An Investigation into Hogwild!

Abhijit Chowdhary
ac6361@nyu.edu

New York University — May 21, 2020

Contents

1 Introduction

Despite being the subject of much modern excitement, the ideas of gradient descent date all the way back to Cauchy in 1847. And although it's main application has been in the solution of recent big-data optimization problems, stochastic gradient has been around since the 1940s, formally by Robbins and Monro in 1951. In the last two decades, however, modern hardware has begun to see a tapering off of Moore's law, and has begun to expand out in a distributed fashion with multicore processors and GPUs; naturally the question becomes: In order to take advantage of the strengths of modern hardware, how can we parallelize a stochastic gradient method?

2 HOGWILD!

Prior to 2011, parallel stochastic gradient methods had been introduced, but most suffered from poor scaling due to the necessity of locks. A naive implementation could look like:

Algorithm 1 Very Naive Parallel Stochastic Gradient

Require: Number of data points N, seperable loss function $f = \sum_{e \in E} f_e(x_e)$, Initial x.

for epoch $= 1 \to$ MAX_EPOCHS **do**
    #pragma omp parallel for
    for $k = 1 \to N$ **do**
        Choose i uniformly from $\{1, \ldots, |E|\}$.
        #pragma omp critical
            Read current parameters x.
            Compute $\nabla f_i(x)$.
            $x \leftarrow x - \eta \nabla f_i(x)$.
    end for
end for

Note that the version presented above is one with a fixed number of iterations, as the discussion of stopping criteria seems to be similar to that of the stochastic gradient method, and for extremely large data sets, is often heuristic. But it's clear here that such an algorithm would only effectively be parallelizing the unform sample of i in $\{1, \ldots, |E|\}$, and it's overall parallel efficiency would likely be poor. Technically, you can improve the above by replacing the critical section with selective locks on components of x based on the sparsity pattern of $\nabla f_i(x)$, but because the process of acquiring locks is much more expensive than floating point arithmetic, this helps little.

However, in 2011, the article "HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent" by Niu et al. [NRRW11] proposed a very simple solution to this problem. Remove the locks![1]

Algorithm 2 HOGWILD!: Asynchronous Stochastic Gradient with replacement

Require: Number of data points N, seperable loss function $f = \sum_{e \in E} f_e(x_e)$, Initial x.

[1] Apparently this was discovered by accident by Feng Niu, one of the original paper's authors, when he was debugging stochastic gradient method code. I wish my troubleshooting was nearly as effective... [Rec14]

for epoch = 1 $\to$ MAX_EPOCHS **do**
  #pragma omp parallel for
  for $k = 1 \to N$ **do**
    Choose i uniformly from $\{1, \ldots, |E|\}$.
    Read current parameters x.
    Compute $\nabla f_i(x)$.
    $x \leftarrow x - \eta \nabla f_i(x)$. ▷ Must be done atomically
  end for
end for

and should we want to sample without replacement the algorithm is easily modified to:

Algorithm 3 HOGWILD!: Asynchronous Stochastic Gradient without replacement

Require: Number of data points N, seperable loss function $f = \sum_{e \in E} f_e(x_e)$, Initial x.
for epoch = 1 $\to$ MAX_EPOCHS **do**
  Let P be a random permutation of $\{1, \ldots, |E|\}$. ▷ i.e. a Fisher-Yates Shuffle.
  #pragma omp parallel for
  for $k = 1 \to N$ **do**
    $i \leftarrow P[k]$.
    Read current parameters x.
    Compute $\nabla f_i(x)$.
    $x \leftarrow x - \eta \nabla f_i(x)$. ▷ Must be done atomically
  end for
end for

It should be noted that although the formal OMP locks have been removed, atomic operations are still required in order to prevent mutual exclusion. However, no guards have been placed to prevent a thread from overwriting another's computation midway through, and it's not obvious as to why such a race condition wouldn't destroy the performance of the Stochastic Gradient method. However, with certain assumptions one can show that HOGWILD! behaves roughly like a noisy stochastic gradient method, and thus shares its convergence properties.

