Lock-Free Optimization for Non-Convex Problems

Shen-Yi Zhao, Gong-Duo Zhang, Wu-Jun Li
National Key Laboratory for Novel Software Technology
Department of Computer Science and Technology, Nanjing University, China
{zhaosy, zhanggd}@lamda.nju.edu.cn, liwujun@nju.edu.cn

Abstract

Stochastic gradient descent (SGD) and its variants have attracted much attention in machine learning due to their efficiency and effectiveness for optimization. To handle large-scale problems, researchers have recently proposed several lock-free strategy based parallel SGD (LF-PSGD) methods for multi-core systems. However, existing works have only proved the convergence of these LF-PSGD methods for convex problems. To the best of our knowledge, no work has proved the convergence of the LF-PSGD methods for non-convex problems. In this paper, we provide the theoretical proof about the convergence of two representative LF-PSGD methods, Hogwild! and AsySVRG, for non-convex problems. Empirical results also show that both Hogwild! and AsySVRG are convergent on non-convex problems, which successfully verifies our theoretical results.

Introduction

Many machine learning models can be formulated as the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}), \tag{1}$$

where $\mathbf{w}$ is the parameter to learn (optimize), n is the number of training instances, $f_i(\mathbf{w})$ is the loss defined on instance i. For example, assuming we are given a set of labeled instances $\{(\mathbf{x}_i, y_i)|i = 1, 2, \ldots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{1, -1\}$ is the label of $\mathbf{x}_i$, $f_i(\mathbf{w})$ can be $\log(1 + e^{-y_i\mathbf{x}_i^T\mathbf{w}}) + \frac{\lambda}{2}\|\mathbf{w}\|^2$ which is known as the regularized loss in logistic regression (LR). We can also take $f_i(\mathbf{w})$ to be $\max\{0, 1 - y_i\mathbf{x}_i^T\mathbf{w}\} + \frac{\lambda}{2}\|\mathbf{w}\|^2$ which is known as the regularized loss in support vector machine (SVM). Here, λ is the regularization hyper-parameter. Moreover, many other machine learning models, including neural networks (Krizhevsky, Sutskever, and Hinton 2012), matrix factorization (Koren, Bell, and Volinsky 2009), and principal component analysis (PCA) (Shamir 2015) and so on, can also be formulated as that in (1).

When the problem in (1) is large-scale, i.e., n is large, researchers have recently proposed stochastic gradient descent (SGD) and its variants like SVRG (Johnson and Zhang 2013) to solve it. Many works (Roux, Schmidt, and Bach 2012; Shalev-Shwartz and Zhang 2013; Johnson and Zhang 2013) have found that SGD-based methods can achieve promising performance in large-scale learning problems. According to the implementation platforms or systems, existing SGD-based methods can be divided into three categories: sequential SGD (SSGD) methods, parallel SGD (PSGD) methods, and distributed SGD (DSGD) methods. SSGD methods are designed for a single thread on a single machine, PSGD methods are designed for multi-core (multi-thread) on a single machine with a shared memory[1], and DSGD methods are designed for multiple machines.

When the problem in (1) is convex, the SGD methods, including SSGD (Roux, Schmidt, and Bach 2012; Shalev-Shwartz and Zhang 2013; Johnson and Zhang 2013), PSGD (Recht et al. 2011) and DSGD (Jaggi et al. 2014; Li et al. 2014; Xing et al. 2015; Zhang, Zheng, and Kwok 2016), have achieved very promising empirical performance. Furthermore, good theoretical results about the convergence of the SGD methods are also provided by these existing works.

In many real applications, the problems to optimize can be non-convex. For example, the problems for the neural networks are typically non-convex. Because many researchers (Li et al. 2014; Xing et al. 2015) find that the SGD methods can also achieve good empirical results for non-convex problems, theoretical proof about the convergence of SGD methods for non-convex problems has recently attracted much attention. Some progress has been achieved. For example, the works in (Ghadimi and Lan 2013; Reddi et al. 2016; Li et al. 2016; Allen-Zhu and Hazan 2016; Allen-Zhu and Yuan 2016) have proved the convergence of the sequential SGD and its variants for non-convex problems. There are also some other theoretical results for some particular non-convex problems, like PCA (Shamir 2015; 2016a; 2016b) and matrix factorization (Sa, Re, and Olukotun 2015). But all these works are only for SSGD methods.

There have appeared only two works (Lian et al. 2015; Huo and Huang 2016) which propose PSGD methods for non-convex problems with theoretical proof of convergence.

[1]In some literatures, PSGD refers to the methods implemented on both multi-core and multi-machine systems. In this paper, PSGD only refers to the methods implemented on multi-core systems with a shared memory.

However, the PSGD methods in (Lian et al. 2015) need write-lock or atomic operation for the memory to prove the convergence [2]. Similarly, the work in (Huo and Huang 2016) also does not prove the convergence for the lock-free case in our paper. Recent works (Recht et al. 2011; Chaturapruek, Duchi, and Ré 2015; J. Reddi et al. 2015; Zhao and Li 2016) find that lock-free strategy based parallel SGD (LF-PSGD) methods can empirically outperform lock-based PSGD methods for multi-core systems. Although some existing works (Chaturapruek, Duchi, and Ré 2015; Zhao and Li 2016) have proved the convergence of these LF-PSGD methods for convex problems, no work has proved the convergence of the LF-PSGD methods for non-convex problems.

