Abstract
Optimizing with group-sparsity is significant in enhancing model interpretation in machine learning applications, e.g., model compression. However, for large-scale training problems, fast convergence and effective group-sparsity exploration are hard to achieved simultaneously in stochastic settings. Particularly, existing state-of-the-art methods, e.g., Prox-SG, RDA, Prox-SVRG and Prox-Spider, usually generate merely dense solutions. To overcome this shortage, we propose a novel stochastic method—Half-Space Proximal Stochastic Gradient Method (HSProx-SG) to promote the group sparsity of the solutions and maintain the convergence guarantee. In general, the HSProx-SG method contains two steps: (i) the proximal stochastic gradient step searches a near-optimal non-sparse solution estimate; and (ii) the half-space step substantially boosts the sparsity level. Numerically, HSProx-SG demonstrates its superiority in both convex settings and non-convex deep neural networks, e.g., VGG16 and ResNet18, by achieving solutions of much higher group sparsity and competitive objective values or generalization accuracy.

1 Introduction

In many recent machine learning optimization tasks, researchers not only focus on finding out the solutions that minimize the empirical loss to reduce the prediction error, but also concentrate on improving the interpretation of model by filtering out redundant parameters then achieve slimmer model architectures. One technique to achieve the above goal is by augmenting the sparsity-inducing regularization terms to the raw objective functions to generate sparse solutions (including numerous zero elements). The popular \( \ell_1 \)-regularization promotes the sparsity of solutions by element-wise penalizing the optimization variables. However, in many practical applications, there exist additional relationships and structures between the variables such that the nonzero coefficients are often not randomly distributed but tend to be clustered into disjoint groups \([1–3]\). As the most natural form of structured sparsity, the group-sparsity regularization, which assumes the pre-specified disjoint blocks of variables are selected (non-zero variables) or ignored (zero variables) simultaneously \([4]\), can be typically formulated as a non-smooth mixed \( \ell_1/\ell_2 \)-regularization problem, and has found many applications in computer vision \([5]\), signal processing \([6]\), medical imaging \([7]\), deep learning \([8]\), etc.
Problem Setting. In this paper, we study the following mixed $\ell_1/\ell_2$-regularization problem

$$\min_{x \in \mathbb{R}^n} \left\{ F(x) \overset{\text{def}}{=} f(x) + \lambda \Omega(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x) + \lambda \sum_{g \in \mathcal{G}} ||[x]_g|| \right\}, \quad (1)$$

where $\lambda > 0$ is a weighting factor, $|| \cdot ||$ denotes $\ell_2$-norm, $f(x)$ is the average of numerous $N$ continuously differentiable instance functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, such as the loss functions measuring the deviation from the observations in various data fitting problems, $\Omega(x)$ is the so-called mixed $\ell_1/\ell_2$ norm, $\mathcal{G}$ is a prescribed fixed partition of index set $\mathcal{I} = \{1, 2, 3, \ldots, n\}$, wherein each component $g \in \mathcal{G}$ indexes a group of variables upon the perspective of applications.

Theoretically, a larger $\lambda$ typically results in a higher group sparsity while sacrifices more on the bias of model estimation, hence $\lambda$ needs to be carefully fine-tuned to achieve both low $f(x)$ and high group-sparse solutions [4].

Literature Review. Problem (1) has been well studied in deterministic optimization with various algorithms that are capable of returning solutions with both low objective value and high group sparsity under proper $\lambda$ [1, 9, 10]. Proximal methods are classical approaches to solve the structured non-smooth optimization (1), including the popular proximal gradient method (Prox-FG) which only uses the first-order derivative information. When $N$ is huge, stochastic methods become ubiquitous to operate on a small subset to avoid the costly evaluation over all instances in deterministic methods for large-scale problems. Proximal stochastic gradient method (Prox-SG) [11] is the natural stochastic extension of Prox-FG. Regularized dual-averaging method (RDA) [12, 13] is proposed by extending the dual averaging scheme in [14]. To improve the convergence rate and reduce the variance due to the stochastic approximation, proximal stochastic variance-reduced gradient method (Prox-SVRG) [15] and proximal spider (Prox-Spider) [16] are developed via the well-known variance reduction technique SVRG proposed in [17] and Spider developed in [18] respectively.

Compared to deterministic methods, the studies of mixed $\ell_1/\ell_2$-regularization (1) in stochastic field become somewhat rare and limited. Prox-SG, RDA, Prox-SVRG and Prox-Spider are valuable state-of-the-art stochastic algorithms for solving problem (1) but with apparent weakness. Particularly, these existing stochastic algorithms typically meet difficulties to achieve both decent convergence and effective group sparsity identification simultaneously (e.g., small function values but merely dense solutions), because of the randomness and the limited sparsity-promotion mechanisms. In depth, Prox-SG, Prox-SVRG, RDA and Prox-Spider derive from proximal gradient method to utilize the proximal operator to produce group of zero variables. Such operator is generic to extensive non-smooth problems, consequently perhaps not sufficiently insightful if the target problems possess certain properties, e.g., the group sparsity structure as problem (1). In fact, in convex setting, the proximal operator suffers from variance of gradient estimate; and in non-convex setting, especially deep learning, the discrete step size (learning rate) further deteriorates its effectiveness on the group sparsity promotion, as will show in Section 2 that the projection region vanishes rapidly except RDA. Besides, the variance reduction techniques are typically required to measure over a huge mini-batch data points in both theory and practice which is probably prohibitive for large-scale problems, and have been observed as sometimes noneffective for deep learning applications [19]. On the other hand, to introduce sparsity, there exist heuristic weight pruning methods [20, 21], whereas they commonly do not equip with theoretical guarantee, so that easily diverge and hurt the generalization accuracy.