3 Convergence Analysis

Given that the result of HOGWILD!, in the absence of noise generated by the asynchronous updates of x, henceforth denoted *asynchronous noise*, is equivalent to a stochastic gradient method, we should expect convergence rates similar to those of stochastic gradient, should noise be small. Take for example the typical linear least squares loss function $f(x) = \frac{1}{2n}\|Dx - b\|_2^2$, where $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n \times n}$ a diagonal matrix. Writing this as a sum of each data entry:

$$f(x) = \frac{1}{2n}\sum_{k=1}^{n}(D_{ii}x_i - b_i)^2 = \frac{1}{n}\sum_{k=1}^{n} f_i(x) \implies (\nabla f_i(x))_k = \begin{cases} D_{ii}(D_{ii}x_i - b_i), & k = i \\ 0, & k = 0 \end{cases}$$

Because our $\nabla f_i(x)$'s only have a single entry in their own component, we can see that as long as no other thread is working on component i simultaneously, no asynchronous noise will be generated; should we use HOGWILD! without replacement then this is guaranteed. In the original paper [NRRW11], sparsity of the vector $\nabla f_i(x)$ was required in order to guarantee convergence, but as we'll see later, this isn't always necessary.

As a baseline of comparison, we first state a result on the convergence of the stochastic gradient method:

Theorem 3.1. *(Convergence of the Stochastic Gradient Method [BCN16]) Let $F : \mathbb{R}^d \to \mathbb{R}$ be an objective function we're seeking to minimize. We can write this as either an expected risk $F(x) = \mathbb{E}_\xi f(x, \xi)$, or an empirical risk $F(x) = \frac{1}{n}\sum_{k=1}^n f_k(x)$. Under the assumptions:*

(1) F is continuously differentiable and ∇F is Lipschitz continuous with Lipshitz constant L, i.e.

$$\left\|\nabla F(x) - \nabla F(y)\right\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^d$$

(2) F is strongly convex with constant c, i.e.

$$F(x) \geq F(y) + \nabla F(y)^T(x - y) + \frac{1}{2}c\|x - y\|^2, \forall x, y \in \mathbb{R}^d$$

(3) F is bounded below over the region explored by stochastic gradient method.

(4) In expectation, the vector $-\nabla f(x_k, \xi_k)$ is a descent direction for F with norm bounded by it's own norm. That is: $\exists \mu_G \geq \mu > 0$ such that $\forall k \in \mathbb{N}$:

$$\nabla F(x_k)^T\mathbb{E}\left[\nabla f(x_k, \xi_k)\right] \geq \mu\left\|\nabla F(x_k)\right\|^2 \text{ and } \left\|\mathbb{E}\left[\nabla f(x_k, \xi_k)\right]\right\| \geq \mu_G\left\|\nabla F(x_k)\right\|$$

(5) $\exists M, M_V \geq 0$ such that $\forall k \in \mathbb{N}$:

$$\mathrm{Var}\left[\nabla f(x_k, \xi_k)\right] \leq M + M_V\left\|\nabla F(x_k)\right\|^2$$

Then, assuming a fixed stepsize α (a.k.a. learning rate), satisfying $\alpha \in (0, \mu/LM_g]$, we have:

$$\mathbb{E}\left[F(x_k) - F_*\right] \leq \frac{\alpha LM}{2c\mu} + (1 - \alpha c\mu)^{k-1}\left(F(x_1) - F_* - \frac{\alpha LM}{2c\mu}\right)$$

and if we instead choose α diminishing, i.e. let $\alpha_k = \frac{\beta}{\gamma + k}$ where $\beta > 1/c\mu, \gamma > 0$ are chose such that $\alpha_1 \leq \mu/LM_G$, then:

$$\mathbb{E}\left[F(x_k) - F_*\right] \leq \frac{1}{\gamma + k}\max\left\{\frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma + 1)(F(x) - F_*)\right\}$$

The result is technical, and very long to prove, so I just refer to the article [BCN16] for it. Regardless, the last result in the above theorem is what we strive for in *Hogwild*!: convergence in $\mathcal{O}(1/k)$ time.

