New Insight into Hybrid Stochastic Gradient Descent: Beyond With-Replacement Sampling and Convexity

Pan Zhou∗ Xiao-Tong Yuan† Jiashi Feng∗
∗ Learning & Vision Lab, National University of Singapore, Singapore
† B-DAT Lab, Nanjing University of Information Science & Technology, Nanjing, China
pzhou@u.nus.edu xtyuan@nuist.edu.cn elefjia@nus.edu.sg

Abstract

As an incremental-gradient algorithm, the hybrid stochastic gradient descent (HSGD) enjoys merits of both stochastic and full gradient methods for finite-sum problem optimization. However, the existing rate-of-convergence analysis for HSGD is made under with-replacement sampling (WRS) and is restricted to convex problems. It is not clear whether HSGD still carries these advantages under the common practice of without-replacement sampling (WoRS) for non-convex problems. In this paper, we affirmatively answer this open question by showing that under WoRS and for both convex and non-convex problems, it is still possible for HSGD (with constant step-size) to match full gradient descent in rate of convergence, while maintaining comparable sample-size-independent incremental first-order oracle complexity to stochastic gradient descent. For a special class of finite-sum problems with linear prediction models, our convergence results can be further improved in some cases. Extensive numerical results confirm our theoretical affirmation and demonstrate the favorable efficiency of WoRS-based HSGD.

1 Introduction

We consider the following finite-sum minimization problem:

$$\min_{x \in \mathcal{X}} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

where each individual $f_i(x)$ is $\ell$-smooth and the feasible set $\mathcal{X} \subseteq \mathbb{R}^d$ is convex. In the field of machine learning, formulation (1) encapsulates a large body of optimization problems including least square regression, logistic regression and deep neural networks training, to name a few. Such a problem can be solved by various algorithms, e.g. full gradient descent (FGD) [1], stochastic GD (SGD) [2], hybrid SGD [3], SDCA [4] and SVRG [5].

In this paper, we are particularly interested in Hybrid SGD (HSGD) [3, 6, 7] which is an inexact gradient method that iteratively samples an evolving mini-batch of the terms in (1) for gradient estimation. The iteration of HSGD is given by

$$x^{k+1} = \Phi_{\mathcal{X}}(x^k - \eta_k g^k), \text{ with } g^k = \frac{1}{s_k} \sum_{i \in S_k} \nabla f_i(x^k),$$

where $\Phi_{\mathcal{X}}(\cdot)$ denotes the Euclidean projection onto $\mathcal{X}$, $\eta_k$ is the learning rate, and $S_k$ denotes the set of the $s_k$ selected samples at the $k$-th iteration. In early iterations, HSGD selects a few samples to compute the full gradient approximately; and along with more iterations, $s_k$ is increased gradually, leading to more accurate full gradient estimation. Such a mechanism allows HSGD to simultaneously enjoy the merits of both SGD and FGD, i.e. rapid initial process of SGD and constant learning rate $\eta_k$ without sacrificing the convergence rate of FGD [6].

Motivation. Though HSGD has been shown, both in theory and practice, to bridge smoothly the gap between full and stochastic gradient descent methods, its rate-of-convergence analysis remains restrictive in several aspects.

First, the convergence behavior of HSGD under without-replacement sampling (WoRS) is not clear. In the existing analysis [6], the stochastic gradient is assumed to be computed under with-replacement sampling (WRS). But for stochastic optimization, it is a more common practice to use WoRS, i.e., to pass the loss functions \( f_i(x) \) sequentially, after random shuffling, without revisiting any of them [8, 9]. This makes significant discrepancy between the theoretical guarantee and practical implementation. As shown in Figure 1 (a), WoRS tends to provide better performance than WRS in actual implementation.

Second, the convergence behavior of HSGD for non-convex problems is not clear. Prior convergence guarantees on HSGD are limited to convex problems. Bertsekas [3] established linear convergence of HSGD for least square problems. Friedlander et al. [6] proved that HSGD converges linearly for strongly convex problems with exponentially increasing \( s_k \), and sub-linearly for arbitrary convex problems with polynomially increasing \( s_k \). Unfortunately, non-convex convergence guarantee on HSGD is still absent, though highly desirable in machine learning applications and extensively studied in other stochastic algorithms, e.g. SVRG [10, 11]. In Figure 1 (b), HSGD has sharper convergence behavior than several state-of-the-art SGD methods in training neural networks.