Our Contributions. Half-Space Proximal Stochastic Gradient Method (HSProx-SG) is designed to overcome the limitations of the existing stochastic algorithms on the group sparsity identification, while maintain comparable convergence characteristics. We summarize our contributions as follows.

- **Algorithmic Design:** We propose the HSProx-SG to solve the disjoint group sparsity regularized problem as (1). The algorithmic framework makes use of Prox-SG Step to predict a support cover of solutions, followed by locally exploiting the sparsity pattern via a novel Half-Space Step. We delicately design the Half-Space Step with the following main features: (i) it utilizes previous iterate as the normal direction to construct a reduced space consisting of a set of half-spaces and the origin; (ii) a new group projection operator maps groups of variables onto zero if they fall out of the constructed reduced space to identify group sparsity considerably more effectively than the proximal operator; and (iii) with proper step size, the Half-Space Step enjoys the sufficient decrease property, and achieves progress to optimum in both theory and practice.

- **Theoretical Guarantee:** We provide the convergence guarantees of HSProx-SG. Moreover, we prove HSProx-SG has looser requirements to identify the sparsity pattern than Prox-SG, revealing
We state the Half-Space Proximal Stochastic Gradient Method (HSProx-SG) in Algorithm 1. In the next iterate $x_{k+1}$ is then updated based on the following proximal mapping

$$x_{k+1} = \text{Prox}_{\alpha_k \lambda \Omega(\cdot)} (\tilde{x}_{k+1} : \arg\min_{x \in \mathbb{R}^n} \frac{1}{2\alpha_k} \|x - \tilde{x}_{k+1}\|^2 + \lambda \Omega(x), \quad (3)$$

where the regularization term $\Omega(x)$ is defined in (1). Notice that the above subproblem (3) has a closed-form solution, where for each $g \in G$, we have

$$[x_{k+1}]_g = \max \{ 0, 1 - \alpha_k \lambda / \|\tilde{x}_{k+1}\| \} \cdot [\tilde{x}_{k+1}]_g. \quad (4)$$

In HSProx-SG, the Prox-SG Step is proceeded $N_P$ times as a localization mechanism to seek an estimation which is close enough to a solution of problem (1) as drawn later in Theorem 2 of Section 3. In practice, although the close-enough requirement is perhaps hard to be verified, we empirically suggest to keep running Prox-SG Step until reaching some acceptable evaluation metrics, e.g., validation accuracy.

2 The HSProx-SG method

We state the Half-Space Proximal Stochastic Gradient Method (HSProx-SG) in Algorithm 1. In general, it contains two steps: Prox-SG Step (Algorithm 2) and Half-Space Step (Algorithm 3). The first step – Prox-SG Step (Algorithm 2) searches for a close-enough but usually non-sparse solution estimate. Then the second – Half-Space Step (Algorithm 3) begins with the non-sparse solution estimate. Then the second – Half-Space Step (Algorithm 3) begins with the non-sparse solution estimate to effectively exploit the group sparsity within a sequence of reduced spaces, and converges to the group-sparse solutions with theoretical convergence property.

Algorithm 1 Outline of HSProx-SG for solving (1).

1: \textbf{Input:} $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, 1)$, $\epsilon \in (0, 1)$, and $N_P \in \mathbb{Z}^+$.
2: for $k = 0, 1, 2, \ldots$ do
3: \hspace{1em} \textbf{if} $k < N_P$ \textbf{then}
4: \hspace{2em} Compute $x_{k+1} \leftarrow \text{Prox-SG}(x_k, \alpha_k)$ by Algorithm 2.
5: \hspace{1em} \textbf{else}
6: \hspace{2em} Compute $x_{k+1} \leftarrow \text{Half-Space}(x_k, \alpha_k)$ by Algorithm 3.
7: \hspace{1em} Update $\alpha_{k+1}$.
8: \textbf{end for}

Algorithm 2 Prox-SG Step.

1: \textbf{Input:} Current iterate $x_k$, and step size $\alpha_k$.
2: Compute the stochastic gradient of $f$ on mini-batch $B_k$.
3: Return $x_{k+1} \leftarrow \text{Prox}_{\alpha_k \lambda \Omega(\cdot)} (x_k - \alpha_k \nabla f_{B_k}(x_k))$.

Prox-SG Step. The Prox-SG Step performs the vanilla proximal stochastic gradient method to approach the solution of (1). At $k$th iteration, a mini-batch $B_k$ is sampled to generate an unbiased estimator of the full gradient of $f$ (line 2, Algorithm 2) to compute a trial iterate $\tilde{x}_{k+1} := x_k - \alpha_k \nabla f_{B_k}(x_k)$, where $\alpha_k$ is the step size, and $f_{B_k}$ is the average of the instance functions $f_i$ cross $B_k$. The next iterate $x_{k+1}$ is then updated based on the following proximal mapping

$$x_{k+1} = \text{Prox}_{\alpha_k \lambda \Omega(\cdot)} (\tilde{x}_{k+1}) := \arg\min_{x \in \mathbb{R}^n} \frac{1}{2\alpha_k} \|x - \tilde{x}_{k+1}\|^2 + \lambda \Omega(x), \quad (3)$$
However, the Prox-SG Step alone is insufficient to exploit the group sparsity structure, i.e., the computed solution estimate is typically fully dense, due to the randomness and the moderate truncation mechanism constrained in its projection region, i.e., the trial iterate \( \tilde{x}_{k+1} \) is projected to zero only if it falls into an \( \epsilon_2 \)-ball centered at the origin with radius \( \alpha_k \lambda \) by (4). Our remedy is to incorporate it with the following Half-Space Step, which exhibits a more effective sparsity promotion mechanism while still remains the sufficient decreasing property.