In this paper, we provide the theoretical proof about the convergence of two representative LF-PSGD methods, Hogwild! (Recht et al. 2011; Chaturapruek, Duchi, and Ré 2015) and AsySVRG (Zhao and Li 2016), for non-convex problems. The contribution of this work can be outlined as follows:

- Theoretical results show that both Hogwild! and AsySVRG can converge with lock-free strategy for non-convex problems.
- Hogwild! gets a convergence rate of $O(1/\sqrt{\tilde{T}})$ for non-convex problems, where $\tilde{T} = p \times T$ is the total iteration number of p threads.
- AsySVRG gets a convergence rate of $O(1/\tilde{T})$ for non-convex problems.
- To get an ϵ-local optimal solution for AsySVRG, the computation complexity by all threads is $O(n^{\frac{2}{3}}/\epsilon)$, or equivalently the computation complexity of each thread is $O(\frac{n^{\frac{2}{3}}}{p\epsilon})$. This is faster than traditional parallel gradient decent methods whose computation complexity is $O(\frac{n}{p\epsilon})$ for each thread.
- Empirical results also show that both Hogwild! and AsySVRG are convergent on non-convex problems, which successfully verifies our theoretical results.

Preliminary

We use $f(\mathbf{w})$ to denote the objective function in (1), which means $f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$. And we use $\|\cdot\|$ to denote the L_2-norm $\|\cdot\|_2$.

Assumption 1. *The function $f_i(\cdot)$ in (1) is smooth, which means that there exists a constant $L > 0$, $\forall \mathbf{a}, \mathbf{b}$,*

$$f_i(\mathbf{b}) \leq f_i(\mathbf{a}) + \nabla f_i(\mathbf{a})^T(\mathbf{b} - \mathbf{a}) + \frac{L}{2}\|\mathbf{b} - \mathbf{a}\|^2,$$

or equivalently

$$\|\nabla f_i(\mathbf{b}) - \nabla f_i(\mathbf{a})\| \leq L\|\mathbf{b} - \mathbf{a}\|.$$

[2]Although the implementation of AsySG-incon in (Lian et al. 2015) is lock-free, the theoretical analysis about the convergence of AsySG-incon is based on an assumption that no over-writing happens, i.e., the theoretical analysis is not for the lock-free case.

This is a common assumption for the convergence analysis of most existing gradient-based methods.

Since we focus on non-convex problems in this paper, it is difficult to get the global solution of (1) based on the gradient methods. Hence, we use $\|\nabla f(\mathbf{w})\|^2$ to measure the convergence instead of $f(\mathbf{w}) - \min_{\mathbf{w}} f(\mathbf{w})$.

Here, we give a Lemma which is useful in the convergence analysis of Hogwild! and AsySVRG.

Lemma 1. *Assume $\mathbf{B}$ is a positive semi-definite matrix with the largest eigenvalue less than or equal to 1 and the minimum eigenvalue $\alpha > 0$, we have: $\forall \mathbf{x}, \mathbf{y}$,*

$$-\nabla f(\mathbf{x})^T \mathbf{B} \nabla f(\mathbf{y}) \leq \frac{L^2}{2}\|\mathbf{x} - \mathbf{y}\|^2 - \frac{\alpha}{2}\|\nabla f(\mathbf{x})\|^2.$$

Proof.

$$\begin{aligned}
&\frac{\alpha}{2}\|\nabla f(\mathbf{x})\|^2 - \nabla f(\mathbf{x})^T \mathbf{B} \nabla f(\mathbf{y}) \\
\leq&\frac{1}{2}\left\|\mathbf{B}^{\frac{1}{2}}\nabla f(\mathbf{x})\right\|^2 - \nabla f(\mathbf{x})^T \mathbf{B} \nabla f(\mathbf{y}) \\
\leq&\frac{1}{2}\left\|\mathbf{B}^{\frac{1}{2}}\nabla f(\mathbf{x})\right\|^2 - \nabla f(\mathbf{x})^T \mathbf{B} \nabla f(\mathbf{y}) + \frac{1}{2}\left\|\mathbf{B}^{\frac{1}{2}}\nabla f(\mathbf{y})\right\|^2 \\
=&\frac{1}{2}\left\|\mathbf{B}^{\frac{1}{2}}(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))\right\|^2 \\
\leq&\frac{L^2}{2}\|\mathbf{x} - \mathbf{y}\|^2.
\end{aligned}$$

□

Hogwild! for Non-Convex Problems

The Hogwild! method (Recht et al. 2011) is listed in Algorithm 1. Each thread reads $\mathbf{w}$ from the shared memory, computes a stochastic gradient and updates the $\mathbf{w}$ in the shared memory. Please note that Hogwild! in (Recht et al. 2011) has several variants with locks or lock-free. Here, we only focus on the lock-free variant of Hogwild!, which means that we do not use any locks, either read-lock or write-lock, for all threads.

Algorithm 1 Hogwild!

```
Initialization: $p$ threads, initialize $\mathbf{w}_0, \eta$;
For each thread, do:
for $l = 0, 1, 2, ..., T-1$ do
    Read current $\mathbf{w}$ in the shared memory, denoted as $\hat{\mathbf{w}}$;
    Randomly pick up an $i$ from $\{1, \ldots, n\}$ and compute the gradient $\nabla f_i(\hat{\mathbf{w}})$;
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f_i(\hat{\mathbf{w}})$;
end for
```

As in (Zhao and Li 2016), we can construct an equivalent write sequence $\{\mathbf{w}_t\}$:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{B}_t \nabla f_{i_t}(\hat{\mathbf{w}}_t), \quad (2)$$

where $0 \leq t \leq p \times T$, $\mathbf{B}_t$ is a random diagonal matrix whose diagonal entries are 0 or 1. The $\mathbf{B}_t$ is used to denote whether over-writing happens. If the kth diagonal entry of $\mathbf{B}_t$ is 0, it means that the kth element in the gradient vector $\nabla f_{i_t}(\hat{\mathbf{w}}_t)$

is overwritten by other threads. Otherwise, that element is not overwritten.