3.1 Theoretical Results

Hogwild!'s original article [NRRW11] manages to prove that the asynchronous noise generated by Hogwild!, under certain sparsity assumptions of the f_i's, converged at the same rate as seen in the stochastic gradient method. However, a newer article from 2015 by De Sa et al. [SZOR15] presented a new framework in the context of martingale theory, which drops said sparsity assumptions and generalizes to certain non-convex formualations. Given that I just learned about martingale theory essentially two weeks ago in Basic Probability, and that much of my synthetic tests are on dense datasets, I feel compelled to present this argument instead.

The following is a cleaned up summary of the arguments presented in the article, except I focus on just demonstrating how martingale theory can be used to generalize convergence arguments for a sequential stochastic gradient method to the asynchronous case, and cut out the generalization required for the non-convex situation.

3.1.1 Initial Machinery

First, recall the definition of a martingale:

Definition 3.1. *[JP04] A sequence of random variables $(X_n)_{n\geq 0}$ is called a martingale, or an $(\mathcal{F}_n)$-martingale, if*

(i) $\mathbb{E}\left[|X_n|\right] < \infty, \forall n.$

(ii) X_n *is* $\mathcal{F}_n$ *measurable,* $\forall n$.

(iii) $\mathbb{E}\left[X_n \mid \mathcal{F}_m\right] = X_m$ *a.s.,* $\forall m \leq n$.

furthermore a supermartingale (submartingale) is one satisfying (i), (ii) exactly, and (iii) with $\leq$ $(\geq)$ *instead.*

Using this, the idea is to model our convergence with a non-negative supermartingale $W_t(x_t, \ldots, x_0)$ which is a function of the previous stochastic gradient iterates. These W_t, when used in the theory later, will be associated with specific stochastic algorithms, as an example below we'll see it applied to serial stochastic gradient. However, given such a supermartingale, and given a bounded stopping time B (in literature known as a horizon), if our stochastic gradient iterates are written as $x_{t+1} = x_t - \eta\nabla f_t(x_t)$, where η is the learning rate, and f_t denotes the random function chosen at time step t, then we see that condition (iii) in the above definition implies that:

$$W_{t+1}(x_t - \eta\nabla f_t(x_t), x_t, \ldots, x_0) \leq W_t(x_t, \ldots, x_0), \forall t \leq B$$

which certainly makes sense if our $-\nabla f_t(x_t)$ is a sufficient search direction. Furthermore, letting our success region be denoted as $S = B_\varepsilon(x^*)$, where x^* is the minimizer of our optimization problem, if we impose that if $x_t \notin S, \forall t \leq T$ then:

$$W_T(x_T, \ldots, x_0) \geq T$$

then we call W_t a *rate supermartingale*. To simplify notation, let F_t be the event where $\nexists t \leq T$ such that $x_t \in S$.

A good example of the power of this machinery is proving a convergence bound on the serial version of stochastic gradient: using (iii) of definition 3.1, one can see that considering F_T:

$$\mathbb{E}\left[W_0(x_0)\right] \underbrace{\geq} \mathbb{E}\left[W_T\right] = \mathbb{P}\left[F_t\right] \underbrace{\mathbb{P}\left[F_T\right]\mathbb{E}\left[W_T|F_T\right]}_{W_T \geq T} + \underbrace{\mathbb{P}\left[F_T^c\right]\mathbb{E}\left[W_T|F_T^c\right]}_{W_T \geq 0} \geq \mathbb{P}\left[F_T\right]T$$

where the first inequality is by Doob's optional sampling theorem. Thus for a simple serial stochastic gradient method: $\mathbb{P}\left[F_t\right] = \mathbb{E}\left[W_0\right]/T$.

Now that we can characterize serial stochastic gradient in this model, we need a method to analyze asynchronous noise. Recall that since we've guaranteed in the description of Hogwild! that writes to the iteration variable x_t are done atomically, the only race condition possible is when updates to entries of x_t are interleaved with either the other thread's updates or its reads on x_t.

