SGD and Hogwild! Convergence Without the Bounded Gradients Assumption

Lam M. Nguyen [1 2] **Phuong Ha Nguyen** [3] **Marten van Dijk** [3] **Peter Richtárik** [4] **Katya Scheinberg** [1] **Martin Takáč** [1]

Abstract

Stochastic gradient descent (SGD) is the optimization algorithm of choice in many machine learning applications such as regularized empirical risk minimization and training deep neural networks. The classical convergence analysis of SGD is carried out under the assumption that the norm of the stochastic gradient is uniformly bounded. While this might hold for some loss functions, it is always violated for cases where the objective function is strongly convex. In (Bottou et al., 2016), a new analysis of convergence of SGD is performed under the assumption that stochastic gradients are bounded with respect to the true gradient norm. Here we show that for stochastic problems arising in machine learning such bound always holds; and we also propose an alternative convergence analysis of SGD with diminishing learning rate regime, which results in more relaxed conditions than those in (Bottou et al., 2016). We then move on the asynchronous parallel setting, and prove convergence of Hogwild! algorithm in the same regime, obtaining the first convergence results for this method in the case of diminished learning rate.

1. Introduction

We are interested in solving the following stochastic optimization problem

$$\min_{w\in\mathbb{R}^d} \left\{ F(w) = \mathbb{E}[f(w;\xi)] \right\}, \tag{1}$$

where ξ is a random variable obeying some distribution.

In the case of empirical risk minimization with a training set $\{(x_i, y_i)\}_{i=1}^n$, ξ_i is a random variable that is defined by a single random sample (x, y) pulled uniformly from the training set. Then, by defining $f_i(w) := f(w;\xi_i)$, empirical risk minimization reduces to

$$\min_{w\in\mathbb{R}^d} \left\{ F(w) = \frac{1}{n}\sum_{i=1}^n f_i(w) \right\}. \tag{2}$$

Problem (2) arises frequently in supervised learning applications (Hastie et al., 2009). For a wide range of applications, such as linear regression and logistic regression, the objective function F is strongly convex and each f_i, $i \in [n]$, is convex and has Lipschitz continuous gradients (with Lipschitz constant L). Given a training set $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, the ℓ_2-regularized least squares regression model, for example, is written as (2) with $f_i(w) \overset{\text{def}}{=} (\langle x_i, w\rangle - y_i)^2 + \frac{\lambda}{2}\|w\|^2$. The ℓ_2-regularized logistic regression for binary classification is written with $f_i(w) \overset{\text{def}}{=} \log(1+\exp(-y_i\langle x_i, w\rangle)) + \frac{\lambda}{2}\|w\|^2$, $y_i \in \{-1, 1\}$. It is well established by now that solving this type of problem by gradient descent (GD) (Nesterov, 2004; Nocedal & Wright, 2006) may be prohibitively expensive and stochastic gradient descent (SGD) is thus preferable. Recently, a class of variance reduction methods (Le Roux et al., 2012; Defazio et al., 2014; Johnson & Zhang, 2013; Nguyen et al., 2017) has been proposed in order to reduce the computational cost. All these methods explicitly exploit the finite sum form of (2) and thus they have some disadvantages for very large scale machine learning problems and are not applicable to (1).

To apply SGD to the general form (1) one needs to assume existence of unbiased gradient estimators. This is usually defined as follows:

$$\mathbb{E}_\xi[\nabla f(w;\xi)] = \nabla F(w),$$

[1] Department of Industrial and Systems Engineering, Lehigh University, USA. [2] IBM Thomas J. Watson Research Center, USA. [3] Department of Electrical and Computer Engineering, University of Connecticut, USA. [4] KAUST, KSA — Edinburgh, UK — MIPT, Russia. Correspondence to: Lam M. Nguyen <LamNguyen.MLTD@gmail.com>, Phuong Ha Nguyen <phuongha.ntu@gmail.com>, Marten van Dijk <marten.van_dijk@uconn.edu>, Peter Richtárik <Peter.Richtarik@ed.ac.uk>, Katya Scheinberg <katyas@lehigh.edu>, Martin Takáč <Takac.MT@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

for any fixed w. Here we make an important observation: if we view (1) not as a general stochastic problem but as the expected risk minimization problem, where ξ corresponds to a random data sample pulled from a distribution, then (1) has an additional key property: for each realization of the random variable ξ, $f(w;\xi)$ is a convex function with Lipschitz continuous gradients. Notice that traditional analysis of SGD for general stochastic problem of the form (1) does not make any assumptions on individual function realizations. In this paper we derive convergence properties for SGD applied to (1) with these additional assumptions on $f(w;\xi)$ and also extend to the case when $f(w;\xi)$ are not necessarily convex.

Regardless of the properties of $f(w;\xi)$ we assume that F in (1) is strongly convex. We define the (unique) optimal solution of F as w_*.

