i(w)) + \mu \right] = \nabla F(w^t). \tag{12}
\]

Thus, \( \nabla g(w^t) \) is an unbiased estimator of the full gradient \( \nabla F(w^t) \). Now, we bound \( E \| \nabla g_Z(w^t) \|^2 \).

For any \( i \in \{1, 2, \cdots, n\} \), we define the following function

\[
h_i(w) = f_i(w) - f_i(w^*) - \langle \nabla f_i(w^*), w - w^* \rangle. \tag{13}
\]

It is easy to find that \( \nabla h_i(w^*) = 0 \), which implies that \( h_i(w^*) = \min_w h_i(w) \). For any vector \( w \), we have

\[
0 = h_i(w^*) \leq \min_{\eta} h_i(w - \eta \nabla h_i(w))
\]

\[
\leq \min_{\eta} h_i(w) - \eta \langle \nabla h_i(w), \nabla h_i(w) \rangle + \frac{\eta^2 \rho_i^+}{2} \| \nabla h_i(w) \|^2
\]

\[
= \min_{\eta} h_i(w) - \eta \| \nabla h_i(w) \|^2 + \frac{\eta^2 \rho_i^+}{2} \| \nabla h_i(w) \|^2 \tag{14}
\]

\[
= h_i(w) - \frac{1}{2 \rho_i^+} \| \nabla h_i(w) \|^2,
\]

where the second inequality follows from the RSS condition, and the second equality holds due to the fact that \( \langle \nabla h_i(w), \nabla h_i(w) \rangle = \| \nabla h_i(w) \|^2 \), and the last equality holds due to the fact that \( \eta = \frac{1}{\rho_i^+} \) minimizes the function. Then, we have

\[
\| \nabla_I f_i(w) - \nabla_I f_i(w^*) \|^2 \leq 2 \rho_i^+ [f_i(w) - f_i(w^*) - \langle \nabla_I f_i(w^*), w - w^* \rangle]. \tag{15}
\]
Since the sampling $i$ is chosen uniformly from $\{1, 2, \cdots, n\}$, we have
\[
\mathbb{E}\|\nabla \mathcal{I} f_i(w) - \nabla \mathcal{I} f_i(w^*)\|^2
= \mathbb{E}\|\nabla \mathcal{I} F(w) - \nabla \mathcal{I} F(w^*)\|^2
= \frac{1}{n} \sum_{i=1}^{n} \|\nabla \mathcal{I} f_i(w) - \nabla \mathcal{I} f_i(w^*)\|^2
\leq 2\rho_s^+ [F(w) - F(w^*) - \langle \nabla \mathcal{I} F(w^*), w - w^* \rangle]
\leq 2\rho_s^+ [F(w) - F(w^*) + \|\nabla \mathcal{I} F(w^*), w - w^*\|]
\leq 4\delta \rho_s^+ [F(w) - F(w^*)],
\]
where the last inequality follows from the restricted strong convexity of $F(w)$. And we use a simple underlying consensus that there must exist a uniform factor $\delta > 1$ which makes the following inequality true:
\[
\delta(F(w) - F(w^*)) \geq \|\langle \nabla \mathcal{I} F(w^*), w - w^* \rangle\| + \frac{\rho_s^+}{2} \|w - w^*\|^2.
\]
This property is called the extended restricted strong convexity used in the following proof. Therefore, we have
\[
\mathbb{E}\|\nabla \mathcal{I} f_i(w^t)\|^2 = \mathbb{E}\left\| \frac{1}{|B|} \sum_{i \in B} (\nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} f_i(w)) + \nabla \mathcal{I} F(w) \right\|^2
= \mathbb{E}\left( \left\| \frac{1}{|B|} \sum_{i \in B} \nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} f_i(w^*) + \nabla \mathcal{I} F(w^*) \right\|^2
+ \nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} F(w^*) + \nabla \mathcal{I} F(w^*)
- \left( \frac{1}{|B|} \sum_{i \in B} \nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} f_i(w^*) + \nabla \mathcal{I} F(w^*) \right) \right\|^2
\leq 4\mathbb{E}\left\| \frac{1}{|B|} \sum_{i \in B} \nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} f_i(w^*) + \nabla \mathcal{I} F(w^*) \right\|^2
+ 4\mathbb{E}\|\nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} F(w^*)\|^2 + 4\|\nabla \mathcal{I} F(w^*)\|^2
= 4 \frac{n - |B|}{|B|(n - 1)} \mathbb{E}\|\nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} f_i(w^*) + \nabla \mathcal{I} F(w^*)\|^2
+ 4 \frac{n - |B|}{|B|(n - 1)} \mathbb{E}\|\nabla \mathcal{I} f_i(w) - \nabla \mathcal{I} F(w) - \nabla \mathcal{I} f_i(w^*) + \nabla \mathcal{I} F(w^*)\|^2
+ 4\mathbb{E}\|\nabla \mathcal{I} F(w^t) - \nabla \mathcal{I} F(w^*)\|^2 + 4\|\nabla \mathcal{I} F(w^*)\|^2
\leq (4 + 4 \frac{n - |B|}{|B|(n - 1)}) \mathbb{E}\|\nabla \mathcal{I} f_i(w^t) - \nabla \mathcal{I} f_i(w^*)\|^2 + 4\|\nabla \mathcal{I} F(w^*)\|^2
+ 4 \frac{n - |B|}{|B|(n - 1)} \mathbb{E}\|\nabla \mathcal{I} f_i(w) - \nabla \mathcal{I} f_i(w^*)\|^2
\leq 16\delta \rho_s^+ (1 + \frac{n - |B|}{|B|(n - 1)}) \mathbb{E}[F(w^t) - F(w^*)] + 4\|\nabla \mathcal{I} F(w^*)\|^2
+ 16\delta \rho_s^+ \frac{n - |B|}{|B|(n - 1)} \mathbb{E}[F(w) - F(w^*)]
\]
and the last inequality follows from \cite{16}. Due to the fact that \( \mathbb{E} \| \nabla_S g^I_{\mathcal{S}}(w^t) \|^2 \leq 1/k \mathbb{E} \| \nabla g^I_{\mathcal{I}}(w^t) \|^2 \) where \( \mathcal{I} = \text{supp}(w^t) \cup \text{supp}(\tilde{w}^{t+1}) \cup \text{supp}(\tilde{\omega}) \) and \( \mathcal{S} = \mathcal{G}_i \cup \text{supp}(\tilde{\omega}) \), we have

\[
\mathbb{E} \| \nabla_S g^I_{\mathcal{S}}(w^t) \|^2 \leq \frac{1}{k} \left( 16\delta \rho_s^+ \left( 1 + \frac{n - |B|}{|B|(n-1)} \right) \mathbb{E} [ F(w^t) - F(w^*) ] + 4 \mathbb{E} \| \nabla F(w^*) \|^2 \right) + 16\delta \rho_s^+ \frac{n - |B|}{|B|(n-1)} \mathbb{E} [ F(\tilde{w}) - F(w^*) ] \tag{18}
\]

where \( k \) is the number of blocks. Note that we take expectation with respect to randomized block in the above equation. This completes the proof. \hfill \Box

\section{Proof of Lemma \ref{lem:G}}

\begin{proof}
By the definition \( \nabla g(w^t) = \nabla f_i(w^t) - \nabla f_i(\tilde{w}) + D_i \nabla F(\tilde{w}) \), we have \( \mathbb{E} [ D_i \nabla F(\tilde{w}) ] = \nabla F(\tilde{w}) \). Therefore, \( \nabla g(w^t) \) is an unbiased estimator of \( \nabla F(\tilde{w}) \).