**Algorithm 3** Half-Space Step

1. **Input:** Current iterate \( x_k \), step size \( \alpha_k \), and \( \epsilon \).
2. Compute the stochastic gradient of \( F \) on \( \mathcal{I}^{\neq 0}(x_k) \) by mini-batch \( B_k \)
   \[
   \nabla F_{B_k}(x_k)|_{\mathcal{I}^{\neq 0}(x_k)} \leftarrow \frac{1}{|B_k|} \sum_{i \in B_k} \nabla F_i(x_k)|_{\mathcal{I}^{\neq 0}(x_k)} \tag{5}
   \]
3. Compute \( \hat{x}_{k+1}|_{\mathcal{I}^{\neq 0}(x_k)} \leftarrow [x_k - \alpha_k \nabla F_{B_k}(x_k)|_{\mathcal{I}^{\neq 0}(x_k)}] \) and \( \hat{x}_{k+1}|_{\mathcal{I}^0(x_k)} \leftarrow 0 \).
4. **for each group \( g \) in \( \mathcal{I}^{\neq 0}(x_k) \) do**
5. **if** \( [\hat{x}_{k+1}]_g \| [x_k]_g \| < \epsilon \| [x_k]_g \| \) **then**
6. \( [\hat{x}_{k+1}]_g \leftarrow 0 \).
7. **Return** \( x_{k+1} \leftarrow \hat{x}_{k+1} \).

**Half-Space Step.** The Half-Space Step is designed to effectively determine the groups of zero variables and capitalize convergence characteristic, which is in sharp contrast to other heuristic aggressive weight pruning methods but typically lacking theoretical guarantee [20, 21]. The underlying intuition of Half-Space Step is to project \( [x_k]_g \) to zero only if \( -[x_k]_g \) serves as a descent step to \( F(x_k) \), hence updating \( [x_{k+1}]_g \leftarrow [x_k]_g - [x_k]_g = 0 \) still results in some progress to the optimality. Before introducing that, we first define the following index sets for any \( x \in \mathbb{R}^n \):

\[
\mathcal{I}^0(x) := \{ g : g \in \mathcal{G}, [x]_g = 0 \} \quad \text{and} \quad \mathcal{I}^{\neq 0}(x) := \{ g : g \in \mathcal{G}, [x]_g \neq 0 \},
\]

(6)

where \( \mathcal{I}^0(x) \) represents the indices of groups of zero variables at \( x \), and \( \mathcal{I}^{\neq 0}(x) \) indexes the groups of nonzero variables at \( x \). To proceed, we further define an artificial set that \( x \) lies in:

\[
\mathcal{S}(x) := \{ z \in \mathbb{R}^n : [z]_g^T [x]_g \geq \epsilon \| [x]_g \| \quad \text{if} \quad g \in \mathcal{I}^{\neq 0}(x), \quad \text{and} \quad [z]_g = 0 \quad \text{if} \quad g \in \mathcal{I}^{0}(x) \} \cup \{0\},
\]

(7)

which consists of half-spaces and the origin. Hence, \( x \) inhabits \( \mathcal{S}(x_k) \), i.e., \( x \in \mathcal{S}(x_k) \), only if: (i) \( [x]_g \) lies in the upper half-space for all \( g \in \mathcal{I}^{\neq 0}(x_k) \) for some prescribed \( \epsilon \in [0, 1] \) as shown in Figure 1a; and (ii) \( [x]_g \) equals to zero for all \( g \in \mathcal{I}^{0}(x_k) \).

The fundamental assumption for Half-Space Step to success is that: the Prox-SG Step has produced a (possibly non-sparse) solution estimate \( x_k \), nearby a group sparse solution \( x^* \) of problem (1), i.e., the optimal distance \( \| x_k - x^* \| \) is sufficiently small. As seen in Appendix, it further indicates that the group sparse optimal solution \( x^* \) inhabits \( \mathcal{S}_k := \mathcal{S}(x_k) \), which implies that \( \mathcal{S}_k \) has already covered the group-support of \( x^* \), i.e., \( \mathcal{I}^{\neq 0}(x^*) \subseteq \mathcal{I}^{\neq 0}(x_k) \). Our goal now becomes minimizing \( F(x) \) over...
\( S_k \) to identify the remaining groups of zero variables, i.e., \( T^0(x^*)/T^0(x_k) \), which is formulated as the following smooth optimization problem:

\[
x_{k+1} = \arg\min_{x \in S_k} F(x) = f(x) + \lambda \Omega(x).
\]

By the definition of \( S_k \), \( [x]_{I^0(x_k)} = 0 \) are constrained as fixed during Algorithm 3 proceeding, and only the entries in \( T^\neq 0(x_k) \) are allowed to move. Hence \( F(x) \) is smooth on \( S_k \), and (8) is a reduced space optimization problem. A direct way to solve problem (8) would be the projected stochastic gradient descent method. However, such a method equipped with Euclidean projection rarely produces groups of variables onto zero as illustrated the nonzero \( x_E \) in Figure 1a.