Assumption 1 (μ-strongly convex). *The objective function $F : \mathbb{R}^d \to \mathbb{R}$ is a μ-strongly convex, i.e., there exists a constant $\mu > 0$ such that $\forall w, w' \in \mathbb{R}^d$,*

$$F(w) - F(w') \geq \langle \nabla F(w'), (w - w') \rangle + \frac{\mu}{2}\|w - w'\|^2. \tag{3}$$

It is well-known in literature (Nesterov, 2004; Bottou et al., 2016) that Assumption 1 implies

$$2\mu[F(w) - F(w_*)] \leq \|\nabla F(w)\|^2 \ , \ \forall w \in \mathbb{R}^d. \tag{4}$$

The classical theoretical analysis of SGD assumes that the *stochastic gradients are uniformly bounded*, i.e. there exists a finite (fixed) constant $\sigma < \infty$, such that

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq \sigma^2 \ , \ \forall w \in \mathbb{R}^d \tag{5}$$

(see e.g. (Shalev-Shwartz et al., 2007; Nemirovski et al., 2009; Recht et al., 2011; Hazan & Kale, 2014; Rakhlin et al., 2012), etc.). However, this assumption is clearly false if F is strongly convex. Specifically, under this assumption together with strong convexity, $\forall w \in \mathbb{R}^d$, we have

$$\begin{aligned} 2\mu[F(w) - F(w_*)] &\overset{(4)}{\leq} \|\nabla F(w)\|^2 = \|\mathbb{E}[\nabla f(w;\xi)]\|^2 \\ &\leq \mathbb{E}[\|\nabla f(w;\xi)\|^2] \overset{(5)}{\leq} \sigma^2. \end{aligned}$$

Hence,

$$F(w) \leq \frac{\sigma^2}{2\mu} + F(w_*) \ , \ \forall w \in \mathbb{R}^d.$$

On the other hand strong convexity and $\nabla F(w_*) = 0$ imply

$$F(w) \geq \mu\|w - w_*\|^2 + F(w_*) \ , \ \forall w \in \mathbb{R}^d.$$

The last two inequalities are clearly in contradiction with each other for sufficiently large $\|w - w_*\|^2$.

Let us consider the following example: $f_1(w) = \frac{1}{2}w^2$ and $f_2(w) = w$ with $F(w) = \frac{1}{2}(f_1(w) + f_2(w))$. Note that F is strongly convex, while individual realizations are not necessarily so. Let $w_0 = 0$, for any number $t \geq 0$, with probability $\frac{1}{2^t}$ the steps of SGD algorithm for all $i < t$ are $w_{i+1} = w_i - \eta_i$. This implies that $w_t = -\sum_{i=1}^{t} \eta_i$ and since $\sum_{i=1}^{\infty} \eta_i = \infty$ then $|w_t|$ can be arbitrarily large for large enough t with probability $\frac{1}{2^t}$. Noting that for this example, $\mathbb{E}[\|\nabla f(w_t;\xi)\|^2] = \frac{1}{2}w_t^2 + \frac{1}{2}$, we see that $\mathbb{E}[\|\nabla f(w_t;\xi)\|^2]$ can also be arbitrarily large.

Recently, in the review paper (Bottou et al., 2016), convergence of SGD for general stochastic optimization problem was analyzed under the following assumption: there exist constants M and N such that $\mathbb{E}[\|\nabla f(w_t;\xi_t)\|^2] \leq M\|\nabla F(w_t)\|^2 + N$, where w_t, $t \geq 0$, are generated by the algorithm. This assumption does not contradict strong convexity, however, in general, constants M and N are unknown, while M is used to determine the learning rate η_t (Bottou et al., 2016). In addition, the rate of convergence of the SGD algorithm depends on M and N. In this paper we show that under the smoothness assumption on individual realizations $f(w,\xi)$ it is possible to derive the bound $\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq M_0[F(w) - F(w_*)] + N$ with specific values of M_0, and N for $\forall w \in \mathbb{R}^d$, which in turn implies the bound $\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq M\|\nabla F(w)\|^2 + N$ with specific M, by strong convexity of F. We also note that, in (Moulines & Bach, 2011), the convergence of SGD without bounded gradient assumption is studied. We then provide an alternative convergence analysis for SGD which shows convergence in expectation with a bound on learning rate which is larger than that in (Bottou et al., 2016; Moulines & Bach, 2011) by a factor of L/μ. We then use the new framework for the convergence analysis of SGD to analyze an asynchronous stochastic gradient method.

In (Recht et al., 2011), an asynchronous stochastic optimization method called Hogwild! was proposed. Hogwild! algorithm is a parallel version of SGD, where each processor applies SGD steps independently of the other processors to the solution w which is shared by all processors. Thus, each processor computes a stochastic gradient and updates w without "locking" the memory containing w, meaning that multiple processors are able to update w at the same time. This approach leads to much better scaling of parallel SGD algorithm than a synchoronous version, but the analysis of this method is more complex. In (Recht et al., 2011; Mania et al., 2015; De Sa et al., 2015) various variants of Hogwild! with a fixed step size are analyzed under the assumption that the gradients are bounded as in (5). In this paper, we extend our analysis of SGD to provide analysis of Hogwild! with diminishing step sizes and without the assumption on bounded gradients.

In a recent technical report (Leblond et al., 2018) Hogwild!

with fixed step size is analyzed without the bounded gradient assumption. We note that SGD with fixed step size only converges to a neighborhood of the optimal solution, while by analyzing the diminishing step size variant we are able to show convergence to the *optimal solution* with probability one. Both in (Leblond et al., 2018) and in this paper, the version of Hogwild! with inconsistent reads and writes is considered.

1.1. Contribution

We provide a new framework for the analysis of stochastic gradient algorithms in the strongly convex case under the condition of Lipschitz continuity of the individual function realizations, but **without requiring any bounds on the stochastic gradients**. Within this framework we have the following contributions:

- We prove the almost sure (w.p.1) convergence of SGD with diminishing step size. Our analysis provides a larger bound on the possible initial step size when compared to any previous analysis of convergence in expectation for SGD.

- We introduce a general recurrence for vector updates which has as its special cases (a) Hogwild! algorithm with diminishing step sizes, where each update involves all non-zero entries of the computed gradient, and (b) a position-based updating algorithm where each update corresponds to only one uniformly selected non-zero entry of the computed gradient.