Based on \cite{16}, we have the following result:

\[
\mathbb{E} \| \nabla g^I_{\mathcal{S}}(w^t) \|^2 = \mathbb{E} \| \nabla I f_i(w^t) - \nabla I f_i(\tilde{w}) + D_i \nabla F(\tilde{w}) \|^2 \\
\leq 3 \mathbb{E} \| D_i \nabla F(\tilde{w}) - \nabla I f_i(\tilde{w}) \|^2 + 3 \mathbb{E} \| \nabla I f_i(w^t) - \nabla I f_i(w^*) \|^2 + 3 \mathbb{E} \| \nabla F(\tilde{w}) - \nabla F(w^*) \|^2 \\
\leq 12\delta \rho_s^+ [ F(w^t) - F(w^*) ] + (3 + 12(D^2_m - 1)) \mathbb{E} \| \nabla F(w^*) \|^2 + (24\delta \rho_s^+ + 48\delta \rho_s^+(D^2_m - 1)) [ F(\tilde{w}) - F(w^*) ] \tag{19}
\]

where the first inequality follows from \( ||a + b + c||^2 \leq 3||a||^2 + 3||b||^2 + 3||c||^2 \), the second inequality follows from \( ||a + b||^2 \leq 2||a||^2 + 2||b||^2 \), the third inequality follows from \( \mathbb{E} [ w - \mathbb{E} w ]^2 \leq \mathbb{E} ||w||^2 \) with \( \mathbb{E} [ \nabla I f_i(\tilde{w}) - \nabla I f_i(w^*) ] = \nabla I F(\tilde{w}) - \nabla I F(w^*) \), and \( \mathbb{E} [ \nabla I f_i(w^t) - \nabla I f_i(w^*) ]^2 \leq 4\delta \rho_s^+[ F(w^t) - F(w^*) ] \) with \( \delta > 1 \), which is also satisfied for \( \tilde{w} \) with \( \delta > 1 \). Since \( D_m = \max_{v=1,\ldots,d} \frac{1}{p_v} \), we bound the \( \| \nabla I F(\tilde{w}) - D \nabla \tilde{I} F(\tilde{w}) \|^2 \) term as follows:

\[
\| (D - I) \nabla I F(\tilde{w}) \|^2 \\
= \sum_{v=1}^d \left( \frac{1}{p_v} - 1 \right) \| \nabla I F(\tilde{w}) \|^2_v \\
\leq (D^2_m - 1) \| \nabla I F(\tilde{w}) \|^2 \\n= (D^2_m - 1) \| \nabla I F(\tilde{w}) - \nabla I F(w^*) + \nabla I F(w^*) \|^2 \\
\leq 2(D^2_m - 1) \| \nabla I F(\tilde{w}) - \nabla I F(w^*) \|^2 + 2(D^2_m - 1) \| \nabla I F(w^*) \|^2 \\
\leq 8\delta \rho_s^+ (D^2_m - 1) [ F(\tilde{w}) - F(w^*) ] + 2(D^2_m - 1) \| \nabla I F(w^*) \|^2.
\]

Due to the fact that \( \mathbb{E} \| \nabla_S g^I_{\mathcal{S}}(w^t) \|^2 \leq \frac{1}{k} \mathbb{E} \| \nabla g^I_{\mathcal{I}}(w^t) \|^2 \), we have

\[
\mathbb{E} \| \nabla_S g^I_{\mathcal{S}}(w^t) \|^2 \leq \frac{1}{k} \left( 12\delta \rho_s^+ [ F(w^t) - F(w^*) ] + (3 + 12(D^2_m - 1)) \| \nabla I F(w^*) \|^2 \right) + (24\delta \rho_s^+ + 48\delta \rho_s^+(D^2_m - 1)) [ F(\tilde{w}) - F(w^*) ] \tag{21}
\]

where \( k \) is the number of blocks. Note that we take expectation with respect to randomized block in the above equation. This completes the proof. \hfill \Box

\section{Proofs of Theorem \ref{thm:1} and Corollaries \ref{cor:1} and \ref{cor:2}}

In this section, we give the detailed proofs of Theorem \ref{thm:1} and Corollaries \ref{cor:1} and \ref{cor:2} which can guarantee the convergence properties of our SBCD-HTP.
C.1 Proof of Theorem [1]

Proof. In this subsection, we provide the proof of Theorem 1. Let \( w^{t+1} = w^t - \eta \nabla g_x(w^t) \) and \( I = I^* \cup I^{*+1} \), where \( I^* = \text{supp}(w^*), I^{*+1} = \text{supp}(\tilde{w}^{*+1}) \) and \( I^{*+1} = \text{supp}(\tilde{w}^{*+1}) \). Conditioning on \( w^t \), we have the following expectation

\[
E\|w^{t+1} - w^*\|^2
\]

\[
= E\|w^t - \eta \nabla g_x(w^t) - w^*\|^2
\]

\[
= E\|w^t - w^*\|^2 + \eta^2 E\|\nabla g_x(w^t)\|^2 - 2\eta \langle w^t - w^*, \nabla g_x(w^t) \rangle
\]

\[
\leq E\|w^t - w^*\|^2 + \eta^2 E\|\nabla g_x(w^t)\|^2 + \frac{2\eta^2}{k} \langle w^*-w^t, \nabla F(w^t) \rangle
\]

\[
\leq E\|w^t - w^*\|^2 + \eta^2 E\|\nabla g_x(w^t)\|^2 + \frac{2\eta^2}{k} [F(w^*) - F(w^t)]
\]

\[
\leq E\|w^t - w^*\|^2 + \frac{1}{k} (16\rho_s^+ \eta^2 (1 + \frac{n - |B|}{|B|(n-1)}) E[F(w^t) - F(w^*)])
\]

\[
+ 16 \rho_s^+ \eta^2 \frac{n - |B|}{|B|(n-1)} E[F(\tilde{w}) - F(w^*)] - 2\eta \delta E[F(w^t) - F(w^*)] + \frac{4\eta^2}{k} \|\nabla F(w^*)\|^2
\]

where the first inequality follows from the expectation with respect to randomized block, the second inequality follows from our extended restricted strong convexity in the proof of Lemma 5, and the third inequality holds by Lemma 5. Notice that \( \tilde{w} = w^0 = \tilde{w}^{*+1} \). Summing over \( t = 0, \ldots, m - 1 \) and taking expectation with respect to all randomness, we have

\[
E\|w^m - w^*\|^2 \leq E\|\tilde{w}^{*+1} - w^*\|^2 + \frac{1}{k} (16\rho_s^+ \eta^2 m (1 + \frac{n - |B|}{|B|(n-1)}) - 2mn\delta) E[F(w^m) - F(w^*)]
\]

\[
+ 16 \rho_s^+ \eta^2 m \frac{n - |B|}{|B|(n-1)} E[F(\tilde{w}^{*+1}) - F(w^*)] + \frac{4\eta^2 m}{k} \|\nabla F(w^*)\|^2
\]

(23)

Notice that the convergence property used here is similar to the proofs of the convergence results for many hard thresholding algorithms [Chen & Gu 2016, 2017; Li et al., 2016b].

Moreover, we can obtain

\[
E\|w^m - w^*\|^2 \leq E\|\tilde{w}^{*+1} - w^*\|^2 + \frac{1}{k} (16\rho_s^+ \eta^2 m (1 + \frac{n - |B|}{|B|(n-1)}) - 2mn\delta) E[F(w^m) - F(w^*)]
\]

\[
+ 16 \rho_s^+ \eta^2 m \frac{n - |B|}{|B|(n-1)} E[F(\tilde{w}^{*+1}) - F(w^*)] + \frac{4\eta^2 m}{k} \|\nabla F(w^*)\|^2
\]

\[
= E\|\tilde{w}^{*+1} - w^*\|^2 + \frac{1}{k} (16\rho_s^+ \eta^2 m (1 + \frac{n - |B|}{|B|(n-1)}) - 2mn\delta) E[F(\tilde{w}^{*+1}) - F(w^m)]
\]

\[
+ 16 \rho_s^+ \eta^2 m \frac{n - |B|}{|B|(n-1)} E[F(\tilde{w}^{*+1}) - F(w^*)] + \frac{4\eta^2 m}{k} \|\nabla F(w^*)\|^2
\]

\[
\leq E\|\tilde{w}^{*+1} - w^*\|^2 + \frac{1}{k} (16\rho_s^+ \eta^2 m (1 + \frac{n - |B|}{|B|(n-1)}) - 2mn\delta) E[F(\tilde{w}^{*+1}) - F(w^{*+1})]
\]

\[
+ 16 \rho_s^+ \eta^2 m \frac{n - |B|}{|B|(n-1)} E[F(\tilde{w}^{*+1}) - F(w^*)] + \frac{4\eta^2 m}{k} \|\nabla F(w^*)\|^2
\]