To end this section, we intuitively illustrate the strength of HSProx-SG on group sparsity exploration. The above projector of form (9) is not the standard Euclidean projection operator in most cases, but still satisfies the following two advantages: (i) the actual search direction \( d_k := (\text{Proj}_{S_k}(\tilde{x}_{k+1}) - x_k) / \alpha_k \) performs as a descent direction to \( F_{B_k}(x_k) := f_{B_k}(x_k) + \lambda \Omega(x_k) \), then the progress to the optimum is made via sufficient decrease property as drawn in Lemma 1; and (ii) effectively project groups of variables to zero simultaneously if the inner product of corresponding entries is sufficiently small. In contrast, the Euclidean projection operator is far away effective to promote group sparsity, as the Euclidean projected point \( x_E \neq 0 \) versus \( x_{k+1} = \text{Proj}_{S_k}(\tilde{x}_{k+1}) = 0 \) shown in Figure 1a.

**Lemma 1.** Algorithm 3 yields the next iterate \( x_{k+1} \) as \( \text{Proj}_{S_k}(x_k - \alpha_k \nabla F_{B_k}(x_k)) \), then the search direction \( d_k := (x_{k+1} - x_k) / \alpha_k \) is a descent direction for \( F_{B_k}(x_k) \), i.e., \( d_k^T \nabla F_{B_k}(x_k) < 0 \). Moreover, letting \( L \) be the Lipschitz constant for \( \nabla F_{B_k} \) on the feasible domain, and \( \tilde{G}_k := T^\neq 0(x_k) \cap T^0(x_{k+1}) \) and \( \tilde{G}_k := T^\neq 0(x_k) \cap T^0(x_{k+1}) \) be the sets of groups which projects or not onto zero, we have

\[
F_{B_k}(x_{k+1}) \leq F_{B_k}(x_k) - \left( \alpha_k - \frac{\alpha_k^2 L}{2} \right) \sum_{g \in \tilde{G}_k} \| \nabla F_{B_k}(x_k) \|_g^2 - \left( 1 - \epsilon - \frac{\alpha_k^2 L}{2} \right) \sum_{g \in \tilde{G}_k} \|x_k\|_g^2. \tag{10}
\]

To end this section, we intuitively illustrate the strength of HSProx-SG on group sparsity exploration. In fact, the half-space projection (9) is a more effective sparsity promotion mechanism compared to existing methods. Particularly, it benefits from a much larger projection region to map a reference point \( \tilde{x}_{k+1} := x_k - \alpha_k \nabla F_{B_k}(x_k) \) or its variants to zero. As the 2D case described in Figure 1b, the projection regions of Prox-SG, Prox-SVRG and Prox-Spider are \( \ell_2 \)-balls with radius as \( \alpha_k \lambda \). In stochastic learning, especially deep learning tasks, the step size \( \alpha_k \) is usually selected around \( 10^{-3} \) to \( 10^{-4} \) or even smaller for convergence. Together with the common setting of \( \lambda \ll 1 \), their projection regions would vanish rapidly, resulting in the difficulties to produce group sparsity. As a sharp contrast, even though \( \alpha_k \lambda \) is near zero, the projection region of HSProx-SG \( \{ x : x_k^T x < (\alpha_k \lambda + \epsilon) \|x_k\| \} \) (seen in Appendix) is still an open half-space which contains these \( \ell_2 \) balls as well as RDA’s if \( \epsilon \) is large enough. Moreover, a positive control parameter \( \epsilon \) adjusts the level of aggressiveness of group sparsity promotion (9), i.e., the larger the more aggressive, and meanwhile maintains the progress to the optimality by Lemma 1. In practice, proper fine tuning \( \epsilon \) is sometimes required to achieve both group sparsity enhancement and sufficient decrease on objective value as will see in Section 4.

### 3 Convergence Analysis

In this section, we give the convergence guarantee of our HSProx-SG. Towards that end, we make the following assumption which is widely used in many optimization literature [23, 15, 24].

**Assumption 1.** Each \( f_i : \mathbb{R}^n \to \mathbb{R}, \) for \( i = 1, 2, \ldots, N \), is differentiable and bounded below. Their gradients \( \nabla f_i(x) \) is differentiable and bounded below. Their gradients \( \nabla f_i(x) \) is \( \|x\| \) where each \( g \in G \) is singleton, then the \( S_k \) degenerates to an orthant face [22].

---

1 Unless \( \Omega(x) \) is \( \|x\|_1 \) where each \( g \in G \) is singleton.
Notations: Let $x^*$ be an optimal solution of problem (1) with group sparsity property, $F^*$ be the global minimum value corresponding to $x^*$, and $\{x_k\}_{k=0}^{\infty}$ be the iterates generated from Algorithm 1.

Denote the gradient mapping of $F(x)$ and its estimator on mini-batch $B$ as $\xi(x) := \frac{1}{n}(x - \text{Prox}_{\eta}(x - \eta \nabla F(x)))$ and $\xi_B(x) := \frac{1}{n}(x - \text{Prox}_{\eta}(x - \eta \nabla F_B(x)))$ respectively. We say $x$ a stationary point of $F(x)$ if $\xi(x) = 0$.

The noise $\epsilon(x) := \xi_B(x) - \xi(x)$ is zero-mean because of the random sampling of mini-batch $B$, i.e., $E_B[\epsilon(x)] = 0$.

