Probabilistic Variance-Reduced Shuffling Gradient Descent Algorithm for Nonconvex Optimization

Xia Jiang[1], Xianlin Zeng[2], Anthony Man-Cho So[1]

1. Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, 999077, Hong Kong
E-mail: xiajiang@cuhk.edu.hk, manchoso@se.cuhk.edu.hk

2. National Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, P. R. China
E-mail: xianlin.zeng@bit.edu.cn

Abstract: Nonconvex finite-sum optimization finds wide applications in various signal processing and machine learning tasks. The well-known stochastic gradient algorithms generate unbiased stochastic gradient estimates by taking the uniform sampling mechanism with replacement, which can be difficult to implement in practice. This paper explores a sampling-without-replacement shuffling scheme to generate stochastic gradients. Additionally, to handle the variance of gradients, this paper develops a novel variance reduction step in the shuffling gradient descent algorithm. Specifically, this paper proposes a probabilistic variance-reduced shuffling gradient descent algorithm, which reduces the number of gradient oracle calls in the variance reduction step. The proposed algorithm owns a better convergence rate compared to existing stochastic gradient algorithms for nonconvex optimization. Finally, numerical results are presented to demonstrate the efficiency of the proposed algorithm.

Key Words: Random reshuffling, Variance reduction, Bernoulli distribution, Nonconvex optimization

1 Introduction

The nonconvex finite-sum optimization has wide applications in signal processing [1], cooperative control [2], and neural network training [3]. Various first-order stochastic gradient methods become leading algorithms for large-scale nonconvex optimization due to their scalability and low computational requirements. Stochastic algorithms can be roughly divided into sampling-with-replacement stochastic algorithms and sampling-without-replacement stochastic algorithms. Most existing stochastic gradient descent (SGD) algorithms [4, 5] are sampling-with-replacement algorithms, where the data are sampled uniformly to generate unbiased gradient estimates. These sampling-with-replacement algorithms suffer from the drawback of potentially missing out on learning certain data samples. In contrast, the sampling-without-replacement shuffling gradient algorithms, which are widely adopted in PyTorch and TensorFlow, learn from each data and generally achieve better performance than SGD algorithms [6, 7].

Despite the outstanding practical performance, the sampling-without-replacement scheme in shuffling gradient algorithms introduces a significant complication that the gradients are now biased, which implies that a single iteration does not approximate a full gradient descent step. This poses challenges in providing theoretical convergence guarantees, especially for nonconvex optimization. The recent works [6, 8, 9] provided inspiring convergence analyses of the superiority of shuffling gradient algorithms over SGD for (strongly) convex optimization, despite the presence of biased gradients. For nonconvex optimization, the work [8] has established that the shuffling gradient algorithm achieves a sublinear convergence rate with a constant error term proportional to gradient variances. When employing diminishing step sizes, the work [10] showed that shuffling gradient algorithms can converge without any constant error term. Moreover, the work [11] established a strong limit-point convergence rate for random reshuffling gradient algorithms with appropriate diminishing step sizes under the well-known Kurdyka–Łojasiewicz inequality [12]. These works indicate that existing shuffling gradient algorithms with constant step sizes only converge to a neighborhood of the stationary points of nonconvex optimization due to the stochastic gradient variance.

Facing this challenge, numerous variance-reduced algorithms have emerged to tackle the gradient variance introduced by random sampling steps in stochastic algorithms. These variance-reduced algorithms [13–16] construct an iterative gradient estimator, whose variance diminishes over time. For (strongly) convex optimization, the work [15] developed an explicit variance reduction technique for SGD, known as Stochastic Variance Reduced Gradient (SVRG), which mitigates the inherent variance of SGD. Building upon [15], the work [16] introduced the Stochastic Recursive Gradient Algorithm (SARAH), achieving a faster convergence rate than SVRG while also eliminating the requirement for storing gradients as in [14]. In nonconvex optimization, existing works [17, 18] made significant contributions to establish non-asymptotic convergence rates of SVRG and its variants, which are faster than SGD and gradient descent. Further advancements were made in [19] to propose a Stochastic Path-Integrated Differential Estimator (SPIDER) algorithm, which achieves a faster convergence rate than [17, 18] for nonconvex optimization.