Third, the Incremental First-order Oracle (IFO) complexity (i.e. stochastic gradient computation; see Definition 2) of HSGD is largely left unknown. Although Friedlander et al. [6] showed that HSGD maintains steady convergence rates of FGD, its IFO complexity is not explicitly analyzed, making it less clear where HSGD should be positioned w.r.t. existing stochastic gradient approaches in overall computational complexity.

**Summary of contributions.** In this work, we address the aforementioned three limitations in the existing analysis of HSGD. We analyze the rate-of-convergence of HSGD under WoRS in a wide problem spectrum including strongly convex, non-strongly convex and non-convex problems. Table 1 summarizes our main results on IFO complexity of HSGD (WoRS) and compares them against state-of-the-art WoRS-oriented results for (stochastic) gradient methods. These results are divided into two groups: for general problems and for a special class of problems with linear prediction loss \( f_i(x) = h(a_i^T x) \). As shown in the bottom row of Table 1, we contribute several new theoretical insights into HSGD, which are elaborated in the following paragraphs.

The bounds highlighted in **green**: For both general and certain specially structured strongly convex problems, HSGD is \( n \times \) faster than FGD. Compared to the results for SAGA and AVRG [12], the IFO complexity of HSGD is not relying on the sample size \( n \) but dependent on \( 1/\epsilon \). This suggests that HSGD will converge faster when \( n \) dominates \( 1/\epsilon \). Finally, compared to the results for SGD in linear prediction problems [13], ours has removed the dependency on the logarithm term \( \log(\kappa/\epsilon) \).

The bounds highlighted in **red**: To our best knowledge, for the first time these new results establish guarantees on WoRS-based stochastic approaches for non-strongly convex and non-convex problems.

The bounds highlighted in **blue**: If the loss function \( h(a_i^T x) \) in the linearly structured problem is strongly convex in terms of \( a_i^T x \) (but \( f(x) \) may still be non-strongly convex), HSGD has \( O(1/\epsilon) \) IFO complexity. The least square regression and logistic regression (with a bounded feasible set) models have such a linear prediction structure.

The bounds highlighted in **brown**: When the specially structured problem is non-strongly convex, HSGD converges to the minimum of problem (I), while SGD can only be shown to converge to a sub-optimum up to some statistical error (see footnote 2 below Table 1).

**Related work.** Understanding randomized algorithms under WoRS and random reshuffling is gaining considerable attention in recent years. By focusing on least squares problems, Recht et al. [14] utilized arithmetic-mean inequality on matrices to show that for randomized algorithms, WoRS is always faster than WRS if the data are randomly generated from a certain distribution. For more general
1) For the first time, we provide WoRS based theoretical analysis for HSGD. 2) Our analysis covers
Ying et al.

We first introduce the concepts of strong convexity and Lipschtiz smoothness which are commonly
Assumption 1 (Bounded gradient). For each loss $f_i(x)$, the distance between its gradient $\nabla f_i(x)$ and
the full gradient $\nabla f(x)$ is upper bounded as $\max_i \|\nabla f_i(x) - \nabla f(x)\|_2 \leq G$.

If $f_i(x)$ is $\ell$-smooth and the domain of interest $\mathcal{X}$ is bounded, then the bounded gradient assumption
can be naturally implied. We explicitly write out this assumption for the sake of notation simplicity.
Following [5, 20, 21], we also employ the incremental first order oracle (IFO) complexity as the
computational complexity metric for solving the finite-sum minimization problem (1).

Definition 2. An IFO takes an index $i \in [n]$ and a point $x \in \mathcal{X}$, and returns the pair $(f_i(x), \nabla f_i(x))$.

The IFO complexity can more accurately reflect the overall computational performance of a first-order
algorithm, as objective value and gradient evaluation usually dominate the per-iteration complexity.
WRS [23]. To tackle this challenge, we introduce the following sequence of random variables:

where

Algorithm 1 Hybrid SGD under WoRS

Input: Initial point \( x^0 \), sample index set \( S = \{1, \cdots, n\} \), learning rate \( \{\eta_k\} \), mini-batch size \( \{s_k\} \).

for \( k = 0 \) to \( T - 1 \) do

Select \( s_k \) samples \( S_k \) by WoRS from \( S - \bigcup_{i=0}^{k-1} S_i \).

Compute the gradient \( g^k = \frac{1}{s_k} \sum_{i \in S_k} \nabla f_{i_k}(x^k) \).