- We analyze this general recurrence under inconsistent vector reads from and vector writes to shared memory (where individual vector entry reads and writes are atomic in that they cannot be interrupted by writes to the same entry) assuming that there exists a delay τ such that during the $(t+1)$-th iteration a gradient of a read vector w is computed which includes the aggregate of all the updates up to and including those made during the $(t-\tau)$-th iteration. In other words, τ controls to what extend past updates influence the shared memory.
  - Our upper bound for the expected convergence rate is sublinear, i.e., $O(1/t)$, and its precise expression allows comparison of algorithms (a) and (b) described above.
  - For SGD we can improve this upper bound by a factor 2 and also show that its initial step size can be larger.
  - We show that τ can be a function of t as large as $\approx \sqrt{t/\ln t}$ without affecting the asymptotic behavior of the upper bound; we also determine a constant T_0 with the property that, for $t \geq T_0$, higher order terms containing parameter τ are smaller than the leading $O(1/t)$ term. We give intuition explaining why the expected convergence rate is not more affected by τ. Our experiments confirm our analysis.
  - We determine a constant T_1 with the property that, for $t \geq T_1$, the higher order term containing parameter $\|w_0 - w_*\|^2$ is smaller than the leading $O(1/t)$ term.

- All the above contributions generalize to the non-convex setting where we do not need to assume that the component functions $f(w;\xi)$ are convex in w.

1.2. Organization

We analyse the convergence rate of SGD in Section 2 and introduce the general recursion and its analysis in Section 3. Experiments are reported in Section 4.

2. New Framework for Convergence Analysis of SGD

We introduce SGD algorithm in Algorithm 1.

Algorithm 1 Stochastic Gradient Descent (SGD) Method

```
Initialize: $w_0$
Iterate:
for $t = 0, 1, 2, \dots$ do
    Choose a step size (i.e., learning rate) $\eta_t > 0$.
    Generate a random variable $\xi_t$.
    Compute a stochastic gradient $\nabla f(w_t; \xi_t)$.
    Update the new iterate $w_{t+1} = w_t - \eta_t \nabla f(w_t; \xi_t)$.
end for
```

The sequence of random variables $\{\xi_t\}_{t\geq 0}$ is assumed to be i.i.d.[1] Let us introduce our key assumption that each realization $\nabla f(w;\xi)$ is an L-smooth function.

Assumption 2 (L-smooth)**.** *$f(w;\xi)$ is L-smooth for every realization of ξ, i.e., there exists a constant $L > 0$ such that, $\forall w, w' \in \mathbb{R}^d$,*

$$\|\nabla f(w;\xi) - \nabla f(w';\xi)\| \leq L\|w - w'\|. \tag{6}$$

Assumption 2 implies that F is also L-smooth. Then, by the property of L-smooth function (in (Nesterov, 2004)), we have, $\forall w, w' \in \mathbb{R}^d$,

$$F(w) \leq F(w') + \langle \nabla F(w'), (w - w') \rangle + \frac{L}{2}\|w - w'\|^2. \tag{7}$$

The following additional convexity assumption can be made, as it holds for many problems arising in machine learning.

[1] Independent and identically distributed.

Assumption 3. *$f(w;\xi)$ is convex for every realization of ξ, i.e., $\forall w, w' \in \mathbb{R}^d$,*

$$f(w;\xi) - f(w';\xi) \geq \langle \nabla f(w';\xi), (w - w') \rangle.$$

We first derive our analysis under Assumptions 2, and 3 and then we derive weaker results under only Assumption 2.

2.1. Convergence With Probability One

As discussed in the introduction, under Assumptions 2 and 3 we can now derive a bound on $\mathbb{E}\|\nabla f(w;\xi)\|^2$.

Lemma 1. *Let Assumptions 2 and 3 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq 4L[F(w) - F(w_*)] + N, \qquad (8)$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Using Lemma 1 and Super Martingale Convergence Theorem (Bertsekas, 2015) (Lemma 4 in the supplementary material), we can provide the sufficient condition for almost sure convergence of Algorithm 1 in the strongly convex case without assuming any bounded gradients.

Theorem 1 (Sufficient conditions for almost sure convergence)**.** *Let Assumptions 1, 2 and 3 hold. Consider Algorithm 1 with a stepsize sequence such that*

$$0 < \eta_t \leq \frac{1}{2L}, \sum_{t=0}^{\infty} \eta_t = \infty \text{ and } \sum_{t=0}^{\infty} \eta_t^2 < \infty.$$

Then, the following holds w.p.1 (almost surely)

$$\|w_t - w_*\|^2 \to 0.$$

Note that the classical SGD proposed in (Robbins & Monro, 1951) has learning rate satisfying conditions

$$\sum_{t=0}^{\infty} \eta_t = \infty \text{ and } \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

However, the original analysis is performed under the bounded gradient assumption, as in (5). In Theorem 1, on the other hand, we do not use this assumption, but instead assume Lipschitz smoothness and convexity of the function realizations, which does not contradict the strong convexity of $F(w)$.

The following result establishes a sublinear convergence rate of SGD.