$\hat{\mathbf{w}}_t$ is read by the thread who computes $\nabla f_{i_t}(\hat{\mathbf{w}}_t)$ and has the following format:

$$\hat{\mathbf{w}}_t = \mathbf{w}_{a(t)} - \eta \sum_{j=a(t)}^{t-1} \mathbf{P}_{t,j-a(t)} \nabla f_{i_j}(\hat{\mathbf{w}}_j), \tag{3}$$

where $a(t)$ means that some old stochastic gradients have been completely written on the $\mathbf{w}$ in the shared memory. $\mathbf{P}_{t,j-a(t)}$ is a diagonal matrix whose diagonal entries are 0 or 1, which means $\hat{\mathbf{w}}_t$ might include parts of new stochastic gradients.

In the lock-free strategy, we need the following assumptions to guarantee convergence:

Assumption 2. *$a(t)$ is bounded by:* $0 \le t - a(t) \le \tau$

It means that the old stochastic gradients $\nabla f_{i_0}, \ldots, \nabla f_{i_{t-\tau-1}}$ have been completely written on $\mathbf{w}$ in the shared memory.

Assumption 3. *We consider the matrix $\mathbf{B}_t$ as a random matrix and $\mathbb{E}[\mathbf{B}_t|\mathbf{w}_t, \hat{\mathbf{w}}_t] = \mathbf{B} \succ 0$ with the minimum eigenvalue $\alpha > 0$.*

According to the definition of $\mathbf{B}_t$, it is easy to find $\mathbf{B}_t, \mathbf{B}$ are positive semi-definite matrices and the largest eigenvalue of $\mathbf{B}$ is less than or equal to 1. Assumption 3 means that the probability that over-writing happens is at most $1 - \alpha < 1$ for each write step.

Assumption 4. *$\mathbf{B}_t$ and i_t are independent.*

Since i_t is the random index selected by each thread while $\mathbf{B}_t$ is highly affected by the hardware, the independence assumption is reasonable.

For Hogwild!, the following assumption is also necessary:

Assumption 5. *There exists a constant V, $\|\nabla f_i(\mathbf{w})\| \le V, i = 1, \ldots, n$.*

For convenience, in this section, we denote

$$q(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \|f_i(\mathbf{x})\|^2.$$

It is easy to find that $\mathbb{E}q(\hat{\mathbf{w}}_t) = \mathbb{E}[\|\nabla f_{i_t}(\hat{\mathbf{w}}_t)\|^2]$ and note that when $\mathbf{x}$ is close to some stationary point, $q(\mathbf{x})$ may still be far away from 0. Hence, it is not a variance reduction method and we need to control the variance of the stochastic gradient.

The difficulty of the analysis is $\mathbf{w}_t \ne \hat{\mathbf{w}}_t$. Here, we give the following Lemmas [3]:

Lemma 2. *In Hogwild!, we have $\mathbb{E}q(\hat{\mathbf{w}}_t) \le \rho\mathbb{E}q(\hat{\mathbf{w}}_{t+1})$ if ρ, η satisfy*

$$\frac{1}{1 - \eta - \frac{9\eta(\tau+1)L^2(\rho^{\tau+1}-1)}{\rho-1}} \le \rho.$$

[3]The proof of some Lemmas can be found in the supplementary material, which can be downloaded from http://cs.nju.edu.cn/lwj/paper/LFnonConvex_sup.pdf.

Lemma 3. *With the condition about ρ, η in Lemma 2, we have*

$$\mathbb{E}\|\mathbf{w}_t - \hat{\mathbf{w}}_t\|^2 \le \frac{4\eta^2\tau\rho(\rho^\tau - 1)}{\rho - 1}\mathbb{E}q(\hat{\mathbf{w}}_t) \tag{4}$$

Combining with Assumption 5, we can find that the gap of the write sequence and read sequence can always be bounded by a constant $\frac{4\eta^2V^2\tau\rho(\rho^\tau-1)}{\rho-1}$.

Theorem 1. *Let $A = \frac{2f(\mathbf{w}_0)}{\alpha}$ and $B = 2V^2(\frac{2\tau L^2\eta\rho(\rho^\tau-1)}{\alpha(\rho-1)} + \frac{L}{2\alpha})$. If we take the stepsize $\eta = \sqrt{\frac{A}{\tilde{T}B}}$, where $\tilde{T} = p \times T$, we can get the following result:*

$$\frac{1}{\tilde{T}}\sum_{t=0}^{\tilde{T}-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le \sqrt{\frac{AB}{\tilde{T}}}.$$

Proof. According to Assumption 1, we have

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{w}_{t+1})|\mathbf{w}_t, \hat{\mathbf{w}}_t] \\
\le& f(\mathbf{w}_t) - \eta\mathbb{E}[\nabla f(\mathbf{w}_t)^T\mathbf{B}_t\nabla f_{i_t}(\hat{\mathbf{w}}_t)|\mathbf{w}_t, \hat{\mathbf{w}}_t] \\
&+ \frac{L\eta^2}{2}\mathbb{E}[\|\nabla f_{i_t}(\hat{\mathbf{w}}_t)\|^2|\mathbf{w}_t, \hat{\mathbf{w}}_t] \\
=& f(\mathbf{w}_t) - \eta\nabla f(\mathbf{w}_t)^T\mathbf{B}\nabla f(\hat{\mathbf{w}}_t) \\
&+ \frac{L\eta^2}{2}\mathbb{E}[\|\nabla f_{i_t}(\hat{\mathbf{w}}_t)\|^2|\mathbf{w}_t, \hat{\mathbf{w}}_t] \\
\le& f(\mathbf{w}_t) - \frac{\alpha\eta}{2}\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L^2\eta}{2}\|\mathbf{w}_t - \hat{\mathbf{w}}_t\|^2 \\
&+ \frac{L\eta^2}{2}\mathbb{E}[\|\nabla f_{i_t}(\hat{\mathbf{w}}_t)\|^2|\mathbf{w}_t, \hat{\mathbf{w}}_t],
\end{aligned}$$