(24)
where the second inequality holds due to the fact that we have the following results: When $F(\tilde{w}^r) - F(w^m) > 0$, by using Lemma 3 on $w^m$ and $\tilde{w}^r$, we have
\[
\langle \nabla F(w^m), \tilde{w}^r - w^m \rangle \leq F(\tilde{w}^r) - F(w^m) - \frac{1}{2\rho^+_s} \| \nabla F(\tilde{w}^r) - \nabla F(w^m) \|^2 
\]
\[
\leq (1 - \sigma)[F(\tilde{w}^r) - F(w^m)]
\] (25)
where the last inequality holds due to the fact that we assume that there exists a constant factor $\sigma > 0$ making this inequality true. Then by using the RSS condition, we have
\[
F(\tilde{w}^r) - F(w^m) \leq \langle \nabla F(w^m), \tilde{w}^r - w^m \rangle + \frac{\rho^+_s}{2} \| \tilde{w}^r - w^m \|^2 
\]
\[
\leq (1 - \sigma)[F(\tilde{w}^r) - F(w^m)] + \frac{\rho^+_s}{2} \| \tilde{w}^r - w^m \|^2.
\] (26)
After simplification, we have $F(\tilde{w}^r) - F(w^m) \leq \frac{\rho^+_s}{2\sigma} \| \tilde{w}^r - w^m \|^2$. Using Lemma 1 we have
\[
\| \tilde{w}^r - w^m \|^2 = \| \mathcal{H}_k(w^m) - w^m \|^2 \leq \frac{|I| - s}{|I| - s^*} \| w^m - w^* \|^2,
\] (27)
and
\[
F(\tilde{w}^r) - F(w^m) \leq \frac{\rho^+_s(d - s)}{2\sigma(d - s^*)} \| w^m - w^* \|^2.
\] (28)
When $F(\tilde{w}^r) - F(w^m) < 0$, we can omit $\frac{1}{k}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)(F(\tilde{w}^r) - F(w^m))$ directly since $16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta$ is less than zero. Because this is the simple situation, we only analyze the complex one in our whole proof. Therefore, we have the following inequality:
\[
(1 + \frac{1}{k}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)\rho^+_s(d - s)\frac{2\sigma(d - s^*)}{|B|(n - 1)})(F(\tilde{w}^r - 1) - F(w^*)) 
\leq \mathbb{E} \| \tilde{w}^r - 1 - w^* \|^2 + \frac{1}{|B|(n - 1)}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)(F(\tilde{w}^r - 1) - F(w^*)) 
\]
\[
+ 16\rho^+_s \eta^2 m(2\eta m - 1) \mathbb{E} \| F(\tilde{w}^r - 1) - F(w^*) \| + \frac{4\eta^2 m\alpha}{k} \| \nabla \mathcal{I} F(w^*) \|^2
\] (29)
Since $\tilde{w}^r = \mathcal{H}_k(w^m)$, i.e., $\tilde{w}^r$ is the best $s$-sparse approximation of $w^m$, then we have the following result due to Lemma 2
\[
\| \tilde{w}^r - w^* \|^2 \leq (1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}) \| w^m - w^* \|^2.
\] (30)
Let $\alpha = 1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}$, and by combining (29) and (30), we have
\[
(1 + \frac{1}{k}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)\rho^+_s(d - s)\frac{2\sigma(d - s^*)}{|B|(n - 1)})(F(\tilde{w}^r - 1) - F(w^*)) 
\leq \mathbb{E} \| \tilde{w}^r - 1 - w^* \|^2 + \frac{\alpha}{k}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)(F(\tilde{w}^r - 1) - F(w^*)) 
\]
\[
+ 16\rho^+_s \eta^2 m(2\eta m - 1) \mathbb{E} \| F(\tilde{w}^r - 1) - F(w^*) \| + \frac{4\eta^2 m\alpha}{k} \| \nabla \mathcal{I} F(w^*) \|^2
\]
\[
\leq \frac{2\alpha}{\rho^+_s} \mathbb{E} \| F(\tilde{w}^r - 1) - F(w^*) \| + \frac{\alpha}{k}(16\rho^+_s \eta^2 m(1 + \frac{n - |B|}{|B|(n - 1)}) - 2m\eta \delta)(F(\tilde{w}^r - 1) - F(w^*)) 
\]
\[
+ 16\rho^+_s \eta^2 m(2\eta m - 1) \mathbb{E} \| F(\tilde{w}^r - 1) - F(w^*) \| + \frac{4\eta^2 m\alpha}{k} \| \nabla \mathcal{I} F(w^*) \|^2
\] (31)
where the last inequality follows from the RSC condition and the definition of $\tilde{T}$, i.e., $\tilde{T} = \text{supp}(HT(\nabla F(w^*), 2s)) \cup \text{supp}(w^*)$. 
Through proper simplification, we have the following result:

\[
\frac{\alpha}{K}(2m\eta\delta - 16\delta\rho_s^+\eta^2m(1 + \frac{n - |B|}{|B|(n - 1)})E[F(\bar{w}^r) - F(w^*)] \\
\leq \left(\frac{2\alpha}{\rho_s^+} + 16\delta\rho_s^+\eta^2m\alpha \frac{n - |B|}{|B|(n - 1)}\right)E[F(\bar{w}^{r-1}) - F(w^*)] + \frac{4\eta^2m\alpha}{k}\|\nabla F(w^*)\|^2
\]

(32)

where the first term of the LHS of (31) is omitted due to the fact that the coefficient of \(E\|\bar{w}^r - w^*\|^2\) is larger than zero, when \(8\rho_s^+\eta^2m(1 + \frac{n - |B|}{|B|(n - 1)}) - m\eta + \frac{\eta k\eta}{\delta} > 0\), which can be naturally satisfied under our parameter setting with \(s \geq \kappa_s\). Following from (32), we have

\[
E[F(\bar{w}^r) - F(w^*)] \leq \beta E[F(\bar{w}^{r-1}) - F(w^*)] + \frac{2\eta}{\delta(1 - \omega)}\|\nabla F(w^*)\|^2
\]

(33)

where \(\beta = \frac{k}{\eta m\rho_s^+\delta(1 - \omega)} + \frac{8\rho_s^+\eta(n - |B|)}{|B|(1 - \omega)(n - 1)}\) and \(\omega = 8\rho_s^+\eta(1 + \frac{n - |B|}{|B|(n - 1)})\). By applying (33) recursively, then we have the following desired bound with \(\beta \leq \frac{1}{2} < 1\):

\[
E[F(\bar{w}^r) - F(w^*)] \leq \left(\frac{1}{2}\right)^r E[F(\bar{w}^0) - F(w^*)] + \frac{\eta}{1 - \omega}\|\nabla F(w^*)\|^2.
\]

(34)

This completes the proof.

\(\square\)

C.2 Proof of Corollary 1

Proof. We then give the proof for the statistical estimation analysis. Following from the RSC condition of \(F(\cdot)\), we have

\[
F(w^*) \leq F(\bar{w}^r) + \langle \nabla F(w^*), w^* - \bar{w}^r \rangle - \frac{\rho_s^-}{2}\|\bar{w}^r - w^*\|^2.
\]

(35)

Let \(\phi = \left(\frac{1}{2}\right)^r E[F(\bar{w}^0) - F(w^*)] + \frac{\eta}{1 - \omega}\|\nabla F(w^*)\|^2\). Combining (34) and (35), we have

\[
E[F(\bar{w}^r) - \phi] \leq F(w^*) \leq F(\bar{w}^r) + \langle \nabla F(w^*), w^* - \bar{w}^r \rangle - \frac{\rho_s^-}{2}\|\bar{w}^r - w^*\|^2.
\]

(36)

Using the duality of norms, we have

\[
E(\nabla F(w^*), w^* - \bar{w}^r) \leq \|\nabla F(w^*)\|_\infty E\|\bar{w}^r - w^*\|_1 \leq \sqrt{s}\|\nabla F(w^*)\|_\infty E\|\bar{w}^r - w^*\|.
\]

(37)