While there are various variance reduction techniques available for sampling-with-replacement SGD algorithms, there has been limited research focusing on variance reduction techniques tailored for sampling-without-replacement shuffling gradient algorithms. For strongly convex optimization, the work [9] pioneered the investigation of the shuffling gradient algorithms with variance reduction, developing a novel amortized variance-reduced gradient descent algorithm. Following this, the work [20] utilized the SVRG technique to develop variance-reduced variants of shuffling gra-

This work is supported in part by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) project CUHK 14204823, and NSFC under Grants 62222303 and 62088101.

dient algorithms, improving the convergence rates for both convex and nonconvex optimization. In the case of nonconvex optimization, the recent work [21] further established a limit-point convergence result for the variance-reduced random reshuffling algorithm under the Kurdyka–Łojasiewicz condition. However, in both [20] and [21], the variance reduction steps require calculating the full gradient once per epoch, which may not be efficient for large-scale problems. Motivated by these existing works and the excellent practical performance of sampling-without-replacement algorithms, this paper explores an efficient probabilistic variance reduction technique for shuffling gradient algorithms to solve nonconvex optimization. Our contributions are summarized as follows.

- This paper develops a probabilistic variance-reduced stochastic gradient descent algorithm under the sampling-without-replacement scheme for smooth nonconvex finite-sum optimization. The proposed variance reduction step employs a Bernoulli distribution, which decreases the number of full gradient oracle calls in comparison to the works [20, 21]. This work extends the existing variance-reduced SGD algorithms [15, 17, 18] to the sampling-without-replacement shuffling scheme for nonconvex optimization.
- In contrast to existing works that necessitate an assumption of bounded gradient variance, a condition that may only sometimes hold in practical scenarios [22], this paper provides theoretical convergence results of the proposed stochastic algorithm for nonconvex optimization only under the smooth assumption. Compared with the shuffling gradient algorithms in [8, 10], the variance caused by stochastic gradients is eliminated in the convergence result of the proposed algorithm without requiring diminishing step sizes. This is attributed to the carefully designed variance reduction step in the proposed algorithm.

The remainder of the paper is organized as follows. The problem description and variance-reduced shuffling gradient algorithm are developed in section 2. The convergence analysis of the proposed algorithm is provided in section 3. The efficiency of the proposed algorithm is verified by simulations in Section 4 and the conclusion is made in section 5.

1.1 Notation

We denote by $\mathbb{R}$ the set of real numbers and $\mathbb{R}^n$ the set of n-dimensional real column vectors. The symbol $[n]$ denotes the set $\{0, \ldots, n-1\}$. For a real vector v, we denote by $\|v\|$ its the Euclidean norm. The gradient vector of a differentiable function f is represented by ∇f. In the symbol x_k^i, the superscript i indicates the ith sample, and the subscript k indicates the kth epoch.

2 Problem Description and Algorithm Design

In this paper, we study the following finite-sum nonconvex optimization problem

$$\min_{x \in \mathbb{R}^d} f(x), \ f(x) = \frac{1}{n} \sum_{i \in [n]} f_i(x), \tag{1}$$

where $x \in \mathbb{R}^d$ is the optimization variable and f_i for $i \in [n]$ is a smooth function associated with the ith sample. While numerous stochastic algorithms have been developed for convex finite-sum optimization problems represented by (1), nonconvex finite-sum optimization remains a prominent research focus.

Different from the well-studied uniform sampling-with-replacement mechanism in SGD works, we investigate the sampling-without-replacement shuffling scheme, which is widely applied in practice. In the proposed stochastic algorithm, which is summarized in Algorithm 1, data samples are shuffled to generate a new random or deterministic permutation of the index set $[n]$ at the beginning of each epoch. Then, consecutive gradient descents are performed with this shuffled permutation during the inner iterations. Moreover, to handle the variance of stochastic gradients, we generate a gradient estimator by the following variance reduction step

$$g_k^i = \nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(\tau_k) + \nabla f(\tau_k), \tag{2}$$

where τ_k is a random variable following a probability distribution. The proposed algorithm uses the gradient estimator g_k^i, instead of the stochastic gradient ∇f_i, to execute a gradient descent of the variable x_k^i. After completing the inner iterations, the variable for the next epoch x_{k+1} is set to the final variable from the previous epoch.