Indeed, when going to update the ith component of x_{t+1}, the variable used to compute the gradient may have long since changed, making our iteration look more like $x_{t+1} = x_t - \nabla f_t(v_t)$ where v_t's entries were the entries of some previous iterate x. Let $\tau_{i,t}$ denote the lag for the update of the ith component of x_{t+1}, i.e. $(v_t)_i = (x_{t-\tau_{i,t}})_i$. Then we recognize that this lag for each component i (supposing the computer hasn't crashed) must be bounded; let τ' be the the maximum of such bounds and let $\tau = \mathbb{E}\left[\tau'\right]$. This is known in literature as the *worst-case expected delay*.

Finally, we need one last definition, mainly one of convinience, to proceed onward. This describes the main conditions upon a rate supermartingale necessary to prove that the asynchronous noise error is irrelevant to the convergence rate.

Definition 3.2. *An algorithm with associated rate supermartingale W is (H, R, ξ)-bounded if the following conditions hold.*

(1) W must be Lipschitz continuous in the current iterate with parameter H, i.e.

$$\left\|W_t(u, x_{t-1}, \ldots, x_0) - W_t(v, x_{t-1}, \ldots, x_0)\right\| \leq H\|u - v\|, \forall t, u, v, x_t, \ldots, x_0.$$

(2) ∇f must be Lipschitz continuous in expectation with parameter R, i.e.

$$\mathbb{E}\left[||\nabla f(x) - \nabla f(y)||\right] \leq R\|u - v\|$$

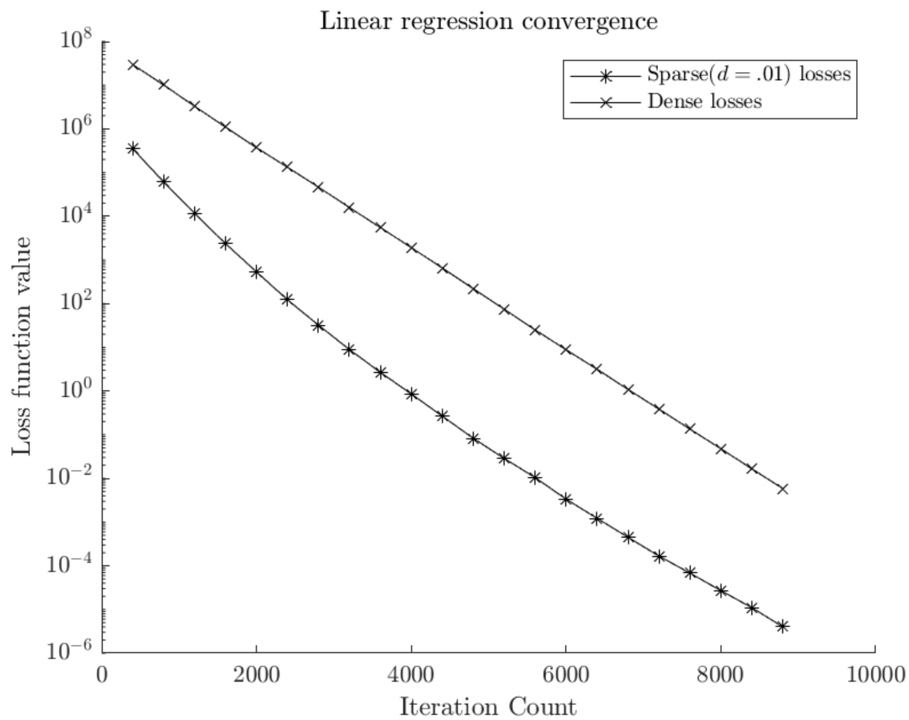
(3) The expected magnitude of the update must be bounded by ξ, i.e.

$$\mathbb{E}\left[||\nabla f(x)||\right] \leq \xi$$

Note that these look very familiar to the conditions in the stochastic gradient method convergence theorem (theorem 3.1).