Combining (36), (37) and \(\|\bar{w}^r\|^2 \leq E\|w\|^2\), we have

\[
\frac{\rho_s^-}{2}(E\|\bar{w}^r - w^*\|^2) \leq \sqrt{s}\|\nabla F(w^*)\|_\infty E\|\bar{w}^r - w^*\| + \phi.
\]

(38)

Let \(\alpha = E\|\bar{w}^r - w^*\|\). From the above inequality, we solve the following quadratic function with respect to \(\alpha\),

\[
\frac{\rho_s^-}{2}a^2 - \sqrt{s}\|\nabla F(w^*)\|_\infty a - \phi \leq 0,
\]

(39)

which yields the bound

\[
E\|\bar{w}^r - w^*\| \leq \sqrt{\frac{2(\frac{1}{2})^r E[F(\bar{w}^0) - F(w^*)]}{\rho_s^-} + \frac{2\eta}{1 - \omega}\|\nabla F(w^*)\| + \frac{2\sqrt{s}\|\nabla F(w^*)\|_\infty}{\rho_s^-}}
\]

(40)

\[
\leq \sqrt{2(\frac{1}{2})^r E[F(\bar{w}^0) - F(w^*)]}\rho_s^- + \left(\frac{2}{\rho_s^-} + \frac{2\eta}{1 - \omega}\right)\sqrt{s}\|\nabla F(w^*)\|_\infty.
\]

This completes the proof.
C.3 Bound $\beta < 1$

Now we show that we can guarantee that $\beta \leq 1$ by providing the appropriate choices of constants with $\eta$, $k$ and $m$ used in the theorem. More specifically, let $\eta \leq \frac{C_2}{\rho_s}$, and $|\mathcal{B}| = 1$, then we have

$$\frac{8\rho_s^2 \eta(n - |\mathcal{B}|)}{|\mathcal{B}|(1 - \omega)(n - 1)} \leq \frac{8C_3(n - |\mathcal{B}|)}{|\mathcal{B}|(n - 1)(1 - 8C_3(1 + \frac{n - |\mathcal{B}|}{|\mathcal{B}|(n - 1)}) - \frac{1}{4}. \quad (41)$$

If $\eta \leq \frac{C_2}{\rho_s}$ with $C_2 \leq C_3$, then we have

$$\frac{k}{\eta \delta \rho_s^2 (1 - \omega)} \leq \frac{1}{23mC_2}. \quad (42)$$

If we guarantee $\frac{3}{4ka^2} \leq \frac{1}{4}$, then we have

$$m \geq \frac{6k\kappa_2}{\delta C_2} = C_4\kappa_2, \quad (43)$$

where $\delta > 1$. If we choose $C_2 = \frac{1}{60}$, $C_3 = \frac{1}{35}$, $C_4 = 1800$, $k = 10$ and $\delta = 2$, then we have $\beta \leq \frac{1}{7}$.

Note that the convergence rate $\beta$ does not include $\alpha$, therefore, the required value of $k$ comes from the restricted constraint of the coefficient of $\|\hat{w}^* - w^*\|^2$, which implies that $s = \Omega(\kappa_2 s^*)$.

C.4 Proof of Corollary 2

Proof.

$$\mathbb{E}[F(\tilde{w}^*) - F(w^*)] \leq \epsilon + \frac{\eta}{1 - \omega} \|\nabla F(w^*)\|^2. \quad (44)$$

Let $\varrho_1, \varrho_2, \ldots$ be a non-negative sequence of random variables which is defined as

$$\varrho_r = \max \left\{ F(\tilde{w}^r) - F(w^*) - \frac{\eta}{1 - \omega} \|\nabla F(w^*)\|^2, 0 \right\}. \quad (45)$$

For a fixed $\epsilon > 0$, it follows from (44) and the Markov inequality,

$$\mathbb{P}(\varrho_r \geq \epsilon) \leq \frac{\mathbb{E}\varrho_r}{\epsilon} \leq \frac{(\frac{1}{2})^r[F(\tilde{w}^r) - F(w^*)]}{\epsilon}. \quad (46)$$

For a given $\zeta \in (0, 1)$, let the right side of (46) be not greater than $\zeta$, which requires

$$r \geq \log_2 \frac{F(\tilde{w}^r) - F(w^*)}{\epsilon \zeta}. \quad (47)$$

Therefore, the result in (44) holds with probability at least $1 - \zeta$. Thus, we need $\mathcal{O}(\log(\frac{1}{\epsilon}))$ outer iterations to get $\hat{w}^r$ satisfying (44). Since within each outer-iteration, we need to calculate a full gradient and $m$ stochastic variance reduced gradients with mini-batch size $|\mathcal{B}|$, the overall oracle complexity is $\mathcal{O}((n + \frac{\kappa_2}{k}) \log \frac{1}{\epsilon})$. Note that in high-dimensional region, the effect of support of snapshot point can be ignored. On the other hand, we only perform a hard thresholding operation at each outer iteration. Then we have

$$\log_2 \left( \frac{1}{\epsilon} \right) = \frac{\log(\frac{1}{\epsilon})}{\log(2)} = \mathcal{O}(\log(\frac{1}{\epsilon})). \quad (48)$$

Thus, the hard thresholding oracle complexity is $\mathcal{O}(\log(\frac{1}{\epsilon}))$. This completes the proof. \qed

D Proof of Theorem of Sparse variant

In this section, we theoretically analyze the convergence properties of our Serial Sparse SBCD-HTP ($S^2$BCD-HTP).
Algorithm 3 Serial Sparse SBCD-HTP (S²BCD-HTP)

**Input:** number of outer-loops $R$, number of inner-loops $m$, step size $\eta$, sparsity level $s$;

**Initialize:** $\tilde{w}^0$;

1: for $r = 0, 1, \ldots, R - 1$ do
2: \[ w^0 = \tilde{w} = \tilde{w}^r; \]
3: \[ \nabla F(\tilde{w}) = \frac{1}{n} \sum^n_{i=1} \nabla f_i(\tilde{w}); \]
4: \[ \tilde{G} = \text{supp}(\tilde{w}); \]
5: for $t = 0, 1, \ldots, m - 1$ do
6: \[ S = \tilde{G} \cup \tilde{G}_t; \]
7: \[ \nabla_S g(w^t) = \nabla_S f_i([w^t]_{T_i}) - \nabla_S f_i([\tilde{w}]_{T_i}) + D_i \nabla_S F(\tilde{w}); \]
8: \[ w^{t+1} = w^t - \eta \nabla_S g([w^t]_{T_i}); \]
9: end for
10: $\tilde{w}^{r+1} = HT(w^m, s);$
11: end for

**Output:** $\tilde{w}^R.$

### D.1 Proof

**Proof.** We start with the iterate difference between $w^{t+1}$ and $w^*$. By expanding iterate difference and taking expectation with respect to all randomness, we get

\[
\mathbb{E}\|w^{t+1} - w^*\|^2 = \mathbb{E}\|w^t - \eta \nabla_S g_S(w^t) - w^*\|^2
\]
\[
= \|w^t - w^*\|^2 + \eta^2 \|\nabla_S g_S(w^t)\|^2 + 2\eta \mathbb{E}\langle \nabla_S g_S(w^t), w^* - w^t \rangle
\]
\[
\leq \|w^t - w^*\|^2 + \eta^2 \|\nabla_S g_S(w^t)\|^2 + \frac{2\eta}{k} \|\nabla F(w^t), w^* - w^t\|^2,
\]

where the last inequality holds by the unbiasedness of the sparse gradient estimator $\nabla_S g_S(w^t)$ and the expectation with respect to randomized block. By using the $\rho_S^*$-strongly convex condition, we get the bound for $\langle \nabla F(w^t), w^* - w^t \rangle$ as follows:

\[
\langle \nabla F(w^t), w^* - w^t \rangle \leq F(w^*) - F(w^t) - \frac{\rho_S^*}{2} \|w^* - w^t\|^2.
\]

Due to the support set $\mathcal{I}$, we need the extended RSC condition with $\delta > 1$, which has been introduced in the proof of Lemma 5. Based on this underlying consensus, we can obtain the bound for $\langle \nabla F(w^t), w^* - w^t \rangle$.