Algorithm 1 Probabilistic variance-reduced shuffling gradient (PVR-RG) algorithm

Initialization: Set $x_0 \in \mathbb{R}^d$, the number of epochs K.
for $k = 1, \ldots, K$ **do**
  Generate any permutation $\pi_k = (\pi_k^0, \ldots, \pi_k^{n-1})$ of $[n]$ (either deterministic or random)
  $x_k^0 = x_k$
  $\tau_k = \begin{cases} x_k & \text{with probability } p \\ \tau_{k-1} & \text{with probability } 1-p \end{cases}$
  for $i = 0, \ldots, n-1$ **do**
    $g_k^i = \nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(\tau_k) + \nabla f(\tau_k)$
    $x_k^{i+1} = x_k^i - \frac{\eta}{n} g_k^i$
  end for
  $x_{k+1} = x_k^n$
end for

The variance reduction step (2) is a modification of the well-known SVRG technique in [15]. The main difference is that τ_k in (2) is updated following the Bernoulli distribution, while in SVRG, it is updated periodically. Furthermore, SVRG employs the uniform sampling-with-replacement scheme as in the classic SGD algorithm, while the proposed PVR-RG takes a sampling-without-replacement shuffling scheme. In SVRG, the gradient estimator g_k^i is an unbiased gradient estimator [15], but this is not the case in our proposed algorithm PVR-RG. This modification is vital for the convergence of the proposed shuffling gradient algorithm for the smooth nonconvex optimization.

Remark 2.1. *Compared with the variance-reduced shuffling gradient algorithms in [20, 21], the proposed algorithm decreases the number of full gradient oracle calls at each epoch. Specifically, the computation of the full gradient on τ_k follows a Bernoulli distribution with parameter $p \in (0, 1)$, whereas in [20, 21], the probability p is set to 1. In addition, when p is zero, the proposed algorithm reduces* *to the random reshuffling algorithm with random permutation or shuffle once algorithm with deterministic permutation in [8].*

Remark 2.2. *The proposed PVR-RG algorithm exhibits strong scalability in multi-agent systems. By employing the average consensus technique as in [21], the proposed stochastic algorithm can be extended into a distributed algorithm, offering excellent privacy and robustness for solving large-scale problems. This is also one of the future directions for the proposed algorithm.*

3 Theoretical analysis

This section provides the convergence analysis of Algorithm 1 under the following standard assumption of objective function.

Assumption 3.1. *Each objective function f_i in (1) is bounded from below and is L-smooth, i.e. $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x-y\|$. It also implies that for all $x, y \in \mathbb{R}^d$,*

$$f_i(x) \leq f_i(y) + \langle \nabla f_i(y), x-y\rangle + \frac{L}{2}\|x-y\|^2. \tag{3}$$

For conciseness, the updating of each epoch in PVR-RG algorithm can be rewritten as $x_{k+1} = x_k - \eta v_k$, where $v_k = \frac{g_k}{n}$ and $g_k = \sum_{i=0}^{n-1} g_k^i$. In addition, we define the backward per-epoch deviation over the kth epoch as

$$B_k \triangleq \sum_{i\in[n]} \|x_k^i - x_k^0\|^2.$$

First, we provide a lemma to describe the change of function value before and after an epoch.

Lemma 3.1. *Suppose that Assumption 3.1 holds. For any step-size $\eta > 0$ and $k = 1, \ldots, K$, we have*

$$\begin{aligned} f\left(x_{k+1}\right) \leq & f\left(x_k\right) - \frac{\eta}{2}\left\|\nabla f(x_k)\right\|^2 \\ & - \left(\frac{1}{2\eta} - \frac{L}{2}\right)\left\|x_{k+1} - x_k\right\|^2 \\ & + \frac{\eta}{2}\left\|v_k - \nabla f\left(x_k\right)\right\|^2. \end{aligned} \tag{4}$$

Proof. Using the L-smoothness of f, we have

$$\begin{aligned} & f(x_{k+1}) \\ \leq & f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ = & f(x_k) - \eta\langle \nabla f(x_k), v_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ = & f(x_k) - \frac{\eta}{2}\left(\|\nabla f(x_k)\|^2 + \|v_k\|^2 - \|v_k - \nabla f(x_k)\|^2\right) \\ & + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ = & f(x_k) - \frac{\eta}{2}\|\nabla f(x_k)\|^2 + \frac{\eta}{2}\|v_k - \nabla f(x_k)\|^2 \\ & - \left(\frac{1}{2\eta} - \frac{L}{2}\right)\left\|x_{k+1} - x_k\right\|^2, \end{aligned}$$

where we use $v_k = \frac{x_k - x_{k+1}}{\eta}$ in the last equality. □

Then, with the backward per-epoch deviation B_k, we bound the last variance term $\|v_k - \nabla f(x_k)\|^2$ of (4) in the following lemma.