To be simple, let $\mathcal{X}$ be a neighborhood of $x^*$ as $\mathcal{X} := \{x : \|x - x^\ast\| \leq R\}$ with $R := \frac{4\delta_1^2 - 2\delta_2^2}{2\delta_1 + \epsilon}$, and $M$ be the supremum of $\|\partial F(x)\|$ on the compact set $\mathcal{X}$. To establish the convergence results, we further require the below assumption.

**Assumption 2.** The least and the largest $\ell_2$-norm of non-zero groups in $x^*$ are lower and upper bounded by some constants, i.e., $0 < \delta_1 := \min_{g \in \mathcal{X}^0}(\|x^\ast\|_g) \text{ and } 0 < 2\delta_2 := \max_{g \in \mathcal{X}^0}(\|x^\ast\|_g)$. Moreover, we request a common result of strict complementarity on any $B$, i.e., $0 < 2\delta_3 := \min_{g \in \mathcal{X}^0}(\lambda - \|\nabla F(x^\ast)\|_g))$ for regularization optimization [25, 26].

**Remark:** Assumption 1 implies that $\nabla F_B(x)$ measured on mini-batch $B$ is Lipschitz continuous on $\mathbb{R}^n$ with the same Lipschitz constant $L$, while $\nabla F_B(x)$ is not as shown in Appendix. However, the Lipschitz continuity of $\nabla F_B(x)$ still holds on $\mathcal{X} = \{x : \|x\|_g \leq \delta_1\}$ for each $g \in G$ by excluding a $\ell_2$-ball centered at the origin with radius $\delta_1$ from $\mathbb{R}^n$.

For simplicity, let $\nabla F_B(x)$ share the same Lipschitz constant $\lambda$ on $\mathcal{X}$ with $\nabla F_B(x)$, since we can always select the bigger value as their shared Lipschitz constant.

Now, we state the first main theorem of HSProx-SG.

**Theorem 1.** Suppose $f$ is convex on $\mathcal{X}$, $\epsilon \in \left[0, \frac{2\delta_1^2}{\delta_2}\right]$, $\|x_k - x^\ast\| \leq \frac{R}{2}$ for some $K \geq N_\mathcal{P}$. Set $k := K + t$ (for $t \in \mathbb{Z}^+$), step size $\alpha_k = \mathcal{O}(\frac{1}{\sqrt{N_\mathcal{P}}}) \in \left(0, \min\left\{\frac{1}{2}, \frac{\delta_1^2}{\|x^\ast\|_g}\right\}\right)$, and mini-batch size $|B_k| = \mathcal{O}(t) \leq N - \frac{N}{2M}$. Then for any $\tau \in (0, 1)$, we have $\{x_k\}$ converges to some stationary point in expectation with probability at least $1 - \tau$, i.e., $\mathbb{P}(\lim_{k \to \infty} E[\|\xi_{\alpha_k B_k}(x_k)\|]) = 0 \geq 1 - \tau$.

**Validity:** Theorem 1 only requires local convexity of $f$ on a neighborhood $\mathcal{X}$ of $x^\ast$ while itself can be non-convex in general. This local convexity-type assumption appears in many non-convex problem analysis, such as: tensor decomposition [27] and one-hidden-layer neural networks [28].

**Remark:** Theorem 1 implies that if the $K^\text{th}$ iterates locate close enough to $x^\ast$, the step size $\alpha_k$ and mini-batch size $|B_k|$ is set as above, (it further indicates $x^\ast$ inhabits the $\{x_k\}_{k \geq K}$ of all subsequent iterates updated by Half-Space Step with high probability, as shown in Appendix), then the Half-Space Step in Algorithm 3 guarantees the convergence to the stationary point. The $\mathcal{O}(t)$ mini-batch size is commonly used in the analysis of stochastic algorithms, e.g., Adam and Yogi [29]. Later based on numerical results in Section 4, we observe that a much weaker increasing or even constant mini-batch size is sufficient. In fact, experiments show that practically, a reasonably large mini-batch size can work well if the variance is not large. Although the assumption $\|x_k - x^\ast\| < R/2$ is hard to be verified in practice, setting $N_\mathcal{P}$ large enough usually performs quite well.

To satisfy the pre-requirement of Theorem 1, in next Theorem 2, we claim after $N_\mathcal{P}$ Prox-SG Steps, HSProx-SG computes iterate $x_{N_\mathcal{P}}$ sufficiently close to $x^\ast$ with high probability, and provides an upper bound of $N_\mathcal{P}$. This result is established under a popular Polyak-Lojasiewicz (PL) condition for non-smooth problem [30], i.e., there exists a $\mu > 0$ such that for any $x \in \mathbb{R}^n$ and $\eta > 0$,

$$\|\xi_\eta(x)\|^2 \geq 2\mu(F(x) - F^\ast).$$

(11)

**Theorem 2.** Suppose $f$ is convex and $F$ satisfies the PL condition (11). There exist some constants $C > 0, \gamma \in (0, 1/2L)$, for any fixed $\tau \in (0, 1)$, if $\alpha_k \equiv \alpha \equiv \min\{\frac{\gamma M^2}{2L^2(1-\gamma)^2}, \frac{1}{2\gamma L}, \frac{1}{8}\}$, and mini-batch size $|B_k| \equiv |B| > \frac{32\mu^2 M^2}{\gamma L R^2 - (2L^2 - 1)\alpha^2 C_0}$ for any $k < N_\mathcal{P}$, then $\|x_{N_\mathcal{P}} - x^\ast\|_2 \leq R/2$ holds with probability at least $1 - \tau$, i.e., $\mathbb{P}(\|x_{N_\mathcal{P}} - x^\ast\|_2 \leq R/2) \geq 1 - \tau$ for any $N_\mathcal{P}$ with $K := \left\lfloor \frac{\log(\text{poly}(\tau^2 R^2/2L)}{\log(1/2\mu \alpha)} \right\rfloor$, where poly(·) is some polynomial of assembled variables.
Remark: Theorem 2 implies that after sufficient number of iterations, Prox-SG produces an iterate $x_{N_{P}}$ that is $R/2$-close to $x^*$ with high probability. However, one should notice that Prox-SG does not guarantee any group sparsity property of $x_{N_{P}}$, due to the limited projection region and randomness. As will be demonstrated in Section 4, the following Half-Space Step will significantly boost the group sparsity level of the solution.