where the first equality uses Assumption 4, the second inequality uses Lemma 1. Taking expectation on the above inequality, we obtain

$$\begin{aligned}
&\mathbb{E}f(\mathbf{w}_{t+1}) \\
\le&\mathbb{E}f(\mathbf{w}_t) - \frac{\alpha\eta}{2}\mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 + \frac{L^2\eta}{2}\mathbb{E}\|\mathbf{w}_t - \hat{\mathbf{w}}_t\|^2 \\
&+ \frac{L\eta^2V^2}{2} \\
\le&\mathbb{E}f(\mathbf{w}_t) - \frac{\alpha\eta}{2}\mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \\
&+ \eta^2V^2(\frac{2\tau L^2\eta\rho(\rho^\tau - 1)}{\rho - 1} + \frac{L}{2}),
\end{aligned}$$

where the first inequality uses Assumption 5 and second inequality uses Lemma 3. Summing the above inequality from $t = 0$ to $\tilde{T} - 1$, we get

$$\begin{aligned}
&\sum_{t=0}^{\tilde{T}-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \\
\le&\frac{2}{\alpha\eta}f(\mathbf{w}_0) + 2\eta\tilde{T}V^2(\frac{2\tau L^2\eta\rho(\rho^\tau - 1)}{\alpha(\rho - 1)} + \frac{L}{2\alpha}).
\end{aligned}$$

For convenience, let $A = \frac{2f(\mathbf{w}_0)}{\alpha}$ and $B = 2V^2(\frac{2\tau L^2\eta\rho(\rho^\tau-1)}{\alpha(\rho-1)} + \frac{L}{2\alpha})$, which are two bounded constants.

If we take the stepsize $\eta = \sqrt{\frac{A}{\tilde{T}B}}$, we get

$$\frac{1}{\tilde{T}} \sum_{t=0}^{\tilde{T}-1} \mathbb{E}\|\nabla f(\mathbf{w}_t)\|^2 \le \sqrt{\frac{AB}{\tilde{T}}}.$$

□

Hence, our theoretical result shows that Hogwild! with lock-free strategy gets a convergence rate of $O(1/\sqrt{\tilde{T}})$ for non-convex problems, where $\tilde{T} = p \times T$ is the total iteration number of p threads.

AsySVRG for Non-Convex Problems

The AsySVRG method (Zhao and Li 2016) is listed in Algorithm 2. AsySVRG provides a lock-free parallel strategy for the original sequential SVRG (Johnson and Zhang 2013). Compared with Hogwild!, AsySVRG includes the full gradient to get a variance reduced stochastic gradient, which has been proved to have linear convergence rate on strongly convex problems (Zhao and Li 2016). In this section, we will prove that AsySVRG is also convergent for non-convex problems, and has faster convergence rate than Hogwild! on non-convex problems.

Algorithm 2 AsySVRG

```
Initialization: $p$ threads, initialize $\mathbf{w}_0, \eta$;
for $t = 0, 1, 2, ...T-1$ do
  $\mathbf{u}_0 = \mathbf{w}_t$;
  All threads parallelly compute the full gradient $\nabla f(\mathbf{u}_0) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{u}_0)$;
  $\mathbf{u} = \mathbf{w}_t$;
  For each thread, do:
  for $j = 0$ to $M-1$ do
    Read current value of $\mathbf{u}$, denoted as $\hat{\mathbf{u}}$, from the shared memory. And randomly pick up an $i$ from $\{1, \ldots, n\}$;
    Compute the update vector: $\hat{\mathbf{v}} = \nabla f_i(\hat{\mathbf{u}}) - \nabla f_i(\mathbf{u}_0) + \nabla f(\mathbf{u}_0)$;
    $\mathbf{u} \leftarrow \mathbf{u} - \eta\hat{\mathbf{v}}$;
  end for
  Take $\mathbf{w}_{t+1}$ to be the current value of $\mathbf{u}$ in the shared memory;
end for
```

Similar to the analysis in the last section, we construct an equivalent write sequence $\{\mathbf{u}_{t,m}\}$ for the t^{th} outer-loop:

$$\mathbf{u}_{t,0} = \mathbf{w}_t,$$
$$\mathbf{u}_{t,m+1} = \mathbf{u}_{t,m} - \eta \mathbf{B}_{t,m}\hat{\mathbf{v}}_{t,m}, \tag{5}$$

where $\hat{\mathbf{v}}_{t,m} = \nabla f_{i_{t,m}}(\hat{\mathbf{u}}_{t,m}) - \nabla f_{i_{t,m}}(\mathbf{u}_{t,0}) + \nabla f(\mathbf{u}_{t,0})$. $\mathbf{B}_{t,m}$ is a diagonal matrix whose diagonal entries are 0 or 1. And $\hat{\mathbf{u}}_{t,m}$ is read by the thread who computes $\hat{\mathbf{v}}_{t,m}$. It has the following format:

$$\hat{\mathbf{u}}_{t,m} = \mathbf{u}_{t,a(m)} - \eta \sum_{j=a(m)}^{m-1} \mathbf{P}^{(t)}_{m,j-a(m)} \hat{\mathbf{v}}_{t,j},$$

where $\mathbf{P}^{(t)}_{m,j-a(m)}$ is a diagonal matrix whose diagonal entries are 0 or 1. Note that according to (5), $\mathbf{u}_{t,\tilde{M}} = \mathbf{w}_{t+1}$ since all the stochastic gradients have been written on $\mathbf{w}$ at the end of the t^{th} outer-loop. Here, we also need the assumptions: $0 \le m - a(m) \le \tau$; $\mathbb{E}[\mathbf{B}_{t,m}|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}] = \mathbf{B} \succ 0$ with the minimum eigenvalue $\alpha > 0$; $\mathbf{B}_{t,m}$ and $i_{t,m}$ are independent. These assumptions are similar to those in the previous section.