Update \( x^{k+1} = x^k - \eta_k g^k \).

end for

Output: \( x^n \) sampled uniformly from \( \{x^k\}_{k=0}^{T-1} \) for strongly convex and linearly structured problems or \( \{x^k\}_{k=0}^{T-1} \) for non-strongly/non-convex problems.

3 General Analysis for HSGD under WoRS

The WoRS-based HSGD algorithm is outlined in Algorithm 1. Here we systematically analyze its convergence performance for strongly/non-strongly convex and non-convex problems. Similar to [13], we focus our analysis on the scenario where a single pass (or less) over data is of interest, which occurs, e.g. in streaming data analysis. According to our empirical study (see, e.g., Figure 3), running Algorithm 1 for a single pass over data can provide satisfactory accuracy in many cases.

3.1 A key lemma

It is well understood that unbiased gradient estimation with gradually vanishing variance is important for accelerating randomized algorithms [5, 15]. This is because the increasingly more accurate estimate of full gradient allows the algorithm to move ahead with more aggressive step-size to decrease the objective value. However, for WoRS implementation, the mini-batch terms selected at each iteration are no longer statistically independent, leading to biased gradient estimate \( g^k \), i.e.

Such a biased estimate \( g^k \) brings a challenge to bounding its variance \( \mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] \) with common techniques such as Bernstein inequality [22] and those existing bounds on \( \mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] \) under WRS [23]. To tackle this challenge, we introduce the following sequence of random variables:

where \( \bar{\mu}_k := \frac{1}{n} \sum_{i \in S_k} \nabla f_{i_k}(x^k) \), \( S'_k = S - \bigcup_{i=0}^{k-1} S_i \) and \( s'_k = n - \sum_{i=0}^{k-1} s_i \) in which \( S = \{1, 2, \cdots, n\} \) denotes the index set of all samples. We can prove \( z_k \)'s form a martingale, i.e. \( \mathbb{E}[z_k | z_{k-1}, \ldots, z_0] = z_{k-1} \). Moreover, we can show that its squared Euclidian norm is bounded by

where \( b_k = \sum_{i=0}^{k-1} s_i \). Similarly, we define a sequence of \( \bar{z}_i \) for the process of without-replacement sampling a subset \( S_i \) of size \( \bar{s}_i \) from \( S'_k \) of size \( s'_k \):

Also, we can prove that \( \bar{z}_i \) is a martingale with bounded norm:

Based on the above arguments, we formulate the \( k \)-th WoRS as a stochastic process consisting of two phases. In the first phase, we are given \( s'_k \) samples indexed by \( S'_k = S - \bigcup_{i=0}^{k-1} S_i \) after \( k - 1 \) times of WoRS over all the data. The sampling result is recorded by \( z_k \). Then, in the second phase, we sample \( s_k \) data from the remaining \( s'_k \) samples indexed by \( S'_k \) in a without-replacement fashion,
We analyze the convergence behavior of both the computed solution which corresponds to $\hat{z}_t$. Based on such a WoRS process, we have
\[
\mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] \leq 2\mathbb{E}[\|\hat{\mu}_k - \nabla f(x^k)\|_2^2 + \|g^k - \hat{\mu}_k\|_2^2]
\]
\[
= 2\mathbb{E}[\|z_k\|_2^2 | z_{k-1}, \ldots, z_0] + 2\mathbb{E}[\|\hat{z}_{s_k}\|_2^2 | \hat{z}_{s_k}, \ldots, \hat{z}_0; z_k, \ldots, z_0]
\]
\[
\leq \frac{8G^2}{n - b_k} \left[ 1 - \frac{(n - b_k)^2 - b_k}{n(n - b_k)} \right] + \frac{8G^2}{s_k} \left[ 1 - \frac{s_k - 1}{n - b_k} \right].
\]

The above claim leads to the following Lemma 1 which is key to our WoRS-based convergence analysis in the sections to follow. Notice, the above results on gradient variance are some intermediate results for proving Lemma 1, whose full proofs are deferred to Appendix A.

**Lemma 1.** The gradient $g_k$ estimated by WoRS in Algorithm 1 satisfies $\mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] \leq \frac{24G^2}{n}$.