3.1.2 Convergence of Hogwild!

Now that we have

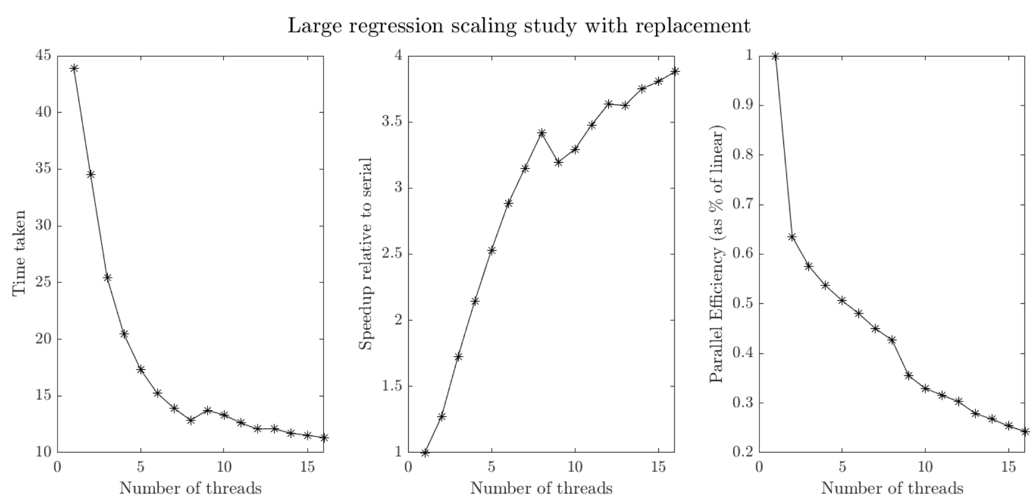


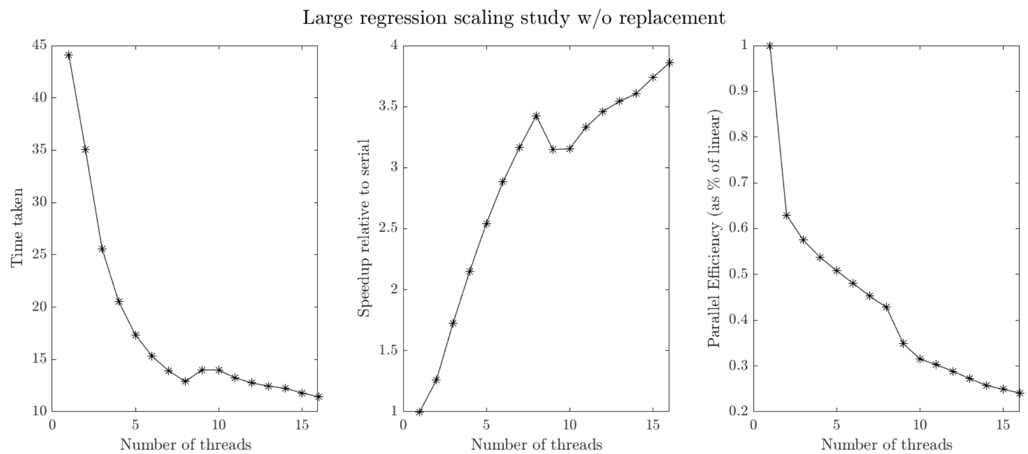
3.2 Numerical Validation

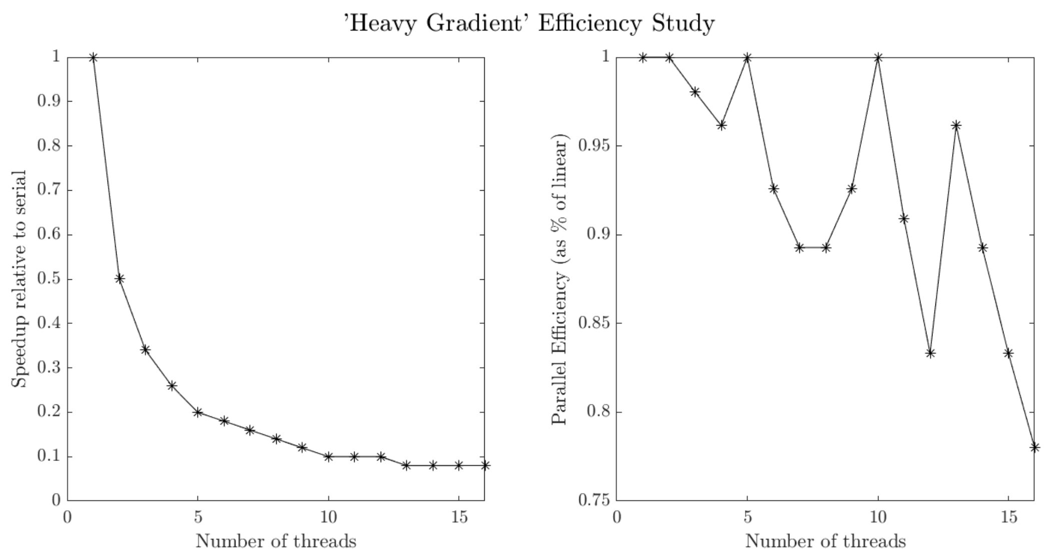
4 Efficiency Analysis

4.1 Theoretical Results

4.2 Numerical Validation







5 Asynchronous Noise Analysis

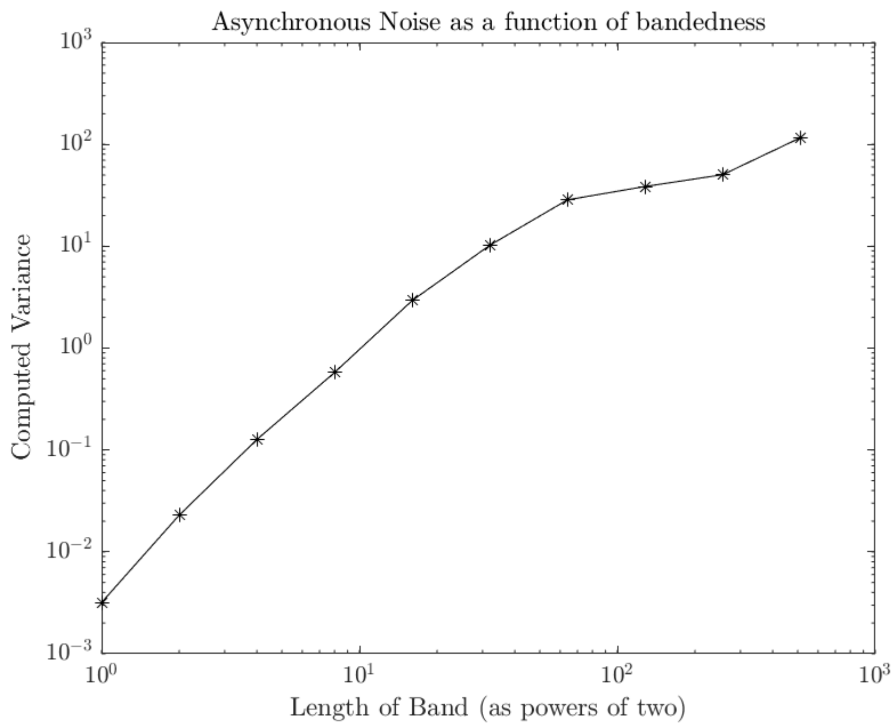
5.1 Quantifying the Error

5.2 Banded Matrix Regression

5.3 Impacts of Noise

5.4 Modern Solutions

6 Conclusions and Future Work


Asynchronous Noise as a function of bandedness
Computed Variance
Length of Band (as powers of two)
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{0}
10^{1}
10^{2}
10^{3}

References

[BCN16] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning, 2016.

[JP04] Jean Jacod and Philip Protter. *Probability Essentials*. Springer Berlin Heidelberg, 2004.

[NRRW11] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent, 2011.

[Rec14] Benjamin Recht. HOGWILD! for machine learning on multicore. `https://youtu.be/l5JqUvTdZts`, June 2014.

[SZOR15] Christopher De Sa, Ce Zhang, Kunle Olukotun, and Christopher Ré. Taming the wild: A unified analysis of hogwild!-style algorithms, 2015.