For convenience, let $\mathbf{p}_i(\mathbf{x}) = \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{u}_{t,0}) + \nabla f(\mathbf{u}_{t,0})$, and in this section, we denote

$$q(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{p}_i(\mathbf{x})\|^2.$$

It easy to find that $\mathbb{E}q(\hat{\mathbf{u}}_{t,m}) = \mathbb{E}[\|\hat{\mathbf{v}}_{t,m}\|^2]$.

The difference between Hogwild! and AsySVRG is the stochastic gradient and we have the following Lemmas which lead to fast convergence rate of AsySVRG:

Lemma 4. $\forall \mathbf{x}$*, we have*

$$q(\mathbf{x}) \le 2L^2\|\mathbf{x} - \mathbf{u}_{t,0}\|^2 + 2\|\nabla f(\mathbf{x})\|^2.$$

Proof.

$$\begin{aligned} q(\mathbf{x}) =& \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{u}_{t,0}) + \nabla f(\mathbf{u}_{t,0})\|^2 \\ \le& \frac{2}{n}\sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{u}_{t,0}) + \nabla f(\mathbf{u}_{t,0}) - \nabla f(\mathbf{x})\|^2 \\ &+ 2\|\nabla f(\mathbf{x})\|^2 \\ \le& \frac{2}{n}\sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{u}_{t,0})\|^2 + 2\|\nabla f(\mathbf{x})\|^2 \\ \le& 2L^2\|\mathbf{x} - \mathbf{u}_{t,0}\|^2 + 2\|\nabla f(\mathbf{x})\|^2. \end{aligned}$$

□

According to Lemma 4, we can find that AsySVRG is a variance reduction method for non-convex problems, because when $\hat{\mathbf{u}}_{t,m}, \mathbf{u}_{t,0}$ get close to some stationary point, $q(\hat{\mathbf{u}}_{t,m})$ gets close to 0. And hence we do not need the bounded gradient assumption for the convergence proof.

Since $\mathbf{u}_{t,m} \ne \hat{\mathbf{u}}_{t,m}$, the difficulty of convergence analysis lies in the gap between $\mathbf{u}_{t,m}$ and $\hat{\mathbf{u}}_{t,m}$, and the relation between $q(\hat{\mathbf{u}}_{t,m})$ and $q(\mathbf{u}_{t,m})$.

Lemma 5. *In AsySVRG, we have* $\mathbb{E}q(\hat{\mathbf{u}}_{t,m}) < \rho\mathbb{E}q(\hat{\mathbf{u}}_{t,m+1})$ *if we choose* ρ *and* η *to satisfy that*

$$\frac{1}{1 - \eta - \frac{9\eta(\tau+1)L^2(\rho^{\tau+1}-1)}{\rho-1}} \le \rho.$$

Lemma 6. *With the condition about* ρ, η *in Lemma 5, we have*

$$\mathbb{E}\|\mathbf{u}_{t,m} - \hat{\mathbf{u}}_{t,m}\|^2 \le \frac{4\eta^2\tau\rho(\rho^\tau - 1)}{\rho - 1}\mathbb{E}q(\hat{\mathbf{u}}_{t,m}). \tag{6}$$

Lemma 7. *With the condition about* ρ, η *in Lemma 5, we have* $\mathbb{E}q(\hat{\mathbf{u}}_{t,m}) < \rho\mathbb{E}q(\mathbf{u}_{t,m})$.

Combining Lemma 6 and Lemma 7, we can directly obtain:

$$\mathbb{E}\left\|\hat{\mathbf{u}}_{t,m} - \mathbf{u}_{t,m}\right\|^2 \le \frac{4\eta^2\tau\rho^2(\rho^\tau - 1)}{\rho - 1}\mathbb{E}q(\mathbf{u}_{t,m}). \tag{7}$$

Theorem 2. *We define* $c_m = c_{m+1}(1+\beta\eta) + 2L^2\eta^2 h_{m+1}$, $h_m = (\frac{\eta L^2}{2} + \frac{2c_m\eta}{\beta})\frac{4\tau\rho^2(\rho^\tau-1)}{\rho-1} + (c_m\rho + \frac{L\rho}{2})$ *with* $c_0, \beta > 0$. *Furthermore, we choose* c_0, η, β *such that* $\gamma = \min \frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta} - 2\eta^2 h_{m+1} > 0$ *and* $c_{\tilde{M}} = 0$, *where* $\tilde{M} = M \times p$. *Then we have*

$$\frac{1}{T\tilde{M}}\sum_{t=0}^{T-1}\sum_{m=0}^{\tilde{M}-1}\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2 \le \frac{\mathbb{E}f(\mathbf{w}_0) - \mathbb{E}f(\mathbf{w}_T)}{T\tilde{M}\gamma}.$$

Proof. In the t^{th} outer-loop, similar to (Reddi et al. 2016), we define $R_{t,m}$ as follows

$$R_{t,m} = f(\mathbf{u}_{t,m}) + c_m\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2.$$