Since PL condition (11) for non-smooth problem or global convexity condition for $f$ implies all stationary points are global minimizers of problem (1), the convergence to global optimum guarantee is consequently attained as stated in Corollary 1.

Corollary 1. If the following conditions holds: (i) Polyak-Lojasiewicz (PL) condition, (ii) $f(x)$ is convex on $\mathbb{R}^n$, then under the assumption of Theorem 1, HSProx-SG computes iterate sequence which converges to global minimum of problem (1) in expectation, i.e., $\lim_{k \to \infty} \mathbb{E}[F_{B_k}(x_k)] = F^*$ w.h.p.

We end this section by the sparsity identification guarantee of HSProx-SG as stated in Theorem 3.

Theorem 3. If $N_{P} \geq N_{\text{P}}$ and $\|x_k - x^*\| \leq \frac{2\alpha_k \delta_3}{1 + \alpha_k L}$, then HSProx-SG yields next iterate $x_{k+1}$ such that $\mathcal{T}^0(x^*) \subseteq \mathcal{T}^0(x_{k+1})$. Moreover, if $\|x_k - x^*\| \leq \min \left\{ \frac{2\alpha_k \delta_3}{1 + \alpha_k L}, R \right\}$, then $\mathcal{T}^0(x^*) = \mathcal{T}^0(x_{k+1})$ and $\mathcal{T} \neq 0(x_{k+1}) = \mathcal{T}^0(x^*)$.

Remark: Theorem 3 shows that when $x_k$ is in the $\ell_2$-ball centered at $x^*$ with radius $\frac{2\alpha_k \delta_3}{1 + \alpha_k L}$, HSProx-SG identifies the optimal sparsity pattern, i.e., $\mathcal{T}^0(x^*) \subseteq \mathcal{T}^0(x_{k+1})$. In contrast, to identify the sparsity pattern, Prox-SG requires the iterates to fall into the $\ell_2$-ball centered at $x^*$ with radius $\alpha_k \delta_3$ [25]. Since $\alpha_k \leq 1/L$ and $\epsilon \in [0, 1)$, then $\frac{2\alpha_k \delta_3}{1 + \alpha_k L} \geq \alpha_k \delta_3$ implies that the $\ell_2$-ball of HSProx-SG contains the $\ell_2$-ball of Prox-SG, i.e., HSProx-SG has a stronger performance in sparsity pattern identification.

Moreover, if the iterate further satisfies $R$-close to $x^*$, then both support recovery and sparsity identification are established. Therefore, Theorem 3 reveals a better sparsity identification property of HSProx-SG than Prox-SG, and no similar results exist for other methods to our knowledge.

4 Numerical Experiments

In this section, we present results of several benchmark numerical experiments in both convex (logistic regression) and non-convex (deep neural network) settings to illustrate the superiority of HSProx-SG than other related algorithms on group sparsity exploration and the comparable convergence.

Logistic Regression (Convex): We first focus on the mixed $\ell_1/\ell_2$-regularized logistic regression problem given $N$ examples $(d_1, l_1), \ldots, (d_N, l_N)$ where $d_i \in \mathbb{R}^{n}$ and $l_i \in \{-1, 1\}$ with the form

$$
\text{minimize} \sum_{i=1}^{N} \frac{1}{N} \log(1 + e^{-l_i (x^T d_i + b)}) + \lambda \sum_{g \in \mathcal{G}} \|x_g\|_2,
$$

for binary classification with a bias $b \in \mathbb{R}$. We set the regularization parameter $\lambda$ as $100/N$ throughout the experiments since it yields high sparse solutions and low object value $f$’s, equally decompose the variables into 10 groups to form $\mathcal{G}$, and test problem (12) on 4 standard publicly available large-scale datasets from LIBSVM repository [31] as summarized in Table 1. (Four more datasets are additionally included in Appendix.) All convex experiments are conducted on a 64-bit operating system with an Intel(R) Core(TM) i7-7700K CPU @ 4.20 GHz and 32 GB random-access memory.

We run the solvers with a maximum number of epochs as 60. The mini-batch size $|\mathcal{B}|$ is set to be $\min\{256, \lfloor 0.01N \rfloor\}$ similarly to [24]. The step size $\alpha_k$ setting follows [Section 4][15]. Particularly,
We conduct all non-convex experiments on one GeForce GTX 1080 Ti GPU for 300 epochs with
we set $N$ which demonstrates that HSProx-SG reaches comparable convergence as Prox-SG and Prox-SVRG in
we mark the best values as bold to facilitate the comparison. Furthermore, Figure 2 plots the relative
L we first compute a Lipschitz constant $L$ as $\max_i ||d_i||^2 / 4$, then fine tune and select constant $\alpha_k \equiv \alpha = 1/L$ to Prox-SG and Prox-SVRG since it exhibits the best results. For RDA, the step size parameter $\gamma$ is finetuned as the one with the best performance among all powers of 10. For HSProx-SG, we set $\alpha_k$ as the same as Prox-SG and Prox-SVRG in practice.