Lemma 3.2. *Suppose that Assumption 3.1 holds. The term $\|v_k - \nabla f(x_k)\|^2$ satisfies*

$$\|v_k - \nabla f(x_k)\|^2 \leq \frac{L^2}{n} B_k.$$

Proof. It follows from (2) in the proposed algorithm that

$$\begin{aligned} & \|v_k - \nabla f(x_k)\|^2 \\ = & \|\frac{1}{n}\sum_{i=0}^{n-1} g_k^i - \nabla f(x_k)\|^2 \\ = & \left\|\frac{1}{n}\sum_{i=0}^{n-1}\left(\nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(\tau_k)\right) + \nabla f(\tau_k) - \nabla f(x_k)\right\|^2 \\ = & \left\|\frac{1}{n}\sum_{i=0}^{n-1}\left(\nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(x_k)\right)\right\|^2 \\ \leq & \frac{L^2}{n}\sum_{i=0}^{n-1}\|x_k^i - x_k\|^2 = \frac{L^2}{n} B_k, \end{aligned}$$

where the last inequality holds by Assumption 3.1. □

Further, the following lemma provides the upper bound of the backward per-epoch deviation B_k.

Lemma 3.3. *Suppose that Assumption 3.1 holds. If the step-size η satisfies $0 < \eta < \frac{1}{L\sqrt{2(2-p)}}$ and $p \in (0,1)$, then the backward per-epoch deviation B_k satisfies*

$$B_k \leq \mathcal{C}n(2-p)\|\nabla f(x_k)\|^2 + 4\mathcal{C}n(1-p)L^2\|x_k - \tau_k\|^2, \tag{5}$$

where $\mathcal{C} \triangleq \frac{2\eta^2}{1-2\eta^2L^2(2-p)}$.

Proof. By the proposed Algorithm 1, we have

$$\begin{aligned} & \|x_k^i - x_k^0\|^2 \\ = & \left\|\frac{\eta}{n}\sum_{j=0}^{i-1} g_k^j\right\|^2 \leq \frac{\eta^2}{n}\sum_{j=0}^{i-1}\|g_k^j\|^2 \\ = & \frac{\eta^2}{n}\sum_{j=0}^{i-1}\|\nabla f_{\pi_k^j}(x_k^j) - \nabla f_{\pi_k^j}(\tau_k) + \nabla f(\tau_k)\|^2 \\ \leq & \frac{\eta^2}{n}\sum_{j=0}^{i-1}\left(2L^2\|x_k^j - \tau_k\|^2 + 2\|\nabla f(\tau_k)\|^2\right) \\ = & \frac{\eta^2}{n}p\left(2L^2B_k + 2i\|\nabla f(x_k)\|^2\right) \\ & + \frac{\eta^2(1-p)}{n}\sum_{j=0}^{i-1}\left(2L^2\|x_k^j - \tau_k\|^2 + 2\|\nabla f(\tau_k)\|^2\right) \\ \leq & \frac{\eta^2}{n}p\left(2L^2B_k + 2i\|\nabla f(x_k)\|^2\right) \\ & + \frac{\eta^2(1-p)}{n}\sum_{j=0}^{i-1}2L^2\|x_k^j - \tau_k\|^2 \\ & + \frac{\eta^2(1-p)}{n}\sum_{j=0}^{i-1}\left(4L^2\|x_k - \tau_k\|^2 + 4\|\nabla f(x_k)\|^2\right), \end{aligned} \tag{6}$$

where the last inequality holds due to the fact that

$$\begin{aligned}\|\nabla f(\tau_k)\|^2 =&\|\nabla f(\tau_k) - \nabla f(x_k) + \nabla f(x_k)\|^2\\ \leq&2L^2\|x_k - \tau_k\|^2 + 2\|\nabla f(x_k)\|^2.\end{aligned}$$