Then $\forall\beta > 0$,

$$\begin{aligned}
&\mathbb{E}[\|\mathbf{u}_{t,m+1} - \mathbf{u}_{t,0}\|^2|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}]\\
\le&\mathbb{E}\|\mathbf{u}_{t,m+1} - \mathbf{u}_{t,m}\|^2 + \|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&- 2\eta(\mathbb{E}\mathbf{B}_{t,m}\hat{\mathbf{v}}_{t,m})^T(\mathbf{u}_{t,m} - \mathbf{u}_{t,0})\\
\le&\eta^2\mathbb{E}\|\hat{\mathbf{v}}_{t,m}\|^2 + (1+\beta\eta)\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&+ \frac{\eta}{\beta}\|\nabla f(\hat{\mathbf{u}}_{t,m})\|^2\\
\le&\eta^2\mathbb{E}\|\hat{\mathbf{v}}_{t,m}\|^2 + (1+\beta\eta)\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&+ \frac{2\eta}{\beta}(\|\nabla f(\mathbf{u}_{t,m})\| + \|\nabla f(\hat{\mathbf{u}}_{t,m}) - \nabla f(\mathbf{u}_{t,m})\|^2)\\
\le&\eta^2\mathbb{E}\|\hat{\mathbf{v}}_{t,m}\|^2 + (1+\beta\eta)\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&+ \frac{2\eta}{\beta}(\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&+ L^2\|\hat{\mathbf{u}}_{t,m} - \mathbf{u}_{t,m}\|^2),
\end{aligned} \tag{8}$$

where the second inequality uses the fact $2ab \le \beta a^2 + \frac{1}{\beta}b^2$. Since the objective function is L-smooth, we have

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{u}_{t,m+1})|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}]\\
\le& - \eta\mathbb{E}[\nabla f(\mathbf{u}_{t,m})^T\mathbf{B}_{t,m}\nabla f_{i_{t,m}}(\hat{\mathbf{u}}_{t,m})|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}]\\
&+ f(\mathbf{u}_{t,m}) + \frac{L\eta^2}{2}\mathbb{E}[\|\hat{\mathbf{v}}_{t,m}\|^2|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}]\\
=&f(\mathbf{u}_{t,m}) - \eta\nabla f(\mathbf{u}_{t,m})^T\mathbf{B}\nabla f(\hat{\mathbf{u}}_{t,m})\\
&+ \frac{L\eta^2}{2}\mathbb{E}[\|\hat{\mathbf{v}}_{t,m}\|^2|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}]\\
\le&f(\mathbf{u}_{t,m}) - \frac{\alpha\eta}{2}\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&+ \frac{\eta L^2}{2}\|\mathbf{u}_{t,m} - \hat{\mathbf{u}}_{t,m}\|^2\\
&+ \frac{L\eta^2}{2}\mathbb{E}[\|\hat{\mathbf{v}}_{t,m}\|^2|\mathbf{u}_{t,m}, \hat{\mathbf{u}}_{t,m}],
\end{aligned} \tag{9}$$

where the first equality uses the independence of $\mathbf{B}_{t,m}, i_{t,m}$, the second inequality uses Lemma 1. Combining (8) and (9), we have

$$\begin{aligned}
&\mathbb{E}R_{t,m+1}\\
=&\mathbb{E}f(\mathbf{u}_{t,m+1}) + c_{m+1}\|\mathbf{u}_{t,m+1} - \mathbf{u}_{t,0}\|^2\\
\le&\mathbb{E}f(\mathbf{u}_{t,m}) - (\frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta})\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&+ (\frac{\eta L^2}{2} + \frac{2c_{m+1}\eta}{\beta})\mathbb{E}\|\mathbf{u}_{t,m} - \hat{\mathbf{u}}_{t,m}\|^2\\
&+ c_{m+1}(1+\beta\eta)\mathbb{E}\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&+ \eta^2(c_{m+1} + \frac{L}{2})\mathbb{E}\|\hat{\mathbf{v}}_{t,m}\|^2\\
\le&\mathbb{E}f(\mathbf{u}_{t,m}) - (\frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta})\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&+ (\frac{\eta L^2}{2} + \frac{2c_{m+1}\eta}{\beta})\frac{4\tau\eta^2\rho^2(\rho^\tau - 1)}{\rho - 1}\mathbb{E}q(\mathbf{u}_{t,m})\\
&+ c_{m+1}(1+\beta\eta)\mathbb{E}\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&+ \eta^2(c_{m+1} + \frac{L}{2})\mathbb{E}\|\hat{\mathbf{v}}_{t,m}\|^2,
\end{aligned}$$

where the last inequality uses equation (7).

For convenience, we use $h_m = (\frac{\eta L^2}{2} + \frac{2c_m\eta}{\beta})\frac{4\tau\rho^2(\rho^\tau-1)}{\rho-1} + \rho(c_m + \frac{L}{2})$. Since $\mathbb{E}[\|\hat{\mathbf{v}}_{t,m}\|^2] = \mathbb{E}q(\hat{\mathbf{u}}_{t,m}) \le \rho\mathbb{E}q(\mathbf{u}_{t,m})$, we have

$$\begin{aligned}
&\mathbb{E}R_{t,m+1}\\
\le&\mathbb{E}f(\mathbf{u}_{t,m}) - (\frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta})\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&+ c_{m+1}(1+\beta\eta)\mathbb{E}\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2 + \eta^2 h_{m+1}\mathbb{E}q(\mathbf{u}_{t,m})\\
\le&\mathbb{E}f(\mathbf{u}_{t,m})\\
&+ [c_{m+1}(1+\beta\eta) + 2L^2\eta^2 h_{m+1}]\mathbb{E}\|\mathbf{u}_{t,m} - \mathbf{u}_{t,0}\|^2\\
&- (\frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta} - 2\eta^2 h_{m+1})\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2,
\end{aligned}$$

where the second inequality uses Lemma 4. Then we can obtain:

$$\begin{aligned}
&(\frac{\alpha\eta}{2} - \frac{2c_{m+1}\eta}{\beta} - 2\eta^2 h_{m+1})\mathbb{E}\|\nabla f(\mathbf{u}_m)\|^2\\
&\le \mathbb{E}R_m - \mathbb{E}R_{m+1},
\end{aligned}$$

where $c_m = c_{m+1}(1+\beta\eta) + 2L^2\eta^2 h_{m+1}$.