Theorem 2. *Let Assumptions 1, 2 and 3 hold. Let $E = \frac{2\alpha L}{\mu}$ with $\alpha = 2$. Consider Algorithm 1 with a stepsize sequence such that $\eta_t = \frac{\alpha}{\mu(t+E)} \leq \eta_0 = \frac{1}{2L}$. The expectation $\mathbb{E}[\|w_t - w_*\|^2]$ is at most*

$$\frac{4\alpha^2 N}{\mu^2} \frac{1}{(t - T + E)}$$

for

$$t \geq T = \frac{4L}{\mu} \max\{\frac{L\mu}{N}\|w_0 - w_*\|^2, 1\} - \frac{4L}{\mu}.$$

2.2. Convergence Analysis without Convexity

In this section, we provide the analysis of Algorithm 1 without using Assumption 3, that is, $f(w;\xi)$ is not necessarily convex. We still do not need to impose the bounded stochastic gradient assumption, since we can derive an analogue of Lemma 1, albeit with worse constant in the bound.

Lemma 2. *Let Assumptions 1 and 2 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq 4L\kappa[F(w) - F(w_*)] + N, \qquad (9)$$

where $\kappa = \frac{L}{\mu}$ and $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Based on the proofs of Theorems 1 and 2, we can easily have the following two results (Theorems 3 and 4).

Theorem 3 (Sufficient conditions for almost sure convergence)**.** *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 1 with the definition of the step size replaced by $0 < \eta_t \leq \frac{1}{2L\kappa}$ with $\kappa = \frac{L}{\mu}$.*

Theorem 4. *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 2 with the definition of the step size replaced by $\eta_t = \frac{\alpha}{\mu(t+E)} \leq \eta_0 = \frac{1}{2L\kappa}$ with $\kappa = \frac{L}{\mu}$ and $\alpha = 2$, and all other occurrences of L in E and T replaced by $L\kappa$.*

We compare our result in Theorem 4 with that in (Bottou et al., 2016) in the following remark.

Remark 1. *By strong convexity of F, Lemma 2 implies $\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq 2\kappa^2\|\nabla F(w)\|^2 + N$, for $\forall w \in \mathbb{R}^d$, where $\kappa = \frac{L}{\mu}$ and $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$. We can now substitute the value $M = 2\kappa^2$ into Theorem 4.7 in (Bottou et al., 2016). We observe that the resulting initial learning rate in (Bottou et al., 2016) has to satisfy $\eta_0 \leq \frac{1}{2L\kappa^2}$ while our results allows $\eta_0 = \frac{1}{2L\kappa}$. We are able to achieve this improvement by introducing Assumption 2, which holds for many ML problems.*

Recall that under Assumption 3, our initial learning rate is $\eta_0 = \frac{1}{2L}$ (in Theorem 2). Thus Assumption 3 provides further improvement of the conditions on the learning rate.

3. Asynchronous Stochastic Optimization aka Hogwild!

Hogwild! (Recht et al., 2011) is an asynchronous stochastic optimization method where writes to and reads from vector positions in shared memory can be inconsistent (this corresponds to (13) as we shall see). However, as mentioned in (Mania et al., 2015), for the purpose of analysis the method in (Recht et al., 2011) performs single vector entry updates that are randomly selected from the non-zero entries of the computed gradient as in (12) (explained later) and requires the assumption of consistent vector reads together with the bounded gradient assumption to prove convergence. Both (Mania et al., 2015) and (De Sa et al., 2015) prove the same result for fixed step size based on the assumption of bounded stochastic gradients in the strongly convex case but now without assuming consistent vector reads and writes. In these works the fixed step size η must depend on σ from the bounded gradient assumption, however, one does not usually know σ and thus, we cannot compute a suitable η a-priori.

As claimed by the authors in (Mania et al., 2015), they can eliminate the bounded gradient assumption in their analysis of Hogwild!, which however was only mentioned as a remark without proof. On the other hand, the authors of recent unpublished work (Leblond et al., 2018) formulate and prove, without the bounded gradient assumption, a precise theorem about the convergence rate of Hogwild! of the form

$$\mathbb{E}[\|w_t - w_*\|^2] \leq (1-\rho)^t (2\|w_0 - w_*\|^2) + b,$$

where ρ is a function of several parameters but independent of the fixed chosen step size η and where b is a function of several parameters and has a linear dependency with respect to the fixed step size, i.e., $b = O(\eta)$.

In this section, we discuss the convergence of Hogwild! with **diminishing** stepsize where writes to and reads from vector positions in shared memory can be **inconsistent**. This is a slight modification of the original Hogwild! where the stepsize is fixed. In our analysis we also **do not use the bounded gradient assumption** as in (Leblond et al., 2018). Moreover, (a) we focus on solving the **more general problem** in (1), while (Leblond et al., 2018) considers the specific case of the "finite-sum" problem in (2), and (b) we show that our analysis generalizes to the **non-convex case**, i.e., we do not need to assume functions $f(w;\xi)$ are convex (we only require $F(w) = \mathbb{E}[f(w;\xi)]$ to be strongly convex) as opposed to the assumption in (Leblond et al., 2018).

3.1. Recursion

We first formulate a general recursion for w_t to which our analysis applies, next we will explain how the different variables in the recursion interact and describe two special cases, and finally we present pseudo code of the algorithm using the recursion.

The recursion explains which positions in w_t should be updated in order to compute w_{t+1}. Since w_t is stored in shared memory and is being updated in a possibly non-consistent way by multiple cores who each perform recursions, the shared memory will contain a vector w whose entries represent a mix of updates. That is, before performing the computation of a recursion, a core will first read w from shared memory, however, while reading w from shared memory, the entries in w are being updated out of order. The final vector $\hat{w}_t$ read by the core represents an aggregate of a mix of updates in previous iterations.