From Lemma 1, we find that the gradient variance $\mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2]$ in Algorithm 1 is controlled by $1/s_k$. Accordingly, the estimated gradient becomes increasingly more accurate and stable. This means that by gradually increasing the mini-batch size, HSGD under WoRS can reduce variance, similar to SVRG and SAGA, but without requiring to integrate historical gradients or full gradient of the snapshot point into current gradient estimate. In the following sections, we will extensively use Lemma 1 to analyze HSGD under WoRS.

By applying Bernstein inequality, Friedlander et al. [6] showed that $\mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] = \mathcal{O}\left(\frac{n - s_k}{n s_k}\right)$ if the $s_k$ samples selected at iteration $k$ are different, but are sampled from the entire data set. In contrast, our considered WoRS strategy assumes the $s_k$ different samples are drawn from the remaining set $S - \bigcup_{i=1}^{t-1} S_i$, and thus needs to take into account the statistical dependence among iterations to bound the stochastic gradient variance.

### 3.2 Strongly convex functions

We analyze the convergence behavior of both the computed solution $x$ and the objective $f(x)$ under the strongly convex setting. Our convergence result on the computed solution is stated in Theorem 1.

**Theorem 1.** Suppose $f(x)$ is $\rho$-strongly-convex and each $f_i(x)$ is $\ell$-smooth. With learning rate $\eta_k = \frac{\tau}{k}$ and mini-batch size $s_k = \frac{\tau}{\ell^2}$ where $\tau = 1 - \frac{\rho^2}{18\ell^2}$ and $\tau \geq \frac{6G^2}{\rho^2} \max\{\frac{324}{\rho^2}, \frac{412}{\rho^2}\}$, we have
\[
\mathbb{E}[\|x^0 - x^*\|_2^2] \leq \left(1 - \frac{\rho^2}{18\ell^2}\right)^{T} \|x^0 - x^*\|_2^2,
\]
where $x^0$ is the output solution of Algorithm 1 and $T$ is the number of iterations.

A proof of this result is given in Appendix B.1. From Theorem 1, if mini-batch size is increased at an exponential rate $\frac{1}{1-\gamma}$ with $\gamma = \frac{\rho^2}{18\ell^2}$, then the objective in HSGD converges linearly at the rate of $\mathcal{O}\left((1 - \gamma)^k\right)$ for strongly convex problems. This implies that HSGD enjoys the merits of both SGD and FGD. Specifically, similar to SGD, the per-iteration computation of HSGD is cheap as it is free of computing the full gradient $\nabla f(x)$. Meanwhile, it uses a constant learning rate and enjoys the steady convergence rate of SGD. As the condition number $\kappa = \ell/\rho$ is usually large in realistic problems, the exponential rate $\frac{1}{1-\gamma}$ is actually only slightly above one. This means even a moderate-scale dataset allows plenty of HSGD iterations in one epoch to decrease the objective value sufficiently, as illustrated in Figure 2 and 3. Friedlander et al. [6] proved that HSGD has linear convergence rate under WRS. Theorem 1 generalizes the result to WoRS. Then we can derive the IFO complexity of HSGD for strongly-convex problems in the following corollary, for which proof is given in Appendix B.2.

**Corollary 1.** Suppose the assumptions in Theorem 1 hold. To achieve $\mathbb{E}[\|x^0 - x^*\|_2^2] \leq \epsilon$, the IFO complexity of HSGD is $\mathcal{O}\left(\frac{\kappa^2 G^2}{\epsilon}\right)$ where $\kappa = \frac{\ell}{\rho}$ denotes the condition number of the objective $f(x)$.

From Corollary 1, the IFO complexity of HSGD for strongly convex problems is at the order of $\mathcal{O}(\frac{\kappa^2}{\epsilon})$, which is not relying on the sample size $n$. So when $n$ dominates $\frac{1}{\ell}$, HSGD can be superior to the algorithms with complexity linearly relying on $n$, such as SVRG and SAGA.

Gürbüzbalaban et al. [9] showed that by processing each individual $f_i(x)$ with random shuffling at each iteration and adopting a diminishing learning rate $\eta_k = \mathcal{O}\left(\frac{1}{k^\beta}\right)$ with $\beta \in (\frac{1}{2}, 1)$, the IFO
complexity of FGD is $\mathcal{O}\left(\frac{\kappa^2}{2} \right)$ for achieving $\mathbb{E}\|x^n - x^*\|^2 \leq \epsilon$. So HSGD is $n$ times faster than FGD. This is because at each iteration, unlike FGD requiring to access all data, HSGD only samples a mini-batch for gradient estimation without sacrificing convergence rate. Ying et al. [12] proved that under WoRS, both SAGA and AVRNG converge linearly and have IFO complexity of $\mathcal{O}\left(\frac{n\kappa^2 \log \left(\frac{1}{\epsilon}\right)}{2}\right)$. Hence, HSGD will outperform SAGA and AVRNG if $n$ dominates $\frac{1}{2}$, which is usually the case when the data scale is huge while the desired accuracy $\epsilon$ is moderately small ($e.g.$ $1/\sqrt{n}$).