Then, substituting the inequality $\|x_k^j - \tau_k\|^2 \leq 2\|x_k^j - x_k\|^2 + 2\|x_k - \tau_k\|^2$ into (6), we have

$$\begin{aligned}&\|x_k^i - x_k^0\|^2\\ \leq&\frac{\eta^2}{n}p(2L^2B_k + 2i\|\nabla f(x_k)\|^2)\\ &+ \frac{\eta^2(1-p)}{n}\sum_{j=0}^{i-1} 4L^2\|x_k^j - x_k\|^2\\ &+ \frac{\eta^2(1-p)}{n}\sum_{j=0}^{i-1}\left(8L^2\|x_k - \tau_k\|^2 + 4\|\nabla f(x_k)\|^2\right)\\ \leq&\frac{2\eta^2}{n}pL^2B_k + 2\eta^2p\|\nabla f(x_k)\|^2\\ &+ \frac{4\eta^2}{n}(1-p)L^2B_k + 8\eta^2(1-p)L^2\|x_k - \tau_k\|^2\\ &+ 4\eta^2(1-p)\|\nabla f(x_k)\|^2\\ =&\frac{2\eta^2}{n}L^2(2-p)B_k + 2\eta^2(2-p)\|\nabla f(x_k)\|^2\\ &+ 8\eta^2(1-p)L^2\|x_k - \tau_k\|^2.\end{aligned}$$

Summing the above inequality over $[n]$ yields

$$\begin{aligned}B_k =& \sum_{i\in[n]}\|x_k^i - x_k^0\|^2\\ \leq&2\eta^2L^2(2-p)B_k + 2n\eta^2(2-p)\|\nabla f(x_k)\|^2\\ &+ 8n\eta^2(1-p)L^2\|x_k - \tau_k\|^2.\end{aligned}$$

Since the step size satisfies $2\eta^2L^2(2-p) < 1$, we obtain

$$\begin{aligned}B_k \leq&\frac{2n\eta^2(2-p)}{1-2\eta^2L^2(2-p)}\|\nabla f(x_k)\|^2\\ &+ \frac{8n\eta^2(1-p)L^2}{1-2\eta^2L^2(2-p)}\|x_k - \tau_k\|^2.\end{aligned}$$

□

Now, we are ready to prove the convergence rate of Algorithm 1.

Theorem 3.1. *Suppose that Assumption 3.1 holds and the positive step size η satisfies*

$$\eta < \min\left\{\frac{(2p-1)(2-p)}{2L^2(1-p)}, \frac{1}{2L\sqrt{2-p}}, \frac{1}{L+4-4p}\right\}, \tag{7}$$

where $p \in (0,1)$, then

$$\frac{1}{K}\sum_{k=1}^{K}(\|\nabla f(x_k)\|^2 + \|x_{k+1} - x_k\|^2) = \mathcal{O}\left(\frac{1}{K}\right).$$

Proof. Combining the results in Lemmas 3.1, 3.2, and 3.3, we have

$$\begin{aligned}f(x_{k+1}) \leq&f(x_k) - \frac{\eta}{2}\|\nabla f(x_k)\|^2 - \frac{1-\eta L}{2\eta}\|x_{k+1} - x_k\|^2\\ &+ \frac{\eta}{2}\|v_k - \nabla f(x_k)\|^2\\ \leq&f(x_k) - \frac{\eta}{2}\|\nabla f(x_k)\|^2 - \frac{1-\eta L}{2\eta}\|x_{k+1} - x_k\|^2\\ &+ \frac{\eta}{2}\frac{L^2}{n}B_k\\ \leq&f(x_k) - \frac{1-\eta L}{2\eta}\|x_{k+1} - x_k\|^2\\ &- \left(\frac{\eta}{2} - \eta L^2\frac{\eta^2(2-p)}{1-\eta^2 2L^2(2-p)}\right)\|\nabla f(x_k)\|^2\\ &+ \frac{\eta L^2}{2n}\frac{n\eta^2(1-p)8L^2}{1-\eta^2 2L^2(2-p)}\|x_k - \tau_k\|^2.\end{aligned} \tag{8}$$