We set $c_0 > 0$. It is easy to see that $c_m > c_{m+1}$. We can choose c_0, η, β to make $c_{\tilde{M}} = 0$. Then we have:

$$\begin{aligned}
&\sum_{m=0}^{\tilde{M}-1}\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2\\
&\le \frac{\mathbb{E}R_0 - \mathbb{E}R_{\tilde{M}}}{\gamma} = \frac{\mathbb{E}f(\mathbf{w}_t) - \mathbb{E}f(\mathbf{w}_{t+1})}{\gamma},
\end{aligned}$$

which is equivalent to

$$\frac{1}{T\tilde{M}}\sum_{t=0}^{T-1}\sum_{m=0}^{\tilde{M}-1}\mathbb{E}\|\nabla f(\mathbf{u}_{t,m})\|^2 \le \frac{\mathbb{E}f(\mathbf{w}_0) - \mathbb{E}f(\mathbf{w}_T)}{T\tilde{M}\gamma}.$$

□

Computation Complexity

In Theorem 2, we construct a sequence $\{c_m\}$ and need $\gamma > 0$. According to the definition of h_m, we can write h_m as $h_m = gc_m + f$, where $g = \frac{2\eta}{\beta}\frac{4\tau\rho^2(\rho^\tau - 1)}{\rho - 1} + \rho, f = \frac{\eta L^2}{2}\frac{4\tau\rho^2(\rho^\tau-1)}{\rho-1} + \frac{L\rho}{2}$ are constants.

First, we choose $\beta > \eta$, then both g, f are bounded positive constants. We have

$$c_m = c_{m+1}(1 + \beta\eta + 2L^2\eta^2 g) + 2L^2\eta^2 f.$$

Let $a = \beta\eta + 2L^2\eta^2 g$. Because $c_{\tilde{M}} = 0$, it is easy to get

$$c_0 = 2L^2\eta^2 f\frac{(1+a)^{\tilde{M}} - 1}{a}.$$

We take $\tilde{M} = \lfloor \frac{1}{a} \rfloor \le \frac{1}{a}$, then we have $c_0 \le \frac{4L^2\eta^2 f}{a}$ and

$$\gamma = \frac{\alpha}{2}(\eta - \frac{4c_0\eta}{\alpha\beta} - \frac{4gc_0}{\alpha}\eta^2 - \frac{4f}{\alpha}\eta^2).$$

As recommended in (Reddi et al. 2016), we can take $\eta = \mu/n^{2/3}, \beta = v/n^{1/3}$ with $\eta < \beta$ (assuming n is large). Then we can get $f = O(1), g = O(1), a = O(1/n)$. By choosing μ, v to satisfy $\frac{16L^2 f\mu}{\alpha v^2} < 1$ such that $\frac{4c_0}{\alpha\beta} < 1$, it is easy to find that $\gamma = O(1/n^{2/3}) > 0$, $\tilde{M} = O(n)$. Hence, to get an ϵ-local optimal solution, the computation complexity by all p threads is $O(n^{\frac{2}{3}}/\epsilon)$, and the computation complexity of each thread is $O(\frac{n^{\frac{2}{3}}}{p\epsilon})$.

Experiment

To verify our theoretical results about Hogwild! and AsySVRG, we use a fully-connected neural network to construct a non-convex function. The neural network has one hidden layer with 100 nodes and the sigmoid function is used for the activation function. We use the soft-max output and a L_2 regularization for training. The loss function is:

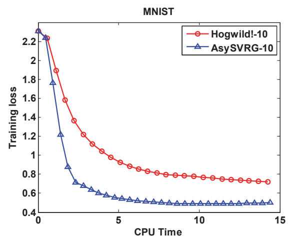
$$f(\mathbf{w}, \mathbf{b}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbf{1}\{y_i = k\}\log o_i^{(k)} + \frac{\lambda}{2}\|\mathbf{w}\|^2,$$

where $\mathbf{w}$ is the weights of the neural network, $\mathbf{b}$ is the bias, y_i is the label of instance $\mathbf{x}_i$, $o_i^{(k)}$ is the output corresponding to $\mathbf{x}_i$, K is the total number of class labels.

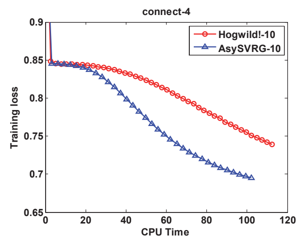
We use two datasets: connect-4 and MNIST[4] to do experiments and $\lambda = 10^{-3}$. We initialize $\mathbf{w}$ by randomly sampling from a Gaussian distribution with mean being 0 and variance being 0.01, and initialize $\mathbf{b} = 0$. During training, we use a fixed stepsize for both Hogwild! and AsySVRG. The stepsize is chosen from $\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$, and the best is reported. For the iteration number of the inner-loop of AsySVRG, we set $M = n/p$, where p is the number of threads. The experiments are conducted on a server with 12 Intel cores and 64G memory.

[4]https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

Figure 1 illustrates the convergence property of both Hogwild! and AsySVRG. The x-axis denotes the CPU time, where we set the CPU time that Hogwild! passes through the whole dataset once with one thread as 1 unit. The y-axis denotes the training loss. In this experiment, we run Hogwild! and AsySVRG with 10 threads. Hogwild!-10 and AsySVRG-10 denote the corresponding methods with 10 threads. It is easy to see that both Hogwild! and AsySVRG are convergent. Furthermore, AsySVRG is faster than Hogwild!. This is consistent with our theoretical results in this paper.