The general recursion is defined as follows: For $t \geq 0$,

$$w_{t+1} = w_t - \eta_t d_{\xi_t} S^{\xi_t}_{u_t} \nabla f(\hat{w}_t; \xi_t), \tag{10}$$

where

- $\hat{w}_t$ represents the vector used in computing the gradient $\nabla f(\hat{w}_t;\xi_t)$ and whose entries have been read (one by one) from an aggregate of a mix of previous updates that led to w_j, $j \leq t$, and
- the $S^{\xi_t}_{u_t}$ are diagonal 0/1-matrices with the property that there exist real numbers d_ξ satisfying

$$d_\xi \mathbb{E}[S^\xi_u|\xi] = D_\xi, \tag{11}$$

where the expectation is taken over u and D_ξ is the diagonal 0/1 matrix whose 1-entries correspond to the non-zero positions in $\nabla f(w;\xi)$, i.e., the i-th entry of D_ξ's diagonal is equal to 1 if and only if there exists a w such that the i-th position of $\nabla f(w;\xi)$ is non-zero.

The role of matrix $S^{\xi_t}_{u_t}$ is that it filters which positions of gradient $\nabla f(\hat{w}_t;\xi_t)$ play a role in (10) and need to be computed. Notice that D_ξ represents the support of $\nabla f(w;\xi)$; by $|D_\xi|$ we denote the number of 1s in D_ξ, i.e., $|D_\xi|$ equals the size of the support of $\nabla f(w;\xi)$.

We will restrict ourselves to choosing (i.e., fixing a-priori) *non-empty* matrices S^ξ_u that "partition" D_ξ in D approximately "equally sized" S^ξ_u:

$$\sum_u S^\xi_u = D_\xi,$$

where each matrix S^ξ_u has either $\lfloor |D_\xi|/D \rfloor$ or $\lceil |D_\xi|/D \rceil$ ones on its diagonal. We uniformly choose one of the matrices $S^{\xi_t}_{u_t}$ in (10), hence, d_ξ equals the number of matrices S^ξ_u, see (11).

In other to explain recursion (10) we first consider two special cases. For $D = \bar{\Delta}$, where

$$\bar{\Delta} = \max_\xi \{|D_\xi|\}$$

represents the maximum number of non-zero positions in any gradient computation $f(w;\xi)$, we have that for all ξ, there are exactly $|D_\xi|$ diagonal matrices S^ξ_u with a single 1 representing each of the elements in D_ξ. Since

$p_\xi(u) = 1/|D_\xi|$ is the uniform distribution, we have $\mathbb{E}[S^\xi_u|\xi] = D_\xi/|D_\xi|$, hence, $d_\xi = |D_\xi|$. This gives the recursion

$$w_{t+1} = w_t - \eta_t |D_\xi| [\nabla f(\hat{w}_t; \xi_t)]_{u_t}, \qquad (12)$$

where $[\nabla f(\hat{w}_t; \xi_t)]_{u_t}$ denotes the u_t-th position of $\nabla f(\hat{w}_t; \xi_t)$ and where u_t is a uniformly selected position that corresponds to a non-zero entry in $\nabla f(\hat{w}_t; \xi_t)$.

At the other extreme, for $D = 1$, we have exactly one matrix $S^\xi_1 = D_\xi$ for each ξ, and we have $d_\xi = 1$. This gives the recursion

$$w_{t+1} = w_t - \eta_t \nabla f(\hat{w}_t; \xi_t). \qquad (13)$$

Recursion (13) represents Hogwild!. In a single-core setting where updates are done in a consistent way and $\hat{w}_t = w_t$ yields SGD.

Algorithm 2 gives the pseudo code corresponding to recursion (10) with our choice of sets S^ξ_u (for parameter D).

Algorithm 2 Hogwild! general recursion

```
Input: $w_0 \in \mathbb{R}^d$
for $t = 0, 1, 2, \dots$ in parallel do
    read each position of shared memory $w$ denoted by $\hat{w}_t$ (each position read is atomic)
    draw a random sample $\xi_t$ and a random "filter" $S^{\xi_t}_{u_t}$
    for positions $h$ where $S^{\xi_t}_{u_t}$ has a 1 on its diagonal do
        compute $g_h$ as the gradient $\nabla f(\hat{w}_t; \xi_t)$ at position $h$
        add $\eta_t d_{\xi_t} g_h$ to the entry at position $h$ of $w$ in shared memory (each position update is atomic)
    end for
end for
```

3.2. Analysis

Besides Assumptions 1, 2, and for now 3, we assume the following assumption regarding a parameter τ, called the delay, which indicates which updates in previous iterations have certainly made their way into shared memory w.

Assumption 4 (Consistent with delay τ). *We say that shared memory is consistent with delay τ with respect to recursion (10) if, for all t, vector $\hat{w}_t$ includes the aggregate of the updates up to and including those made during the $(t-\tau)$-th iteration (where (10) defines the $(t+1)$-st iteration). Each position read from shared memory is atomic and each position update to shared memory is atomic (in that these cannot be interrupted by another update to the same position).*

In other words in the $(t+1)$-th iteration, $\hat{w}_t$ equals $w_{t-\tau}$ plus some subset of position updates made during iterations $t-\tau, t-\tau+1, \dots, t-1$. We assume that there exists a constant delay τ satisfying Assumption 4.