In addition, the term $\|x_k - \tau_k\|^2$ satisfies

$$\begin{aligned}&\|x_{k+1} - \tau_{k+1}\|^2\\ =&(1-p)\|x_{k+1} - x_k + x_k - \tau_k\|^2\\ \leq&2(1-p)\|x_{k+1} - x_k\|^2 + 2(1-p)\|x_k - \tau_k\|^2.\end{aligned} \tag{9}$$

Combining (8) and (9) yields

$$\begin{aligned}&f(x_{k+1}) + \|x_{k+1} - \tau_{k+1}\|^2\\ \leq&f(x_k) - \left(\frac{\eta}{2} - A(2-p)\right)\|\nabla f(x_k)\|^2\\ &- \left(\frac{1}{2\eta} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 + 2(1-p)\|x_{k+1} - x_k\|^2\\ &+ 4A(1-p)L^2\|x_k - \tau_k\|^2 + 2(1-p)\|x_k - \tau_k\|^2\\ =&f(x_k) - \left(\frac{\eta}{2} - A(2-p)\right)\|\nabla f(x_k)\|^2\\ &- \left(\frac{1}{2\eta} - \frac{L}{2} - 2(1-p)\right)\|x_{k+1} - x_k\|^2\\ &+ \left(4A(1-p)L^2 + 2(1-p)\right)\|x_k - \tau_k\|^2,\end{aligned}$$

where $A \triangleq \frac{\eta L^2}{2}\frac{2\eta^2}{1-\eta^2 2L^2(2-p)}$.

Since the step size satisfies $\eta < \frac{1}{2L\sqrt{2-p}}$, we have $A < \frac{\eta}{2(2-p)}$ and $C_g \triangleq \frac{\eta}{2} - A(2-p) > 0$. In addition, due to $4A(1-p)L^2 < \frac{\eta}{2-p}(1-p)2L^2$ and $\eta < \frac{(2p-1)(2-p)}{2L^2(1-p)}$, we have

$$C_\tau \triangleq 4A(1-p)L^2 + 2(1-p) < 1. \tag{10}$$

Finally, using $\eta < \frac{1}{L+4-4p}$, we have $C_x \triangleq \frac{1}{2\eta} - \frac{L}{2} - 2(1-p) > 0$. Then,

$$\begin{aligned}&f(x_{k+1}) + \|x_{k+1} - \tau_{k+1}\|^2\\ \leq&f(x_k) + \|x_k - \tau_k\|^2 - C_g\|\nabla f(x_k)\|^2 - C_x\|x_{k+1} - x_k\|^2.\end{aligned}$$

From Assumption 3.1, we have $f(x_{K+1}) > -\infty$. Then, by rearranging and averaging the sums on both sides over $k = \{1, \ldots, K\}$, we have

$$\frac{1}{K}\sum_{k=1}^{K}(\|\nabla f(x_k)\|^2 + \|x_{k+1} - x_k\|^2) \leq \frac{f(x_1) + \|x_1 - \tau_1\|^2}{K\cdot\max\{C_g, C_x\}}.$$

□

Corollary 3.1. *Suppose that Assumption 3.1 holds. If the step-size η satisfies* (7)*, then the proposed algorithm requires $\mathcal{O}(n\epsilon^{-2})$ gradient oracle calls for ∇f_i to achieve ϵ-accuracy.*

Proof. In each epoch k, the proposed algorithm makes $(n + pn)$ gradient oracle calls for ∇f_i. To achieve ϵ-accuracy, the number of epochs K satisfies $K \geq \mathcal{O}(\epsilon^{-2})$. Combining these two results, we obtain that the number of gradient oracle calls to achieve ϵ-accuracy is $(n + pn)K = \mathcal{O}(n\epsilon^{-2})$. $\square$

Remark 3.1. *The conventional random reshuffling algorithm, as demonstrated in [8], has a convergence rate of* $\mathcal{O}(\frac{1}{n^{1/3}K^{2/3}})$ *for nonconvex optimization. The convergence result in Theorem 3.1 shows that if the number of epochs* K *is larger than* n*, then the proposed PVR-RG algorithm achieves a faster convergence rate than the random reshuffling algorithm. Additionally, to achieve* ϵ*-accuracy, the random reshuffling algorithm requires* $\mathcal{O}(\frac{\sqrt{n}}{\epsilon^2}(\sqrt{n} + \frac{\sigma}{\epsilon}))$ *gradient oracle calls for* ∇f_i*, where* σ *is the gradient variance. Comparing this result with Corollary 3.1, it is seen that the proposed algorithm eliminates the effect of gradient variance and has a lower gradient complexity than the random reshuffling algorithm.*