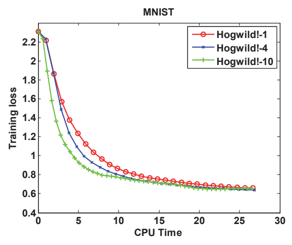
(a) MNIST



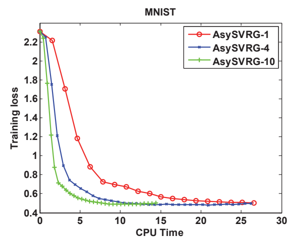
(b) connect-4

Figure 1: Hogwild! vs AsySVRG

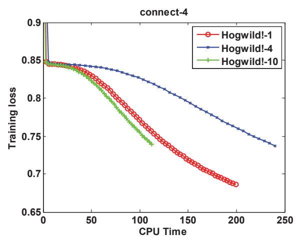
Figure 2 reports the results of Hogwild! and AsySVRG with different numbers of threads, where the number of threads $p = 1, 4, 10$. We can find that in most cases the two methods will become faster with the increase of threads. The only outlier is the case for Hogwild! on dateset connect-4, Hogwild! using 4 threads is slower than using 1 thread. One possible reason is that we have two CPUs in our server, with 6 cores for each CPU. In the 4-thread case, different threads may be allocated on different CPUs, which will cause extra cost.



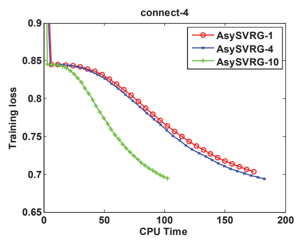
(a) Hogwild! on MNIST



(b) AsySVRG on MNIST



(c) Hogwild! on connect-4



(d) AsySVRG on connect-4

Figure 2: Comparison between different numbers of threads.

Conclusion

In this paper, we have provided theoretical proof about the convergence of two representative lock-free strategy based parallel SGD methods, Hogwild! and AsySVRG, for non-convex problems. Empirical results also show that both Hogwild! and AsySVRG are convergent on non-convex problems, which successfully verifies our theoretical results. To the best of our knowledge, this is the first work to prove the convergence of lock-free strategy based parallel SGD methods for non-convex problems.

Acknowledgements

This work is partially supported by NSFC (No. 61472182) and a fund from Tencent.

References

Allen-Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-convex optimization. In *Proceedings of the 33nd International Conference on Machine Learning*.

Allen-Zhu, Z., and Yuan, Y. 2016. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *Proceedings of the 33nd International Conference on Machine Learning*.

Chaturapruek, S.; Duchi, J. C.; and Ré, C. 2015. Asynchronous stochastic convex optimization: the noise is in the noise and sgd don't care. In *Proceedings of the Advances in Neural Information Processing Systems*.

Ghadimi, S., and Lan, G. 2013. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23(4):2341–2368.

Huo, Z., and Huang, H. 2016. Asynchronous stochastic gradient descent with variance reduction for non-convex optimization. *CoRR* abs/1604.03584.

J. Reddi, S.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. J. 2015. On variance reduction in stochastic gradient descent and its asynchronous variants. In *Proceedings of the Advances in Neural Information Processing Systems*.

Jaggi, M.; Smith, V.; Takac, M.; Terhorst, J.; Krishnan, S.; Hofmann, T.; and Jordan, M. I. 2014. Communication-efficient distributed dual coordinate ascent. In *Proceedings of the Advances in Neural Information Processing Systems*.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the Advances in Neural Information Processing Systems*.

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8):30–37.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*.

Li, M.; Andersen, D. G.; Park, J. W.; Smola, A. J.; Ahmed, A.; Josifovski, V.; Long, J.; Shekita, E. J.; and Su, B. 2014. Scaling distributed machine learning with the parameter server. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation*.

Li, X.; Zhao, T.; Arora, R.; Han; and Haupt, J. 2016. Stochastic variance reduced optimization for nonconvex sparse learning. In *Proceedings of the 33nd International Conference on Machine Learning*.

Lian, X.; Huang, Y.; Li, Y.; and Liu, J. 2015. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Proceedings of the Advances in Neural Information Processing Systems*.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of the Advances in Neural Information Processing Systems*.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Alex. 2016. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33nd International Conference on Machine Learning*.

Roux, N. L.; Schmidt, M. W.; and Bach, F. 2012. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the Advances in Neural Information Processing Systems*.

Sa, C. D.; Re, C.; and Olukotun, K. 2015. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the 32nd International Conference on Machine Learning*.

Shalev-Shwartz, S., and Zhang, T. 2013. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research* 14(1):567–599.

Shamir, O. 2015. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 32nd International Conference on Machine Learning*.

Shamir, O. 2016a. Convergence of stochastic gradient descent for pca. In *Proceedings of the 33nd International Conference on Machine Learning*.

Shamir, O. 2016b. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *Proceedings of the 33nd International Conference on Machine Learning*.

Xing, E. P.; Ho, Q.; Dai, W.; Kim, J. K.; Wei, J.; Lee, S.; Zheng, X.; Xie, P.; Kumar, A.; and Yu, Y. 2015. Petuum: A new platform for distributed machine learning on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhang, R.; Zheng, S.; and Kwok, J. T. 2016. Asynchronous distributed semi-stochastic gradient optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhao, S.-Y., and Li, W.-J. 2016. Fast asynchronous parallel stochastic gradient descent: A lock-free approach with convergence guarantee. In *Proceedings of the AAAI Conference on Artificial Intelligence*.