Shamir [13] proved that for linearly structured problems, SGD under WoRS has IFO complexity $\mathcal{O}\left(\frac{\kappa^2 \log \left(\frac{1}{\epsilon}\right)}{2}\right)$ by measuring the objective (see Section 4). Here we can also establish the shaper convergence behavior of the objective value. The result is presented in Theorem 2 with proof provided in Appendix B.3.

**Theorem 2.** Assume $f(x)$ is $\rho$-strongly-convex and each $f_i(x)$ is $\ell$-smooth. Let learning rate $\eta_k = \frac{1}{k}$ and mini-batch size $s_k = \frac{\tau}{\eta_k}$ with $\tau = 1 - \frac{\rho}{2\ell}$ and $\tau \geq \frac{6G^2}{\rho |f(x^0) - f(x^*)|}$. Then the output $x^n$ of Algorithm 1 satisfies

$$
\mathbb{E}\left[ f(x^n) - f(x^*) \right] \leq \left(1 - \frac{\rho}{2\ell}\right)^k (f(x^0) - f(x^*)).
$$

Moreover, to achieve $\mathbb{E}[f(x^n) - f(x^*)] \leq \epsilon$, the IFO complexity of HSGD is $\mathcal{O}\left(\frac{\kappa^2 \log \left(\frac{1}{\epsilon}\right)}{2}\right)$, where $\kappa = \frac{\kappa}{\rho}$.

Theorem 2 shows that HSGD also enjoys linear convergence rate on the objective by using exponentially mini-batch size. But it has lower complexity $\mathcal{O}\left(\frac{\kappa^2 \log \left(\frac{1}{\epsilon}\right)}{2}\right)$ under the measurement $\mathbb{E}[f(x^n) - f(x^*)] \leq \epsilon$ which is in contrast to the complexity $\mathcal{O}\left(\frac{\kappa^2 \log \left(\frac{1}{\epsilon}\right)}{2}\right)$ for achieving $\mathbb{E}\|x^n - x^*\|^2 \leq \epsilon$. This is because the objective analysis allows to use more aggressive step-size $\frac{1}{k}$, while the analysis on the solution requires smaller learning rate $\frac{1}{\ell}$. In this way, HSGD with larger step-size converges faster.

### 3.3 Non-strongly convex functions

We proceed to analyze the convergence performance of HSGD for non-strongly convex problems. Our result for this case is summarized in Theorem 3. To our best knowledge, this is the first convergence guarantee of WoRS-based methods for non-strongly convex problems.

**Theorem 3.** Suppose $f(x)$ is convex and each $f_i(x)$ is $\ell$-smooth. Assume that $\|x_1 - x_2\|_2 \leq D$ holds for $\forall x_1, x_2 \in X$. Then with the learning rate $\eta_k = \frac{1}{k\ell^2}$ and mini-batch size $s_k = (k + 1)^2$, we have

$$
\mathbb{E}[f(x^n) - f(x^*)] \leq \frac{4\ell D^2 + 24GD}{T} + \frac{48G^2}{\ell T^2},
$$

where $x^n$ denotes the output solution of Algorithm 1 and $T$ is the number of iterations.

A proof of this result is given in Appendix B.4. Theorem 3 shows that if one expands the mini-batch size at $\mathcal{O}\left(\kappa^2\right)$, then the convergence rate of HSGD under WoRS is $\mathcal{O}\left(\frac{1}{k}\right)$. In [13], a sub-linear rate was established for WoRS-based SGD in a special class of convex problems with $f_i(x) = h_i((\alpha, x))$. A detailed comparison between their result and ours for such a structured formulation will be discussed in Section 4. Under the assumption $\sum_{k=0}^{\infty} \|g^k - \nabla f(x^k)\|_2 < +\infty$, Friedlander et al. [6] showed that WRS-based HSGD outputs $f(x^n) - f(x^*) = \mathcal{O}\left(\frac{1}{k}\right)$. However, such an assumption holds only if HSGD selects at least $\mathcal{O}\left(\kappa^2\right)$ samples at the $k$-th iteration due to $\mathbb{E}[\|g^k - \nabla f(x^k)\|_2^2] = \mathcal{O}\left(\frac{n - n_k}{n \kappa^2}\right)$. In this way, their result under WRS is of the same order as ours under WoRS. The following corollary gives the corresponding IFO complexity. A proof of this result is given in Appendix B.5.