4 Simulation

To demonstrate the performance of Algorithm 1, we apply the proposed PVR-RG algorithm to address the binary classification task by solving the following logistic regression optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x), \ f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + \exp(l_i p_i^\top x)} + \lambda_1 \|x\|_2^2,$$

where $x \in \mathbb{R}^d$ is the optimization variable, $\{(p_i, l_i)\}_{i=1}^n$ is the set of samples, $p_i \in \mathbb{R}^d$ is the feature vector, $l_i \in \{-1, 1\}$ is the classification label, and $\lambda_1 \triangleq 0.1$. For comparison, we apply some popular stochastic algorithms, i.e., the conventional stochastic gradient descent algorithm (SGD) and the random reshuffling gradient algorithm (RG) in [8], to solve the logistic regression problem.

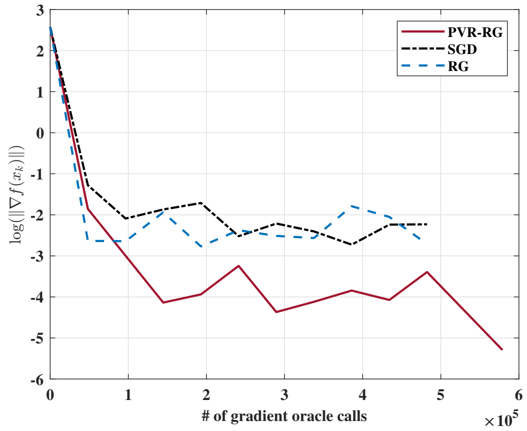
Table 1: Real datasets for logistic regression

datasets	#samples	#features	#classes
a9a	32561	123	2
w8a	64700	300	2
covtype.binary	581012	54	2

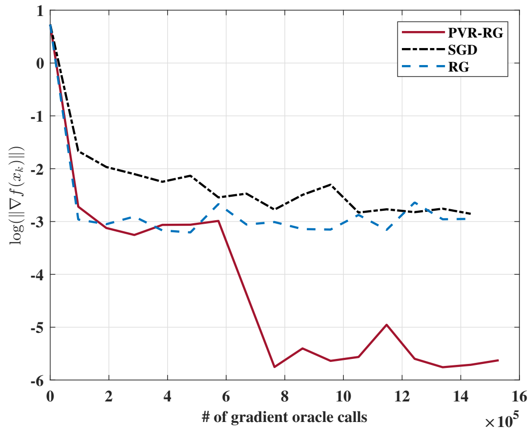
The initial conditions for all stochastic algorithms are identical. The constant step sizes for PVR-RG and RG are both set to 0.1, and the diminishing step size for SGD is set as $1/(k + 1)$. The experiments are conducted on different large-scale datasets,[1] which are summarized in Table 1. We provide a comparison of the gradient norm against the number of gradient oracle calls for different comparative algorithms in Fig. 1.

It is seen that all trajectories of the gradient norm converge with respect to the number of gradient oracle calls, and the PVR-RG algorithm converges faster than the SGD algorithm. Initially, the convergence behavior of PVR-RG algorithm is comparable to that of the RG algorithm. However, as the iterations progress, the PVR-RG algorithm exhibits a

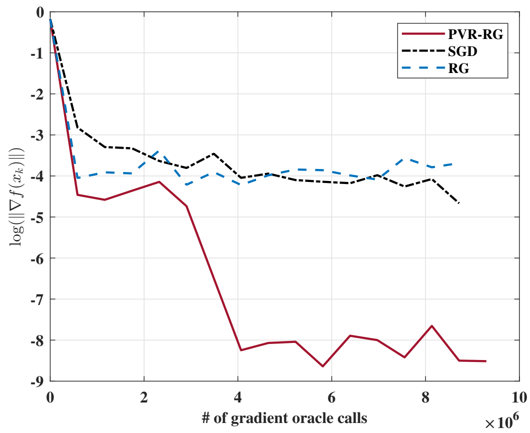
[1]The datasets are available from the website www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/.