\[
\langle \nabla F(w^t), w^* - w^t \rangle \leq \delta[F(w^*) - F(w^t)].
\]

Using Lemma 3, we have the following sparse variance gradient bound,

\[
\mathbb{E}[\|\nabla_S g_S(w^t)\|^2] \leq \frac{1}{k} \left( 12\delta \rho_S^*[F(w^t) - F(w^*)] + (3 + 12(D_m^2 - 1)) \|\nabla F(w^*\|^2
\]
\[
+ (24\delta \rho_S^2 + 48\delta \rho_S^2 (D_m^2 - 1))[F(\tilde{w}) - F(w^*)] \right).
\]

By combining the above inequalities, we have

\[
\mathbb{E}[\|w^{t+1} - w^*\|^2 \leq \|w^t - w^*\|^2 + \eta^2 \|\nabla_S g_S(w^t)\|^2 - \frac{2\eta\delta}{k}[F(w^t) - F(w^*)]
\]
\[
\leq \|w^t - w^*\|^2 + \frac{12\eta^2 \delta \rho_S^2 - 2\eta\delta}{k}[F(w^t) - F(w^*)]
\]
\[
+ \frac{2\delta \eta^2}{k} (12 \rho_S^2 + 24 \rho_S^2 (D_m^2 - 1))[F(\tilde{w}) - F(w^*)]
\]
\[
+ \frac{\eta^2}{k} (3 + 12(D_m^2 - 1)) \|\nabla F(w^*)\|^2,
\]

20
where the second inequality holds by Lemma [3]. Summing the above inequality over $t = 0, \ldots, m - 1$ and taking expectation with respect to all randomness in this epoch, we have

$$
\mathbb{E}[\|w^m - w^*\|^2] \leq \mathbb{E}[\|\tilde{w}^{r-1} - w^*\|^2] - \frac{2\eta \delta m (1 - 6\eta \rho^+_{\alpha})}{k} [F(w^m) - F(w^*)]
$$

$$
+ \frac{\delta \eta^2 m}{k} (12\rho^+_{\alpha} + 24\rho^+_{\alpha}(D^2_m - 1)) [F(\tilde{w}) - F(w^*)]
$$

$$
+ \frac{\eta^2 m}{k} (3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2.
$$

(54)

Using (28), we obtain

$$
\mathbb{E}[\|w^m - w^*\|^2] \leq \mathbb{E}[\|\tilde{w}^{r-1} - w^*\|^2] - \frac{2\eta \delta m (1 - 6\eta \rho^+_{\alpha})}{k} [F(\tilde{w}^r) - F(w^*)]
$$

$$
+ \frac{\delta \eta^2 m}{k} (24\rho^+_{\alpha} + 48\rho^+_{\alpha}(D^2_m - 1)) [F(\tilde{w}^{r-1}) - F(w^*)]
$$

$$
+ \frac{\eta^2 m}{k} (3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2.
$$

(55)

Since $\tilde{w}^r = HT(w^m, s)$, i.e., $\tilde{w}^r$ is the best $s$-sparse approximation of $w^m$, and due to Lemma [2], then we have the following result,

$$
\frac{2\eta \delta m}{k} (1 - 6\eta \rho^+_{\alpha}) [F(\tilde{w}^r) - F(w^*)] + \frac{\delta \eta^2 m}{k} (24\rho^+_{\alpha} + 48\rho^+_{\alpha}(D^2_m - 1)) [F(\tilde{w}^{r-1}) - F(w^*)]
$$

$$
+ \frac{\eta^2 m}{k} (3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2 
$$

$$
\leq \alpha \mathbb{E}[\|\tilde{w}^{r-1} - w^*\|^2] + \frac{\delta \eta^2 m}{k} (24\rho^+_{\alpha} + 48\rho^+_{\alpha}(D^2_m - 1)) [F(\tilde{w}^{r-1}) - F(w^*)]
$$

$$
+ \frac{\eta^2 m}{k} (3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2,
$$

(56)

where $\alpha = 1 + \frac{2\eta \sigma}{\sqrt{\rho_{\alpha} - \rho^+_{\alpha}}}$, and we use $\rho^+_{\alpha}$-strongly convex of $F(w)$ to bound $\|\tilde{w}^{r-1} - w^*\|^2$ in the last inequality. Through proper simplification, we can obtain

$$
\frac{2\eta \delta m}{k} (1 - 6\eta \rho^+_{\alpha}) [F(\tilde{w}^r) - F(w^*)] 
$$

$$
\leq \left( \frac{2\alpha}{\rho^+_{\alpha}} + \frac{(24\rho^+_{\alpha} + 48\rho^+_{\alpha}(D^2_m - 1)) \delta \eta^2 m}{k} \right) [F(\tilde{w}^{r-1}) - F(w^*)]
$$

$$
+ \frac{\eta^2 m}{k} (3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2,
$$

(57)

where the second term of LHS of (56) is omitted due to the fact that the coefficient of $\mathbb{E}[\|\tilde{w}^r - w^*\|^2]$ is larger than zero, when $6\eta \rho^+_{\alpha} m^2 - m \eta + \frac{\kappa \sigma (d - s^*)}{\rho^+_{\alpha} \delta (d - s^*)} > 0$, which can be naturally satisfied under our parameter setting with $s \geq \kappa \eta s^*$. Therefore, we can obtain the following result,

$$
\mathbb{E}[F(\tilde{w}^r) - F(w^*)] \leq \frac{2\eta \delta m}{k} (1 - 6\eta \rho^+_{\alpha}) [F(\tilde{w}^{r-1}) - F(w^*)]
$$

$$
+ \frac{(3 + 12(D^2_m - 1)) \delta m \eta^2}{2\eta \delta m (1 - 6\eta \rho^+_{\alpha})} \|\nabla F(w^*)\|^2.
$$

(58)
where $\hat{I} = \text{supp}(w^*) \cup \text{supp}(HT(\nabla F(w^*), 2s))$. We define $\xi := \frac{[3+12(D_m^2-1)]\eta}{2^9(1-6\rho_s^2)\eta} \|
abla F(w^*)\|^2$. In order to obtain a linear convergence rate, we have to bound

$$
\beta := \frac{2k\alpha}{\rho_s^2} + (24 + 48(D_m^2 - 1))\delta m \alpha \rho_s^2 \eta^2 = \frac{k}{(\eta - 6\rho_s^2 \eta^2)\delta m \rho_s^2} + \frac{(12 + 24(D_m^2 - 1))\rho_s^2 \eta}{1 - 6\rho_s^2 \eta} < 1. 
$$

(59)

If we choose $\eta \leq \frac{1}{48D_m^2\rho_s^2}$, then the second term of $\beta$ will be less than $\frac{1}{2}$, i.e., $\frac{(12+24(D_m^2-1))\rho_s^2 \eta}{1 - 6\rho_s^2 \eta} \leq \frac{1}{2}$. As for the first term, when $m \geq \frac{500D_m^2\kappa_2^2}{\delta^2 m^2 - 1}$ with $\eta \geq \frac{1}{50D_m^2\rho_s^2}$, $\delta = 4$ and $k = 10$, we have $\frac{1}{(\eta - 6\rho_s^2 \eta^2)\delta m \rho_s^2} \leq \frac{1}{4}$. Thus, we get the linear convergence for our serial sparse SBCD-HTP ($S^2$BCD-HTP).

$$
E[F(\tilde{w}^r) - F(w^*)] \leq \frac{3}{4} \cdot |F(\tilde{w}^{r-1}) - F(w^*)| + \xi, \quad (60)
$$

which means that the total gradient evaluation oracle complexity is $O((n + \frac{n^2}{k} \log \frac{1}{\epsilon})$, and the hard thresholding oracle complexity is $O(\log(\frac{1}{\epsilon})).$ This completes the proof. \qed

## E PROOF OF THEOREM 2

In this section, we analyze the convergence properties of the proposed Asynchronous Sparse SBCD-HTP (ASBCD-HTP). We first have to specify the iterates labeling order used in our asynchronous analysis.

We use “After Read” labeling order ([Leblond et al., 2017]) in our proof, which enjoys a simpler analysis but requires the order of randomly sampling step to be an uniform distributed sample. In order to give a clear proof, we adopt the “After Read” labeling order and make the following assumptions:

**Assumption 4** The labeling order increases after Step 8 in Algorithm[2] finished, and thus the future perturbation is not considered in the effect of asynchrony in the current step.