**Corollary 2.** Suppose the assumptions in Theorem 3 hold. To achieve $\mathbb{E}[f(x^n) - f(x^*)] \leq \epsilon$, the IFO complexity of HSGD is $\mathcal{O}\left(\frac{(6GD + 4D^2)^3}{\epsilon^3}\right)$.

### 3.4 Non-convex functions

Now we analyze HSGD for non-convex problems, which to our knowledge has not yet been addressed elsewhere in literature. The result is stated in Theorem 4 with proof provided in Appendix B.6.

**Theorem 4.** Suppose each $f_i(x)$ is $\ell$-smooth and for $\forall x_1, x_2 \in X$, $\|x_1 - x_2\|_2 \leq D$. With learning rate $\eta_k = \frac{1}{k\ell^2}$ and mini-batch size $s_k = k + 1$, the output $x^n$ of Algorithm 1 with $T$ iterations satisfies

$$
\mathbb{E}\left[ \|\nabla f(x^n)\|^2 \right] \leq \frac{4\ell^2 D^2 + 35G^2}{T}.
$$

6
Theorem 4 guarantees that for non-convex problems, HSGD exhibits $O\left(\frac{1}{t^2}\right)$ rate of convergence by linearly expanding the mini-batch size at each iteration. Here we follow the convention in [10, 11, 23] to adopt the value $\|\nabla f(x^a)\|^2_2$ as a measurement of quality for approximate stationary solutions. Then we drive the IFO complexity of HSGD in the following corollary with proof in Appendix B.7.

**Corollary 3.** Suppose the assumptions in Theorem 4 hold. To achieve $\mathbb{E}\left[\|\nabla f(x^a)\|^2_2\right] \leq \epsilon$, the IFO complexity of the HSGD in Algorithm 1 is $O\left(\frac{4\ell^2D^2 + 35G^2}{\epsilon^2}\right)$.

The IFO complexity for non-convex problems looks lower than that for non-strongly convex ones in Corollary 2. This is because we use $\mathbb{E}\left[|f(x^a) - f(x^*)|\right] \leq \epsilon$ as sub-optimality measurement for arbitrary convex problems and $\mathbb{E}\left[\|\nabla f(x^a)\|^2_2\right] \leq \epsilon$ for non-convex problems.

### 4 Analysis for Linearly Structured Problems

We further consider a special case of problem (1) where each $f_i(x)$ has a linear prediction structure:

$$f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \text{ where } f_i(x) = h((a_i, x)). \quad (2)$$

Here $a_i$ denotes the $i$-th sample vector and $h(\cdot)$ denotes a convex loss function. Such a formulation covers several common problems in machine learning, such as $f_i(x) = \frac{1}{2}(b_i - a_i^T x)^2$ for least square regression and $f_i(x) = \log(1 + \exp(-b_i a_i^T x))$ for logistic regression, where $b_i$ is the real or binary target output. Such a special problem setting has been considered in [13] for analyzing SGD under WoRS. To make a comprehensive comparison, we specify our strongly convex analysis to (2), and improve our non-strongly-convex results when the surrogate loss $h(\cdot)$ is strongly convex.

**Strongly convex case.** In this case, according to Theorem 2, HSGD converges linearly and its IFO complexity is $O\left(\frac{1}{\epsilon}\right)$. By comparison, SGD under WoRS in [13] converges at $O\left(\frac{1}{\epsilon} \log \left(\frac{1}{\epsilon}\right)\right)$ and has IFO complexity $O\left(\frac{1}{\epsilon} \log \left(\frac{1}{\epsilon}\right)\right)$, slightly higher than ours due to the presence of the factor $\log \left(\frac{1}{\epsilon}\right)$. Moreover, it is allowed in HSGD to use constant step-size which is required to be shrinking in [13].