The supplementary material proves the following theorem where

$$\bar{\Delta}_D \overset{\text{def}}{=} D \cdot \mathbb{E}[\lceil |D_\xi|/D \rceil].$$

Theorem 5. *Suppose Assumptions 1, 2, 3 and 4 and consider Algorithm 2 for sets S^ξ_u with parameter D. Let $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4 \leq \alpha_t \leq \alpha$ and $E = \max\{2\tau, \frac{4L\alpha D}{\mu}\}$. Then, the expected number of single vector entry updates after t iterations is equal to*

$$t' = t\bar{\Delta}_D / D$$

and expectations $\mathbb{E}[\|\hat{w}_t - w_*\|^2]$ and $\mathbb{E}[\|w_t - w_*\|^2]$ are at most

$$\frac{4\alpha^2 DN}{\mu^2} \frac{t}{(t+E-1)^2} + O\left(\frac{\ln t}{(t+E-1)^2}\right).$$

In terms of t', the expected number single vector entry updates after t iterations, $\mathbb{E}[\|\hat{w}_t - w_*\|^2]$ and $\mathbb{E}[\|w_t - w_*\|^2]$ are at most

$$\frac{4\alpha^2 \bar{\Delta}_D N}{\mu^2} \frac{1}{t'} + O\left(\frac{\ln t'}{t'^2}\right).$$

Remark 2. *In (12) $D = \bar{\Delta}$, hence, $\lceil |D_\xi|/D \rceil = 1$ and $\bar{\Delta}_D = \bar{\Delta} = \max_\xi\{|D_\xi|\}$. In (13) $D = 1$, hence, $\bar{\Delta}_D = \mathbb{E}[|D_\xi|]$. This shows that the upper bound in Theorem 5 is better for (13) with $D = 1$. If we assume no delay, i.e. $\tau = 0$, in addition to $D = 1$, then we obtain SGD. Theorem 2 shows that, measured in t', we obtain the upper bound*

$$\frac{4\alpha^2_{SGD} \bar{\Delta}_D N}{\mu^2} \frac{1}{t'}$$

with $\alpha_{SGD} = 2$ as opposed to $\alpha \geq 4$.

With respect to parallelism, SGD assumes a single core, while (13) and (12) allow multiple cores. Notice that recursion (12) allows us to partition the position of the shared memory among the different processor cores in such a way that each partition can only be updated by its assigned core and where partitions can be read by all cores. This allows optimal resource sharing and could make up for the difference between $\bar{\Delta}_D$ for (12) and (13). We hypothesize that, for a parallel implementation, D equal to a fraction of $\bar{\Delta}$ will lead to best performance.

Remark 3. *Surprisingly, the leading term of the upper bound on the convergence rate is independent of delay τ. On one hand, one would expect that a more recent read which contains more of the updates done during the last τ iterations will lead to better convergence. When inspecting the second order term in the proof in the supplementary material, we do see that a smaller τ (and/or smaller sparsity) makes the convergence rate smaller. That is, asymptotically t should be large enough as a function of τ (and other parameters) in order for the leading term to dominate.*

Nevertheless, in asymptotic terms (for larger t) the dependence on τ is not noticeable. In fact, the supplementary material shows that we may allow τ to be a monotonic increasing function of t with

$$\frac{2L\alpha D}{\mu} \leq \tau(t) \leq \sqrt{t \cdot L(t)},$$

where $L(t) = \frac{1}{\ln t} - \frac{1}{(\ln t)^2}$ (this will make $E = \max\{2\tau(t), \frac{4L\alpha D}{\mu}\}$ also a function of t). The leading term of the convergence rate does not change while the second order terms increase to $O(\frac{1}{t \ln t})$. We show that, for

$$t \geq T_0 = \exp[2\sqrt{\Delta}(1 + \frac{(L+\mu)\alpha}{\mu})],$$

where $\Delta = \max_i \mathbb{P}(i \in D_\xi)$ measures sparsity, the higher order terms that contain $\tau(t)$ (as defined above) are at most the leading term.

Our intuition behind this phenomenon is that for large τ, all the last τ iterations before the t-th iteration use vectors $\hat{w}_j$ with entries that are dominated by the aggregate of updates that happened till iteration $t-\tau$. Since the average sum of the updates during the last τ iterations is equal to

$$-\frac{1}{\tau} \sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_t) \tag{14}$$

and all $\hat{w}_j$ look alike in that they mainly represent learned information before the $(t-\tau)$-th iteration, (14) becomes an estimate of the expectation of (14), i.e.,

$$\sum_{j=t-\tau}^{t-1} \frac{-\eta_j}{\tau} \mathbb{E}[d_{\xi_j} S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_t)] = \sum_{j=t-\tau}^{t-1} \frac{-\eta_j}{\tau} \nabla F(\hat{w}_j). \tag{15}$$

This looks like GD which in the strong convex case has convergence rate $\leq c^{-t}$ for some constant $c > 1$. This already shows that larger τ could help convergence as well. However, estimate (14) has estimation noise with respect to (15) which explains why in this thought experiment we cannot attain c^{-t} but can only reach a much smaller convergence rate of e.g. $O(1/t)$ as in Theorem 5.

Experiments in Section 4 confirm our analysis.

Remark 4. *The higher order terms in the proof in the supplementary material show that, as in Theorem 2, the expected convergence rate in Theorem 5 depends on $\|w_0 - w_*\|^2$. The proof shows that, for*

$$t \geq T_1 = \frac{\mu^2}{\alpha^2 N D}\|w_0 - w_*\|^2,$$

the higher order term that contains $\|w_0 - w_*\|^2$ is at most the leading term. This is comparable to T in Theorem 2 for SGD.