**Assumption 5** We assume that uniform distributed samples and the independence of the sample $i_{t+1}$ with $\hat{w}_t$.

Following ([Leblond et al., 2017]), we can explicitly the effect of asynchrony as follows:

$$
\hat{w}_t - w^t = \eta \sum_{u=(t-\tau)_+}^{t-1} G_u \nabla g(\hat{w}_u), \quad (61)
$$

where $w^t$ is the $t$-th inner iteration solution. In the proof of Theorem 2, we use $w^t$ to represent the temporary solution of the $t$-th iteration for a clean representation. And $G_u$ is a diagonal matrix with entries in $\{0, +1\}$. This explicitly defines the coordinate perturbation from the past updates. Here, $\tau$ denotes the maximum number of overlaps between concurrent threads. We also denote $\Delta = \max_{j=1,\ldots,d} p_j$, which provides a measure of sparsity following ([Leblond et al., 2017]).

### E.1 Proof

**Proof** We analyze our asynchronous algorithm based on the “perturbed iterate analysis” framework ([Mania et al., 2017]) with “After Read” labeling order. By expanding the iterate difference and taking expectation with respect to the sample $i_{t+1}$, we get
Under review as a conference paper at ICLR 2020

\[ E_{i_t+1} \| w^{t+1} - w^* \|^2 = E_{i_t+1} \| w^t - \eta \nabla g_{\mathcal{I}}(\hat{w}_t) - w^* \|^2 \]

\[ = \| w^t - w^* \|^2 + \eta^2 \| \nabla g_{\mathcal{I}}(\hat{w}_t) \|^2 + 2 \eta E_{i_t+1} \langle w^* - w^t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle \]

\[ \leq \| w^t - w^* \|^2 + \eta^2 \| \nabla g_{\mathcal{I}}(\hat{w}_t) \|^2 + \frac{2 \eta}{k} E_{i_t+1} \langle w^* - \hat{w}_t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle + 2 \eta E_{i_t+1} \langle \hat{w}_t - w^t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle, \]  

(62)

where \( \hat{w}_t \) is the “perturbed” iterate with perturbation \( \psi \) and the last inequality follows from the expectation with respect to randomized block. We need to bound the term, \( 2 \eta \langle w^* - \hat{w}_t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle \).

Since \( i_{t+1} \) is independent to \( \hat{w}_t \), we can write:

\[ E_{i_t+1} \langle w^* - \hat{w}_t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle = \langle w^* - \hat{w}_t, E_{i_t+1} \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle = \langle w^* - \hat{w}_t, \nabla_{\mathcal{I}} F(\hat{w}_t) \rangle. \]  

(63)

We can use \( \frac{\rho_2}{\hat{w}} \)-strong convexity bound of \( F(\hat{w}) \) with constant factor \( \delta > 1 \) as well as a squared triangle inequality to get:

\[ -\langle \hat{w}_t - w^*, \nabla F(\hat{w}_t) \rangle \leq -\delta (F(\hat{w}_t) - F(w^*)) - \frac{\rho_2}{2} \| \hat{w}_t - w^* \|^2, \]  

(64)

\[ -\| \hat{w}_t - w^* \|^2 \leq \| \hat{w}_t - w^* \|^2 - \frac{1}{2} \| w^t - w^* \|^2, \quad (\|a + b\| \leq 2\|a\|^2 + 2\|b\|^2), \]  

(65)

\[ \langle w^* - \hat{w}_t, \nabla F(\hat{w}_t) \rangle \leq -\delta (F(\hat{w}_t) - F(w^*)) + \frac{\rho_2}{2} \| \hat{w}_t - w^* \|^2 - \frac{\rho_2}{4} \| w^t - w^* \|^2. \]  

(66)

Then we make an identical deformation, and we have

\[ \frac{2 \eta}{k} E_{i_t+1} \langle w^* - \hat{w}_t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle \leq -\frac{\eta \rho_2}{k} E_{i_t+1} \| w^t - w^* \|^2 + \frac{\eta \rho_2}{k} E_{i_t+1} \| \hat{w}_t - w^t \|^2 \]

\[ - \frac{2 \eta \delta}{k} (F(\hat{w}_t) - F(w^*)), \]  

(67)

Putting it all together, we get the initial recursive inequity, written here explicitly:

\[ a^{t+1} \leq (1 - \frac{\eta \rho_2}{2k}) a^t + \eta \| \nabla g_{\mathcal{I}}(\hat{w}_t) \|^2 + \frac{\eta \rho_2}{k} E_{i_t+1} \| \hat{w}_t - w^t \|^2 \]

\[ + 2 \eta E_{i_t+1} \langle \hat{w}_t - w^t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle - \frac{2 \eta \delta}{k} \hat{e}^t, \]  

(68)

where \( a^t := E_{i_t} \| w^t - w^* \|^2 \) and \( \hat{e}^t := E_{i_{t+1}} [F(\hat{w}_t) - F(w^*)] \).

From Lemma 1 in [Leblond et al. 2017], we can bound the asynchronous terms \( \| \hat{w}_t - w^t \|^2, \langle \hat{w}_t - w^t \rangle \)

\[ \langle \hat{w}_t - w^t, \nabla g_{\mathcal{I}}(\hat{w}_t) \rangle = \eta \sum_{u=(t-r)}^{t-1} E_{t} \nabla g_{\mathcal{I}}(\hat{w}_u), \nabla g_{\mathcal{I}}(\hat{w}_u) \]

\[ \leq \eta \sum_{u=(t-r)}^{t-1} E \| \nabla g_{\mathcal{I}}(\hat{w}_u) \| \]  

(69)

\[ \leq \eta \sum_{u=(t-r)}^{t-1} \frac{\sqrt{\Delta}}{2} (E \| \nabla g_{\mathcal{I}}(\hat{w}_u) \|^2 + E \| \nabla g_{\mathcal{I}}(\hat{w}_u) \|^2) \]

\[ \leq \eta \frac{\sqrt{\Delta}}{2} \sum_{u=(t-r)}^{t-1} E \| \nabla g_{\mathcal{I}}(\hat{w}_u) \|^2 + \eta \frac{\sqrt{\Delta}}{2} E \| \nabla g_{\mathcal{I}}(\hat{w}_u) \|^2. \]
where $G_u$ are $d \times d$ diagonal matrices whose entries in $\{0, +1\}$. Each update in $\hat{w}_t$ is already in $w^t$—this is the case of $0$. On the contrary, some updates might be late: this is the case of $+1$. Then by combining (68), (69) and (70), we get

\begin{align}
a_t^{t+1} &\leq (1 - \frac{\eta^2 \rho_t^2}{2k})a_t^t + \eta^2 \|\nabla g_{\mathcal{I}}(\hat{w}_t)\|^2 + \eta^2 \rho_t^2 (1 + \sqrt{\Delta_\tau}) + \sum_{u=(t-\tau)_+}^{t-1} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 - \frac{2\eta}{k} \hat{e}^t
+ \eta^2 \sqrt{\Delta} \sum_{u=(t-\tau)_+}^{t-1} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 + \eta^2 \sqrt{\Delta_r} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 - \frac{2\eta}{k} \hat{e}^t
\leq (1 - \frac{\eta^2 \rho_t^2}{2k})a_t^t + \eta^2 C_1 E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 + \eta^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 - \frac{2\eta}{k} \hat{e}^t,
\end{align}

(71)

where $C_1 := 1 + \sqrt{\Delta_\tau} + \eta \rho_t^2 C_1$. $\tau$ represents the maximum number of overlaps between concurrent threads. And $\Delta = \max_{j=1, \ldots, d, p_j}$, which is a measure of sparsity with $1/n \leq \Delta \leq 1$.