On this special problem, other results on general strongly convex problems can also be applied. As discussed in Section 3.2, HSGD is $n$ times faster than FGD [9], and is superior to SAGA [12] and AVRG [12] when $n$ dominates $\frac{1}{2}$. Shamir [13] showed that SVRG [5] under WoRS has IFO complexity $O\left((n + \kappa \log \left(\frac{1}{\epsilon}\right)) \log \left(\frac{1}{\epsilon}\right)\right)$ in ridge regression with the measurement $\mathbb{E}[f(x) - f(x^*)]$. Comparatively, such an IFO complexity is still higher than HSGD when sample size $n$ is large and the desired accuracy is moderately small.

**Non-strongly convex case with strongly-convex $h(\cdot)$.** When the loss $f(x)$ in (2) is non-strongly convex but the surrogate loss $h(\cdot)$ is strongly convex, we show an improved convergence rate in Theorem 5 than that in Theorem 3 for general cases. See proof of Theorem 5 in Appendix C.1.

**Theorem 5.** Suppose $f_i(x) = h(a_i^T x)$ is $\ell$-smooth and $h(\cdot)$ is $\alpha$-strongly convex. Let $\sigma(A)$ denote the smallest non-zero singular value of the matrix $A = [a_1^T : a_2^T : \ldots : a_n^T]$ and $\mu = \sigma(A)$. If the learning rate $\eta_k = \frac{1}{l}$ and mini-batch size $s_k = \frac{\tau}{\epsilon}$ with $\tau \geq \frac{24G^2}{\mu |f(x^a) - f(x^*)|}$ and $\zeta = 1 - \frac{\mu}{2l}$, we have

$$\mathbb{E}\left[|f(x^a) - f(x^*)|\right] \leq (1 - \frac{\mu}{2l})^T (f(x^0) - f(x^*)), \quad (2)$$

where $x^a$ denotes the output solution of Algorithm 1 and $T$ is the number of iterations.

Theorem 5 shows if the function $h(a_i^T x)$ is strongly convex in terms of the linear prediction $a_i^T x$, by exponentially sampling the data at each iteration, HSGD converges linearly even though $f(x)$ might be non-strongly convex. Based on Theorem 5, we further derive the IFO complexity of Algorithm 1 for such a special problem, as summarized in Corollary 4 with proof in Appendix C.2.

**Corollary 4.** Suppose the assumptions in Theorem 5 hold. To achieve $\mathbb{E}[|f(x^a) - f(x^*)|] \leq \epsilon$ for the special problem, the IFO complexity of the proposed algorithm is $O\left(\frac{12G^2}{\mu \epsilon^2}\right)$.

It is interesting to compare Theorem 5 and Corollary 4 with those existing ones for SGD. Particularly, it was shown by Shamir [13] that $\mathbb{E}[|f(x^a) - f(x^*)|] \leq R_T/T + 2(12 + \sqrt{D}/\sqrt{T})$ for SGD, where $R_T$ is the regret bound of SGD on problem (2), at the order of $O(D\ell\sqrt{T})$. This gives a
convergence rate of $O(1/\sqrt{T})$ and IFO complexity of $O(1/\epsilon^2)$. However, there exists an accuracy barrier $O(1/\sqrt{n})$ due to the statistical error term $2(12 + \sqrt{2D})/\sqrt{n}$ which is the artifact brought by analyzing the regret. In sharp contrast, our result in Theorem 5 guarantees that HSGD converges to the global optimum of problem (2). More importantly, provided that $h(\cdot)$ is strongly convex, HSGD has superior IFO complexity of $O\left(\frac{1}{\epsilon^2}\right)$ to the SGD complexity $O\left(\frac{1}{\epsilon}\right)$ given in [13].

5 Experiments

We compare HSGD with several state-of-the-art algorithms, including SGD [2], SVRG [5], SAGA [15], AVRG [12] and SCGC [23], under WoRS for all. We consider two sets of learning tasks. The first contains two convex problems: $\ell_2$-regularized logistic regression $\min_x \frac{1}{n} \sum_{i=1}^{n} \left[ \log(1 + \exp(-b_i a_i^\top x)) + \frac{\lambda}{2} \|x\|^2 \right]$ and $k$-classes softmax regression $\min_x \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \left[ \frac{1}{2k} \|x_j\|^2 - \log \left( \sum_{l=1}^{k} \exp(a_i^\top x_j) \right) \right]$, where $b_i$ is the target output of $a_i$.

The other one is a non-convex problem of training multi-layer neural networks. We run simulations on 10 datasets (see Appendix D). Hyper-parameters of all the algorithms are tuned to best.