We set $N_P$ as $30N/|B|$ such that Half-Space Step is triggered after employing Prox-SG Step 30 epochs, and the control parameter $\epsilon$ in (9) as 0.05. We skip Prox-Spider in this section since Prox-SVRG has been a superb representative to the Prox-SG with variance reduction techniques. The final objective value $F$ and $f$, and group sparsity in the solutions are reported in Table 2, where we mark the best values as bold to facilitate the comparison. Furthermore, Figure 2 plots the relative runtime of these solvers for each dataset, scaled by the runtime of the most time-consuming solver on that dataset.

Table 2 shows that our HSProx-SG is definitely the best solver on exploring the group sparsity of the solutions. In fact, HSProx-SG achieves the solutions of highest group sparsity on all datasets. The second best solver on group sparsity in convex settings is Prox-SVRG via reducing the variance. Note that Prox-SVRG failed to generate any group sparsity on kdda, on which HSProx-SG reaches 80% group sparsity instead, indicating the reliability of our half-space mechanism on group sparsity identification compared to the standard variance reduction techniques. Moreover, we observe that all solvers perform quite competitively in terms of final objective values (round up to 3 decimals) except RDA, which demonstrates that HSProx-SG reaches comparable convergence as Prox-SG and Prox-SVRG in practice. Finally, Figure 2 indicates that Prox-SG, RDA and HSProx-SG have similar computational cost to proceed, except Prox-SVRG due to its periodical full gradient computation.

**Deep Convolutional Neural Network (Non-Convex):** We now consider the popular non-convex problems, *e.g.*, Deep Convolutional Neural Network (CNN) for image classification tasks.

Specifically, we select two common CNN architectures, *i.e.*, VGG16 [32] and ResNet18 [33] on two benchmark datasets CIFAR10 [34] and Fashion-MNIST [35].

We conduct all non-convex experiments on one GeForce GTX 1080 Ti GPU for 300 epochs with
a mini-batch size of 128 and $\lambda$ as $10^{-3}$, since it returns competitive validation accuracy to the models trained without regularization. The step size $\alpha_k$ in Prox-SG, Prox-SVRG and HSProx-SG is initialized as 0.1, and decayed by a factor 0.1 periodically. Set each filter in the convolution layers as a group variable. Also set $N_P = 150N/|B|$ in HSProx-SG since running Prox-SG Step 150 epochs already achieves an acceptable validation accuracy. The control parameter $\epsilon$ in the half-space projection (9) is first set as 0, then fine tuned to be around 0.02 to favor the sparsity level whereas does not hurt the target objective $F$; the detailed procedure is in Appendix. We exclude RDA because of no acceptable experimental results attained during our tests with the step size parameter $\gamma$ setting throughout all powers of 10 from $10^{-3}$ to $10^3$.

Table 3 demonstrates the effectiveness and superiority of HSProx-SG in non-convex settings. In particular, (i) HSProx-SG computes remarkably higher group sparsity than other methods on all non-convex tests under both $\epsilon = 0$ and fine tuned $\epsilon$, of which the solutions are typically multiple times sparser in the manner of group than those of Prox-SG, while Prox-SVRG performs not comparably on the non-convex group sparsity identification since the variance reduction techniques may not work as desired for deep learning applications [19]; (ii) HSProx-SG performs competitively among the methods with respect to the final objective values $F$ and $f$ (see final $f$ comparison in Appendix). In addition, all the methods reach a comparable generalization performance on unseen test data.

Finally, we investigate the group sparsity evolution under different $\epsilon$’s. As shown in Figure 3b, HSProx-SG produces the highest group-sparse solutions compared with other methods. Notably, at the early $N_P$ iterations, HSProx-SG performs merely the same as Prox-SG. However, after switching to Half-Space Step at the 150th epoch, HSProx-SG outperforms all the other methods dramatically, and larger $\epsilon$ results in higher sparsity level. It is a strong evidence that our half-space based technique

| Backbone | Dataset  | Prox-SG | Prox-SVRG | HSProx-SG |
|----------|----------|---------|-----------|-----------|
| VGG16    | CIFAR10  | 0.59 / 53.95% / 90.37% | 0.82 / 14.73% / 89.42% | 0.59 / 74.60% / 91.16% |
| ResNet18 | Fashion-MNIST | 0.31 / 19.05% / 94.93% | 0.50 / 2.59% / 94.17% | 0.31 / 19.05% / 94.93% |

Table 3: Final $F$ / group sparsity / testing accuracy for tested algorithms on non-convex problems.
Figure 3: Evolution of $F$, group sparsity and testing accuracy on ResNet18 with CIFAR10.

is much more successful than the proximal mechanism and its variants in terms of the group sparsity identification. Besides, the evolutions of $F$ and testing accuracy confirm the comparability on convergence among the tested algorithms. Particularly, the objective $F$ generally monotonically decreases for small $\epsilon = 0$ to 0.02, and experiences a mild pulse after switch to Half-Space Step for larger $\epsilon$, e.g., 0.05, which matches Lemma 1. As a result, with the similar generalization accuracy, HSProx-SG allows dropping entire hidden units of networks, which may further achieve automatic dimension reduction and construct smaller model architectures for efficient inference.