(a) a9a dataset



(b) w8a dataset



(c) covtype dataset

Fig. 1: Convergence trajectories with respect to gradient oracle calls

faster convergence rate than the RG algorithm. The fluctuations observed at the tail end of the stochastic algorithms are due to the random sampling. The fluctuation of the proposed PVR-RG is also influenced by the selection of probability p.

In addition, the PVR-RG algorithm achieves a lower gradient norm than the other algorithms, indicating that the constant gradient variance term in the RG algorithm is eliminated. These simulation results verify the theoretical discussions in Remark 3.1 and confirm the effectiveness of the proposed algorithm.

5 Conclusion

This paper developed a novel shuffling gradient descent algorithm with a probabilistic variance reduction step for nonconvex optimization. The proposed variance reduction step utilizes the Bernoulli distribution and the SVRG technique to address the variance of stochastic gradients. While the shuffling gradient algorithms give rise to conditionally biased gradients, this paper established that the proposed stochastic algorithm achieves a faster convergence rate without the effect of gradient variance than the conventional shuffling gradient algorithm and SGD. One future research direction is to explore the global optimum convergence of shuffling gradient algorithms for nonconvex optimization.

References

[1] A. M.-C. So, P. Jain, W.-K. Ma, and G. Scutari, "Nonconvex optimization for signal processing and machine learning [from the guest editors]," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 15–17, 2020.

[2] W. Liu, G. Wang, J. Sun, F. Bullo, and J. Chen, "Learning robust data-based LQG controllers from noisy data," *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8526–8538, June 2024.

[3] Y. Cui, Z. He, and J.-S. Pang, "Multicomposite nonconvex optimization for training deep neural networks," *SIAM Journal on Optimization*, vol. 30, no. 2, pp. 1693–1723, 2020.

[4] P. Toulis, E. Airoldi, and J. Rennie, "Statistical analysis of stochastic gradient methods for generalized linear models," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 667–675.

[5] V. Patel, S. Zhang, and B. Tian, "Global convergence and stability of stochastic gradient descent," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[6] M. Gurbuzbalaban, A. Ozdaglar, and P. A. Parrilo, "Why random reshuffling beats stochastic gradient descent," *Mathematical Programming*, vol. 186, pp. 49–84, 2021.

[7] Q. Meng, W. Chen, Y. Wang, Z.-M. Ma, and T.-Y. Liu, "Convergence analysis of distributed stochastic gradient descent with shuffling," *Neurocomputing*, vol. 337, pp. 46–57, 2019.

[8] K. Mishchenko, A. Khaled, and P. Richtarik, "Random reshuffling: Simple analysis with vast improvements," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 17 309–17 320.

[9] B. Ying, K. Yuan, and A. H. Sayed, "Variance-reduced stochastic learning under random reshuffling," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1390–1408, 2020.

[10] L. M. Nguyen, Q. Tran-Dinh, D. T. Phan, P. H. Nguyen, and M. van Dijk, "A unified convergence analysis for shuffling-type gradient methods," *Journal of Machine Learning Research*, vol. 22, no. 207, pp. 1–44, 2021.

[11] X. Li, A. Milzarek, and J. Qiu, "Convergence of random reshuffling under the kurdyka–Łojasiewicz inequality," *SIAM Journal on Optimization*, vol. 33, no. 2, pp. 1092–1120, 2023.

[12] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 438–457, 2010.

[13] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, vol. 162, no. 1, pp. 83–112, 2017.

[14] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 1646–1654.

[15] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 315–323.

[16] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "SARAH: a novel method for machine learning problems using stochastic recursive gradient," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 2613–2621.

[17] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 314–323.

[18] Z. Allen-Zhu and E. Hazan, "Variance reduction for faster non-convex optimization," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 699–707.

[19] 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*, vol. 31. Curran Associates, Inc., 2018.

[20] G. Malinovsky, A. Sailanbayev, and P. Richtárik, "Random reshuffling with variance reduction: New analysis and better rates," in *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, ser. Proceedings of Machine Learning Research, vol. 216. PMLR, 31 Jul–04 Aug 2023, pp. 1347–1357.

[21] X. Jiang, X. Zeng, L. Xie, J. Sun, and J. Chen, "Variance-reduced reshuffling gradient descent for nonconvex optimization: Centralized and distributed algorithms," *Automatica*, vol. 171, p. 111954, 2025.

[22] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 560–569.