Remark 5. *Step size $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4 \leq \alpha_t \leq \alpha$ can be chosen to be fixed during periods whose ranges exponentially increase. For $t + E \in [2^h, 2^{h+1})$ we define $\alpha_t = \frac{4(t+E)}{2^h}$. Notice that $4 \leq \alpha_t < 8$ which satisfies the conditions of Theorem 5 for $\alpha = 8$. This means that we can choose*

$$\eta_t = \frac{\alpha_t}{\mu(t+E)} = \frac{4}{\mu 2^h}$$

as step size for $t+E \in [2^h, 2^{h+1})$. This choice for η_t allows changes in η_t to be easily synchronized between cores since these changes only happen when $t + E = 2^h$ for some integer h. That is, if each core is processing iterations at the same speed, then each core on its own may reliably assume that after having processed $(2^h - E)/P$ iterations the aggregate of all P cores has approximately processed $2^h - E$ iterations. So, after $(2^h - E)/P$ iterations a core will increment its version of h to $h+1$. This will introduce some noise as the different cores will not increment their h versions at exactly the same time, but this only happens during a small interval around every $t + E = 2^h$. This will occur rarely for larger h.

3.3. Convergence Analysis without Convexity

In the supplementary material, we also show that the proof of Theorem 5 can easily be modified such that Theorem 5 with $E \geq \frac{4L\kappa\alpha D}{\mu}$ also holds in the non-convex case of the component functions, i.e., we do not need Assumption 3. Note that this case is not analyzed in (Leblond et al., 2018).

Theorem 6. *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 5 with $E \geq \frac{4L\kappa\alpha D}{\mu}$ for $\kappa = \frac{L}{\mu}$.*

4. Numerical Experiments

For our numerical experiments, we consider the finite sum minimization problem in (2). We consider ℓ_2-regularized logistic regression problems with

$$f_i(w) = \log(1 + \exp(-y_i \langle x_i, w \rangle)) + \frac{\lambda}{2}\|w\|^2,$$

where the penalty parameter λ is set to $1/n$, a widely-used value in literature (Le Roux et al., 2012).

We conducted experiments on a single core for Algorithm 2 on two popular datasets `ijcnn1` ($n = 91,701$ training data) and `covtype` ($n = 406,709$ training data) from the LIBSVM[2] website. Since we are interested in the expected convergence rate with respect to the number of iterations, respectively number of single position vector updates, we do not need a parallelized multi-core simulation to confirm our analysis. The impact of efficient resource scheduling

[2] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

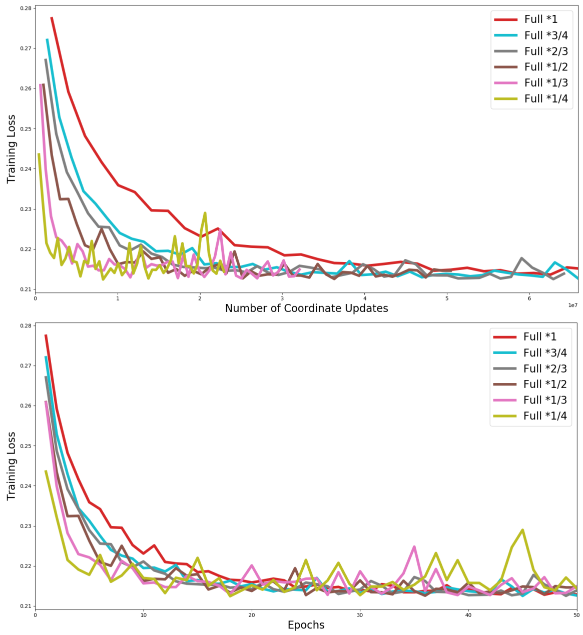

Figure 1: *ijcnn1* for different fraction of non-zero set

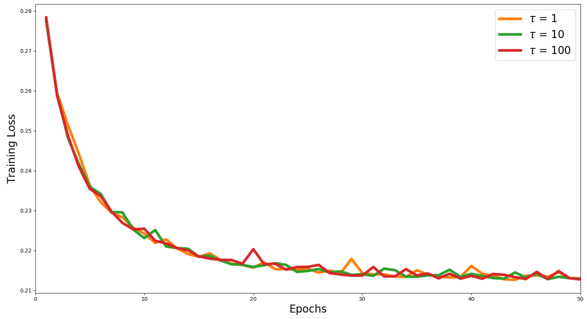

Figure 2: *ijcnn1* for different τ with the whole non-zero set

over multiple cores leads to a performance improvement complementary to our analysis of (10) (which, as discussed, lends itself for an efficient parallelized implementation). We experimented with 10 runs and reported the average results. We choose the step size based on Theorem 5, i.e, $\eta_t = \frac{4}{\mu(t+E)}$ and $E = \max\{2\tau, \frac{16LD}{\mu}\}$. For each fraction $v \in \{1, 3/4, 2/3, 1/2, 1/3, 1/4\}$ we performed the following experiment: In Algorithm 2 we choose each "filter" matrix $S^{\xi_t}_{u_t}$ to correspond with a random subset of size $v|D_{\xi_t}|$ of the non-zero positions of D_{ξ_t} (i.e., the support of the gradient corresponding to ξ_t). In addition we use $\tau = 10$. For the two datasets, Figures 1 and 3 plot the training loss for each fraction with $\tau = 10$. The top plots have t', the number of coordinate updates, for the horizontal axis. The bottom plots have the number of epochs, each epoch counting n iterations, for the horizontal axis. The results show that each fraction shows a sublinear expected convergence rate of $O(1/t')$; the smaller fractions exhibit larger deviations but do seem to converge faster to the minimum solution.