5.1 Convex problems

As the first set of problems are strongly convex, we follow Theorem 2 to exponentially expand the mini-batch size $s_k$ in HSGD with $\tau = 1$. We run FGD until the gradient $\|\nabla f(x)\|_2 \leq 10^{-10}$. Then use the output as the optimal value $f^*$ for sub-optimality estimation in Figure 1 (a), 2 and 3.

**Single-epoch processing in well-conditioned problems.** We first consider the case where the optimization problem is well-conditioned with strong regularization, such that good results can be obtained after only one epoch of data pass. Single-epoch learning is common in online learning. For two problems, we respectively set their regularization parameters to $\lambda = 0.01$ and $\lambda = 0.1$.

Figure 2 summarizes the numerical results. On the simulated well-conditioned tasks most algorithms achieve high accuracy after one epoch, while HSGD (WoRS) converges much faster. This confirms Corollary 1 that HSGD is cheaper in IFO complexity ($O\left(\frac{1}{\epsilon^2}\right)$) than other considered variance-reduced algorithms ($O\left(\frac{nk^2 \log(\frac{1}{\epsilon})}{\epsilon}\right)$) when the desired accuracy is moderately low and data size is large.

**Multi-epoch processing in ill-conditioned problems.** To solve more challenging problems, a method usually needs multiple cycles of data processing to reach high accuracy solution. Thus we develop a practical implementation of HSGD for multiple epochs processing. After visiting all data in one full pass, it continues to increase the mini-batch size, allowing possible with-replacement sampling, until $s_k > n$. After that, HSGD degenerates to standard FGD. But this does not happen in
our testing cases, since we set the exponential rate sufficiently small. To generate more challenging optimization tasks, we reset the regularization strength parameter in softmax regression as \( \lambda = 0.001 \).

Figure 3 shows that HSGD under WoRS outperforms all compared algorithms. These observations align well with those in Figure 2, implying HSGD has sharper convergence behavior when the sample size \( n \) is large and the desired accuracy is moderate. The convergence curves of HSGD also confirm the effectiveness of our practical implementation in continuously decreasing the objective value.

\[
\begin{align*}
\text{Figure 3: Multi-epoch processing: comparison of randomized algorithms for multiple passes over data (Softmax regression. From left to right: protein, satimage, sensorless and letter).}
\end{align*}
\]

5.2 Non-convex problems

Here we evaluate HSGD for optimizing a three-layer feedforward neural network with a logistic loss on \( \text{ijcnn1} \) and \( \text{covtype} \) and softmax loss on \( \text{sensorless} \) (see Figure 1 (b)). For both cases we set \( \lambda = 0.01 \). The network has an architecture of \( d - 30 - c \), where \( d \) and \( c \) respectively denote the input and output dimension and 30 is the neuron number in the hidden layer. We test two versions of HSGD, namely HSGD-lin and HSGD-exp, respectively with linearly and exponentially increasing mini-batch size from \( s_0 = 1 \). We use the same initialization for all algorithms.

From Figure 4, HSGD-exp exhibits similar convergence behavior as above: it decreases the loss very quickly. Comparatively, HSGD-lin outputs more accurate solutions with linearly increasing batch size, which is consistent with Theorem 4. We note HSGD-lin behaves differently in Figure 4 (a) and (b). In Figure 4 (a), it converges relatively slowly at the beginning, while in Figure 4 (b) much faster, of which we attribute the reason to the different characteristics of data.

\[
\begin{align*}
\text{Figure 4: Non-convex results: comparison of randomized algorithms on forward neural networks.}
\end{align*}
\]

6 Conclusion

We analyzed the rate-of-convergence of HSGD under WoRS for strongly/arbitrarily convex and non-convex problems. We proved that under WoRS, HSGD with constant step-size can match FG descent in convergence rate, while maintaining comparable sample-size-independent IFO complexity to SGD. Compared to the variance-reduced SGD methods such as SVRG and SAGA, HSGD tends to gain better efficiency and scalability in the setting where the sample size is large while the required optimization accuracy is moderately small. Numerical results confirmed our theoretical results.

Acknowledgements

Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112, NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112. Xiao-Tong Yuan was supported in part by Natural Science Foundation of China (NSFC) under Grant 61522308 and Grant 61876090, and in part by Tencent AI Lab Rhino-Bird Joint Research Program No.JR201801.
References