Using Lemma 9 to bound the sparse variance term, the above inequality becomes

\begin{align}
a_t^{t+1} &\leq (1 - \frac{\eta^2 \rho_t^2}{2k})a_t^t + \eta^2 C_1 (12 \Delta_t^2 + \|F(\hat{w}_t) - F(w^*)\|) + (24 + 48(D_m^2 - 1)) \delta \rho_t^2 \|F(\hat{w}) - F(w^*)\|
+ (3 + 12(D_m^2 - 1)) \|\nabla F(w^*)\|^2 + \eta^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 - \frac{2\eta}{k} \hat{e}^t
\leq a_t^t + \frac{\eta^2 C_2}{k} \sum_{u=(t-\tau)_+}^{t-1} E_{u_{i+1}} \|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2 - \frac{2\eta}{k} \hat{e}^t
+ (24 \Delta_t^2 + 48 \rho_t^2 (D_m^2 - 1)) \frac{\delta C_1 \eta^2}{k} \hat{e}^t + (3 + 12(D_m^2 - 1)) \frac{\eta^2 C_2}{k} \|\nabla F(w^*)\|^2,
\end{align}

(72)

where $\hat{e} := \mathbb{E}[F(\hat{w}) - F(w^*)], \hat{e}^t := \mathbb{E}[F(\hat{w}_t) - F(w^*)]$. Note that we have $\hat{w} = w^0 = \hat{w} \tau^{-1}$. Summing (72) over $t = 0, \cdots, m - 1$ and taking expectation with all randomness in this epoch, we get

\begin{align}
a_m \leq a_0^0 - \frac{2\eta}{k} \delta m (1 - 6 \rho_s^2 \eta C_1) \hat{e}^m + \frac{\eta^2 C_2}{k} \sum_{t=1}^{m} \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}[\|\nabla g_{\mathcal{I}}(\hat{w}_u)\|^2] + (24 \Delta_t^2 + 48 \rho_t^2 (D_m^2 - 1)) \frac{\delta C_1 \eta^2}{k} \hat{e}^m + (3 + 12(D_m^2 - 1)) \frac{\eta^2 C_2}{k} \|\nabla F(w^*)\|^2,
\end{align}

(73)
where \( a^m = \mathbb{E}\|\omega^m - \omega^*\|^2 \), and \( a^0 = \mathbb{E}\|\omega^0 - \omega^*\|^2 \). Then we focus on upper bounding the third term on the RHS of (73).

\[
\sum_{t=1}^{m} \sum_{u=(t-\tau)+}^{t-1} \mathbb{E}[\|\nabla g_x(\hat{w}_u)\|^2] \leq \tau \sum_{t=1}^{m-1} \mathbb{E}[\|\nabla g_x(\hat{w}_t)\|^2] \leq \tau \sum_{t=1}^{m} \mathbb{E}[\|\nabla g_x(\hat{w}_t)\|^2] 
\leq \tau(12 \rho_s^2 \delta m \bar{e}^m + \delta m (24 \rho_s^2 + 48 \rho_s^2 (D^2_m - 1)) \bar{e}^{r-1} + m(3 + 12(D^2_m - 1)) \|\nabla F(w^*)\|^2).
\]

(74)

Substituting the above inequality into (73), we obtain

\[
a^m \leq a^0 - \frac{2\eta \delta m}{k}(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1)) \bar{e}^m + \frac{\delta \rho_s^2 m \eta^2}{k}(24 + 48(D^2_m - 1))(\tau C_2 + C_1) \bar{e}^{r-1} + \frac{\eta^2 m}{k}(3 + 12(D^2_m - 1))(C_1 + \tau C_2) \|\nabla F(w^*)\|^2.
\]

(75)

Then we bring (28) into the above inequality, we have

\[
a^m \leq a^0 - \frac{2\eta \delta m}{k}(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1)) \mathbb{E}[F(\hat{w}_m) - F(\bar{w}^*)] + \frac{\delta \rho_s^2 m \eta^2}{k}(24 + 48(D^2_m - 1))(\tau C_2 + C_1) \bar{e}^{r-1} + \frac{\eta^2 m}{k}(3 + 12(D^2_m - 1))(C_1 + \tau C_2) \|\nabla F(w^*)\|^2
\]

\[
\leq a^0 - \frac{2\eta \delta m}{k}(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1)) \bar{e}^r + \frac{\delta \rho_s^2 m \eta^2}{k}(24 + 48(D^2_m - 1))(\tau C_2 + C_1) \bar{e}^{r-1} + \frac{\eta^2 m}{k}(3 + 12(D^2_m - 1))(C_1 + \tau C_2) \|\nabla F(w^*)\|^2.
\]

(76)

where the second inequality holds since we use (28) and make a proper simplification. Using Lemma 2 we have

\[
(1 - 2\eta \delta m(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1))) \frac{\rho_s^2 (d - s)}{2k \sigma(d - s^*)} \bar{a}^r \leq \frac{2\alpha}{\rho_s} \bar{e}^{r-1} - \frac{2\eta \delta m}{k}(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1)) \bar{e}^r + \frac{\alpha \delta \rho_s^2 m \eta^2}{k}(24 + 48(D^2_m - 1))(\tau C_2 + C_1) \bar{e}^{r-1} + \frac{\alpha \eta^2 m}{k}(3 + 12(D^2_m - 1))(C_1 + \tau C_2) \|\nabla F(w^*)\|^2,
\]

(77)

where \( \alpha = 1 + \frac{4\sqrt{\tau + \epsilon}}{\sqrt{ho_s^2}} \), \( \bar{a}^r = \|\bar{w}^r - w^*\|^2 \), and \( \hat{F} = \text{supp}(w^*) \cup \text{supp}(HT(\nabla F(w^*), 2s)). \) The first term of LHS of (77) is omitted due to the coefficient of \( \bar{a}^r \) is larger than zero, when \( 6 \rho_s^2 \eta (\tau C_2 + C_1) \eta^2 - m \eta + \frac{\sigma(d - s^*)}{\delta \rho_s^2 (d - s)} > 0 \), which can be naturally satisfied under our parameter settings with \( s \geq \kappa s^* \). Therefore, we obtain the following recursive inequality,

\[
\bar{e}^r \leq \frac{2\kappa \alpha}{\rho_s} + \frac{2\kappa \alpha \rho_s^2 m \eta^2(24 + 48(D^2_m - 1))(\tau C_2 + C_1)}{2\kappa \delta \delta m(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1))(\tau C_2 + C_1)} \cdot \bar{e}^{r-1} + \frac{\alpha \eta^2 m(3 + 12(D^2_m - 1))(\tau C_2 + C_1)}{2\kappa \delta \delta m(1 - 6 \rho_s^2 \eta (\tau C_2 + C_1))} \cdot \|\nabla F(w^*)\|^2.
\]

(78)
We assume that the following settings has satisfied with the above constraint and then by choosing $m = 120\kappa\hat{s}, \eta = \frac{1}{60\rho}, k = 10$, we get
\[
\tilde{e}^r \leq \frac{20 + \frac{\delta}{75(\rho)^2} (1 + 2(D^2_m - 1)) (\tau C_2 + C_1)}{4\delta (1 - \frac{1}{10} (\tau C_2 + C_1))} \cdot \tilde{e}^{r-1} + \xi \cdot \|\nabla_2 F(w^*)\|^2, \quad (79)
\]
where $\xi = \eta (3 + 12(D^2_m - 1)) (\tau C_2 + C_1) \frac{\rho}{2(1 - 6\rho + \hat{s} \eta) (\tau C_2 + C_1)}$. In order to ensure linear speed up, $\tau$ needs to satisfy the following constraint,
\[
\theta \triangleq \frac{20 + \frac{\delta}{75(\rho)^2} (1 + 2(D^2_m - 1)) (\tau C_2 + C_1)}{4\delta (1 - \frac{1}{10} (\tau C_2 + C_1))} \leq 1. \quad (80)
\]
We assume that $3\delta(\rho^2) \geq D^2_m$ and $\delta = 10$, then by simply setting $\tau \leq \min\{\frac{3}{5\sqrt{\Delta}}, 2\kappa\hat{s}, \sqrt{\frac{2\kappa\hat{s}}{\sqrt{\Delta}}}\}$, the above constraint is satisfied with $\theta \leq 0.7538$, which implies that the oracle gradient evaluation and hard thresholding oracle complexities are $O((n + \frac{n}{\sqrt{\Delta}}) \log \frac{1}{\epsilon})$ and $O(\log(\frac{1}{\epsilon}))$, respectively. We can simply infer the statistical estimation result as before. Here, we will omit this part. This completes the proof. \qed

E.2 Discussion about Sparse Variance Bound

In the dense update case (i.e., $D_m = 1$), Lemma 6 is very similar to Lemma 5 when $|B| = 1$. In the sparse update case, Lemma 6 highly correlates with the sparsity of datasets ($\propto D^2_m$), which could be much loose in some extreme cases. If $D_m$ is very large (imaging a dataset with some dimensions contain only one entry among the $n$ samples, so $D_m = n$), the step size of sparse (asynchronous) SBCD-HTP will be much smaller than that of SBCD-HTP. Our algorithms can still converge to the optimal solution up to the statistical error. Actually, it is still an open problem whether we can have a tighter variance bound in the sparse update setting that is uncorrelated with $D_m$.