In Figures 2 and 4, we show experiments with different values of $\tau \in \{1, 10, 100\}$ where we use the whole non-zero set of gradient positions (i.e., $v = 1$) for the update. Our analysis states that, for $t = 50$ epochs times n iterations per epoch, τ can be as large as $\sqrt{t \cdot L(t)} = 524$ for `ijcnn1` and 1058 for `covtype`. The experiments indeed show that $\tau \leq 100$ has little effect on the expected convergence rate.

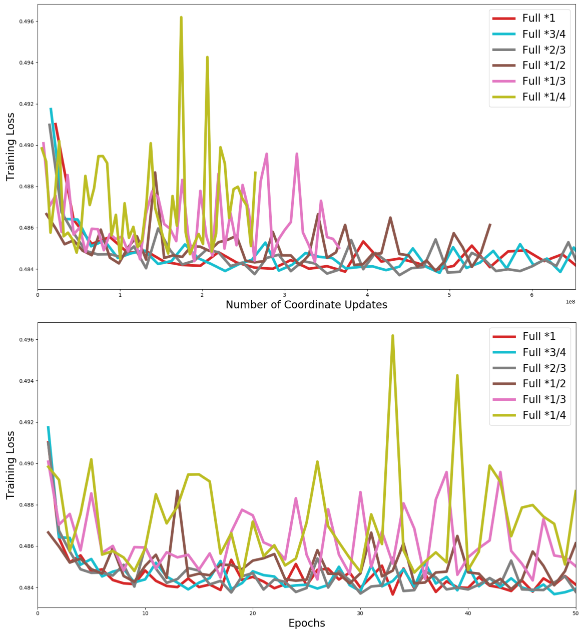

Figure 3: *covtype* for different fraction of non-zero set

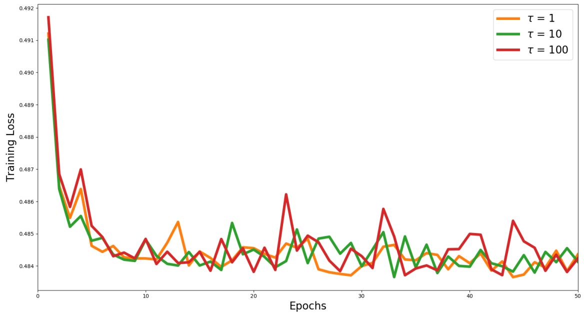

Figure 4: *covtype* for different τ with the whole non-zero set

5. Conclusion

We have provided the analysis of stochastic gradient algorithms with diminishing step size in the strongly convex case under the condition of Lipschitz continuity of the individual function realizations, but without requiring any bounds on the stochastic gradients. We showed almost sure convergence of SGD and provided sublinear upper bounds for the expected convergence rate of a general recursion which includes Hogwild! for inconsistent reads and writes as a special case. We also provided new intuition which will help understanding convergence as observed in practice.

Acknowledgement

The authors would like to thank the reviewers for useful suggestions which helped to improve the exposition in the paper. The authors also would like to thank Francesco Orabona for his valuable comments and suggestions.

Lam M. Nguyen was partially supported by NSF Grants CCF 16-18717. Phuong Ha Nguyen and Marten van Dijk were supported in part by AFOSR MURI under award number FA9550-14-1-0351. Katya Scheinberg was partially supported by NSF Grants CCF 16-18717 and CCF 17-40796. Martin Takáč was supported by U.S. National Science Foundation, under award number NSF:CCF:1618717, NSF:CMMI:1663256 and NSF:CCF:1740796.

References

Bertsekas, Dimitri P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey, 2015.

Bottou, Léon, Curtis, Frank E, and Nocedal, Jorge. Optimization methods for large-scale machine learning. *arXiv:1606.04838*, 2016.

De Sa, Christopher M, Zhang, Ce, Olukotun, Kunle, and Ré, Christopher. Taming the wild: A unified analysis of hogwild-style algorithms. In *Advances in neural information processing systems*, pp. 2674–2682, 2015.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pp. 1646–1654, 2014.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, 2nd edition, 2009.

Hazan, Elad and Kale, Satyen. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15:2489–2512, 2014.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pp. 315–323, 2013.

Le Roux, Nicolas, Schmidt, Mark, and Bach, Francis. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pp. 2663–2671, 2012.

Leblond, Remi, Pedregosa, Fabian, and Lacoste-Julien, Simon. Improved asynchronous parallel optimization analysis for stochastic incremental methods. *arXiv:1801.03749*, 2018.

Mania, Horia, Pan, Xinghao, Papailiopoulos, Dimitris, Recht, Benjamin, Ramchandran, Kannan, and Jordan, Michael I. Perturbed Iterate Analysis for Asynchronous Stochastic Optimization. *arXiv preprint arXiv:1507.06970*, 2015.

Moulines, Eric and Bach, Francis R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 451–459. Curran Associates, Inc., 2011.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM J. on Optimization*, 19(4): 1574–1609, January 2009. ISSN 1052-6234. doi: 10.1137/070704277.

Nesterov, Yurii. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004. ISBN 1-4020-7553-7.

Nguyen, Lam, Liu, Jie, Scheinberg, Katya, and Takáč, Martin. SARAH: A novel method for machine learning problems using stochastic recursive gradient. *ICML*, 2017.

Nocedal, Jorge and Wright, Stephen J. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

Rakhlin, Alexander, Shamir, Ohad, and Sridharan, Karthik. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*. icml.cc / Omnipress, 2012.

Recht, Benjamin, Re, Christopher, Wright, Stephen, and Niu, Feng. Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 693–701. Curran Associates, Inc., 2011.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 807–814, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273598.