5 Conclusions

We proposed a new Half-Space Proximal Stochastic Gradient Method (HSProx-SG) for disjoint group-sparsity induced regularized problem, which makes use of proximal stochastic gradient method to seek a near-optimal solution estimate, followed by a novel half-space group projection to effectively exploit the group sparsity structure. In theory, we provided the convergence guarantee, and showed its better sparsity identification performance. Experiments on both convex and non-convex problems demonstrated that HSProx-SG usually achieves solutions with competitive objective values and significantly higher group sparsity compared with state-of-the-arts stochastic solvers.
References

[1] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[2] J. Huang, T. Zhang, et al., “The benefit of group sparsity,” *The Annals of Statistics*, vol. 38, no. 4, pp. 1978–2004, 2010.

[3] J. Huang, X. Huang, and D. Metaxas, “Learning with dynamic group sparsity,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 64–71, IEEE, 2009.

[4] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al., “Structured sparsity through convex optimization,” *Statistical Science*, vol. 27, no. 4, pp. 450–468, 2012.

[5] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600–1607, IEEE, 2012.

[6] P.-Y. Chen and I. W. Selesnick, “Group-sparse signal denoising: non-convex regularization, convex optimization,” *IEEE Transactions on Signal Processing*, vol. 62, no. 13, pp. 3464–3478, 2014.

[7] S. Liu, J. Cao, H. Liu, X. Zhou, K. Zhang, and Z. Li, “Mri reconstruction via enhanced group sparsity and nonconvex regularization,” *Neurocomputing*, vol. 272, pp. 108–121, 2018.

[8] S. Scardapane, D. Commiello, A. Hussain, and A. Uncini, “Group sparse regularization for deep neural networks,” *Neurocomputing*, vol. 241, pp. 81–89, 2017.

[9] V. Roth and B. Fischer, “The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms,” in *Proceedings of the 25th international conference on Machine learning*, pp. 848–855, 2008.

[10] J. Huang, T. Zhang, and D. Metaxas, “Learning with structured sparsity,” *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3371–3412, 2011.

[11] J. Duchi and Y. Singer, “Efficient online and batch learning using forward backward splitting,” *Journal of Machine Learning Research*, vol. 10, no. Dec, pp. 2899–2934, 2009.

[12] L. Xiao, “Dual averaging methods for regularized stochastic learning and online optimization,” *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2543–2596, 2010.

[13] H. Yang, Z. Xu, I. King, and M. R. Lyu, “Online learning for group lasso,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1191–1198, 2010.

[14] Y. Nesterov, “Primal-dual subgradient methods for convex problems,” *Mathematical programming*, 2009.

[15] L. Xiao and T. Zhang, “A proximal stochastic gradient method with progressive variance reduction,” *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[16] J. Zhang and L. Xiao, “Multi-level composite stochastic optimization via nested variance reduction,” *arXiv preprint arXiv:1908.11468*, 2019.

[17] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Advances in neural information processing systems*, pp. 315–323, 2013.

[18] C. Fang, C. J. Li, Z. Lin, and T. Zhang, “Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator,” in *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

[19] A. Defazio and L. Bottou, “On the ineffectiveness of variance reduced optimization for deep learning,” in *Advances in Neural Information Processing Systems*, 2019.

[20] H. Li, A. Kadav, I. Durduanovic, H. Samet, and H. P. Graf, “Pruning filters for efficient convnets,” *arXiv preprint arXiv:1608.08710*, 2016.

[21] J.-H. Luo, J. Wu, and W. Lin, “Thinet: A filter level pruning method for deep neural network compression,” in *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

[22] T. Chen, T. Ding, B. Ji, G. Wang, Y. Shi, S. Yi, X. Tu, and Z. Zhu, “Orthant based proximal stochastic gradient method for $\ell_1$-regularized optimization,” *arXiv preprint arXiv:2004.03639*, 2020.
[23] T. Chen, F. E. Curtis, and D. P. Robinson, “A reduced-space algorithm for minimizing $\ell_1$-regularized convex functions,” *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1583–1610, 2017.

[24] M. Yang, A. Milzarek, Z. Wen, and T. Zhang, “A stochastic extra-step quasi-newton method for nonsmooth nonconvex optimization,” *arXiv preprint arXiv:1910.09373*, 2019.

[25] J. Nutini, M. Schmidt, and W. Hare, “‘active-set complexity’ of proximal gradient: How long does it take to find the sparsity pattern?,” *Optimization Letters*, vol. 13, no. 4, pp. 645–655, 2019.

[26] T. Chen, F. E. Curtis, and D. P. Robinson, “Farsa for $\ell_1$-regularized convex optimization: local convergence and numerical experience,” *Optimization Methods and Software*, 2018.

[27] R. Ge, F. Huang, C. Jin, and Y. Yuan, “Escaping from saddle points—online stochastic gradient for tensor decomposition,” in *Conference on Learning Theory*, pp. 797–842, 2015.

[28] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon, “Recovery guarantees for one-hidden-layer neural networks,” in *International Conference on Machine Learning*, 2017.

[29] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, “Adaptive methods for nonconvex optimization,” in *Advances in neural information processing systems*, pp. 9793–9803, 2018.

[30] Z. Li and J. Li, “A simple proximal stochastic gradient method for nonsmooth nonconvex optimization,” in *Advances in neural information processing systems*, 2018.

[31] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[34] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” *Master’s thesis, Department of Computer Science, University of Toronto*, 2009.

[35] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.