F More Experimental Results

Table 2: Summary of large-scale and high-dimensional datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th># Data</th>
<th># Feature</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcv1-train</td>
<td>20,242</td>
<td>47,236</td>
<td>0.16%</td>
</tr>
<tr>
<td>rcv1-test</td>
<td>677,399</td>
<td>47,236</td>
<td>0.15%</td>
</tr>
<tr>
<td>real-sim</td>
<td>72,309</td>
<td>20,958</td>
<td>0.024%</td>
</tr>
<tr>
<td>news20</td>
<td>19,996</td>
<td>30,000</td>
<td>0.49%</td>
</tr>
<tr>
<td>E2006-TFIDF</td>
<td>16,087</td>
<td>80,000</td>
<td>1.54%</td>
</tr>
</tbody>
</table>

F.1 Real-World Datasets and Equipments

We summarize the detailed information of all the real-world datasets in Table 2, including rcv1-train, rcv1-test, real-sim, news20 and E2006-TFIDF. All these datasets are provided in the LibSVM website\footnote{https://www.csie.ntu.edu.tw/~cjlin/libsvm/}. The E2006-TFIDF dataset includes 16,087 training data points and 150,360 features.
Instead of using all features, we randomly select 80,000 features for training. As for the news20 dataset, we randomly choose 30,000 features for training.

F.2 Baseline Methods

We compare our serial dense SBCD-HTP method with the state-of-the-art methods:

- **Fast Gradient with Hard Thresholding (FG-HT)** (Yuan et al., 2014): This method is based on a standard gradient descent step and combined with hard thresholding to solve sparsity-constrained problems.
- **Stochastic Gradient with Hard Thresholding (SG-HT)** (Nguyen et al., 2017): This method combines stochastic gradient descent with hard thresholding for solving large-scale sparsity-constrained problems.
- **Stochastic Variance Reduced Gradient with Hard Thresholding (SVRG-HT)** (Li et al., 2016b): This method incorporates the variance reduction technique in (Johnson & Zhang, 2013) into the hard thresholding algorithm to improve the efficiency of stochastic optimization.
- **Accelerated Stochastic Block Coordinate Gradient Descent with Hard Thresholding (ASBCD-HTP)** (Chen & Gu, 2016): This method uses block coordinate descent (BCD) to improve the efficiency of stochastic optimization.
- **Fast Newton Hard Thresholding Pursuit (FNHTP)** (Chen & Gu, 2017): This method tries to iteratively approximate the inverse Hessian matrix and combines a Newton algorithm with hard thresholding.

For the proposed Serial Sparse SBCD-HTP ($S^2$BCD-HTP) and Asynchronous sparse parallel SBCD-HTP (ASBCD-HTP) methods, we compare them with the following asynchronous parallel sparsity-constrained methods:

- **Asynchronous Accelerated Stochastic Block Coordinate Descent Hard Thresholding ($A^2$SBCD-HT)**: This algorithm is the asynchronous variant of ASBCD-HT by using our parallel framework.
- **Asynchronous Stochastic Gradient with Hard Thresholding (ASG-HT)**: This method is an extension of SG-HT from serial dense setting to asynchronous sparse setting.
- **Asynchronous Stochastic Variance Reduced Gradient Hard Thresholding (ASVRG-HT)** (Li et al., 2016a): This method incorporates multicore structure to compute stochastic variance reduced gradient in a parallel way.

F.3 Face Recognition Datasets

Although there are many datasets available for face recognition, we choose a common dataset (i.e. the Extended Yale B database (Georghiades et al., 2001)). The Extended Yale B database contains 2,414 frontal-face images of 38 people under different controlled lighting conditions (Georghiades et al., 2001). For each individual, we randomly choose 26 images for training and 15 images for testing.

F.4 Experimental Setup

For each sparsity-constrained algorithm, the sparse level makes great difference to the solution of sparse representation, especially at different noise levels. Therefore, in order to approach the best performance of all these algorithms, we change the sparsity parameter within a certain range for each algorithm. Thus, we can make sure that all these algorithms achieve the best recognition rates in the parameter setting. In all settings of the experiments, the images are down-sampled to 32×32 pixels.
Based on the sparse representation-based classification algorithm (SRC) (Wright et al., 2008), a series of processing operations are made to the above dataset. We first rescale the training matrix into $[0,1]$ for the convenience of adding noise, and then add Gaussian noise with zero mean and standard deviation $\vartheta$. Finally, we normalize the columns of the training matrix to have unit $\ell_2$-norm.

The parameter settings of comparison algorithms for face recognition task are the same as those described in Section 6.1.

**F.5 Results**

We first measure the effect of mini-batch size $|B|$ in the convergence of SBCD-HTP on rcv1-train dataset, as shown in Figure 6. It is obvious that the large mini-batch size can accelerate the convergence rate and make SBCD-HTP to obtain a better solution. This result is exactly in line with our theoretical analysis. However, when the mini-batch size is relatively large, it will deteriorate the performance of running time. One can simply choose the mini-batch size to 3 or 5 as suggested in Figure 6.
Then, we compare the performance of some hard thresholding algorithms, including SG-HT, SVRG-HT and ASBCDHT, with our SBCD-HTP for solving face recognition task on the Extended Yale B dataset. As shown in Figure 5, SBCD-HTP can achieve the best performance among these stochastic hard thresholding methods in a short time. It is clear that SBCD-HTP can not only optimize the objective function efficiently but also obtain a better representative solution on real applications, which is consistent with our theoretical results as described in Section 4.2.

![Comparison of all the algorithms for solving sparse logistic regression problems.](image)

**Figure 7:** Comparison of all the algorithms for solving sparse logistic regression problems. In each plot, the vertical axis shows the objective value minus the minimum, and the horizontal axis is the number of effective passes over data or running time (seconds).

We show more practical evaluation results for sparse linear regression and sparse logistic regression problems. All hardware environment and parameter settings are the same as in Section 6. For SBCD-HTP, from Figures 7 and 8, we can come to the same conclusion as we did before, that is in both sparse linear regression and sparse logistic regression, our algorithm converges faster and more accurate than the baseline algorithms in terms of effective passes and running time. However, we find an interesting result, which is that for the E2006-TFIDF dataset, all the algorithms converge with high precision solutions, so that we have to use some methods to magnify their differences. Of course, our SBCD-HTP has the best performance, but the gap is not that big. We think that this is due to ease with which E2006-TFIDF dataset can be optimized, since this phenomenon does not occur on other datasets.
For $S^2$BCD-HTP and ASBCD-HTP, from Figure 9, our $S^2$BCD-HTP and ASBCD-HTP far exceed the baseline algorithms in running time. Therefore, we can confirmedly obtain the conclusion that our Sparse and Asynchronous algorithms are the state-of-the-art hard thresholding algorithms for sparse datasets. This is a significant improvement for hard thresholding algorithms to solve sparse learning problems in high-dimensional asynchronous setting.

![Graphs showing performance comparison](image-url)

Figure 8: Comparison of all the algorithms for solving sparse linear regression problems.
Figure 9: Comparison of ASG-HT, ASVRH-HT, $A^2$SBCD-HT, $S^2$BCD-HTP and ASBCD-HTP for solving sparse linear regression problems. The four asynchronous parallel algorithms (i.e., ASG-HT, ASVRH-HT, $A^2$SBCD-HT and ASBCD-HTP) run on 20 threads.