More about Asynchronous Optimization

Jingchang Liu

June 5, 2017

University of Science and Technology of China

Table of Contents

Asynchronous SCD

SCD

Formulations

- $\min_x f(x)$, $x = x_1 \times x_2 \times \cdots \times x_n$

Coordinate Descent

- Choose index $i_k \in \{1, 2, \cdots, n\}$
- $x^{k+1} \leftarrow x^k - \eta_k e_{i_k} \nabla_{i_k} f\left(x^k\right)$ for some $\eta_k > 0$

Convergence rate

- f is convex: $\mathcal{O}\left(\frac{1}{k}\right)$
- f is strongly convex: $\mathcal{O}\left(C^k\right)$

Prox-SCD

Formulations

- $\min_{x \in \Omega} f(x) + \lambda g(x)$, $g(x) = \sum_{i=1}^{n} g_i(x_i)$

Coordinate Descent

- Choose index $i_k \in \{1, 2, \cdots, n\}$
- $x_{i_k}^{k+1} \leftarrow prox_{g_{i_k}}^{\lambda_i}\left(x_{i_k} - \lambda_i \nabla_{i_k} f\left(x^k\right)\right)$

Convergence rate

- f is convex: $\mathcal{O}\left(\frac{1}{k}\right)$
- f is strongly convex: $\mathcal{O}\left(C^k\right)$

Paper

An Asynchronous Parallel Stochastic Coordinate Descent Algorithm(Ji Liu et.al)

Formulations

- $\min_{x} f(x), x = x_1 \times x_2 \times \cdots \times x_n$

Update in each workers

- Choose i_k from $\{1, 2, \cdots, n\}$ with equal probabilities.
- $x_{i_k}^{k+1} \leftarrow x_{i_k}^{k} - \lambda_k \nabla_{i_k} f(\hat{x}_k)$

AsySPCD(asynchronous stochastic proximal coordinate-descent)

Paper

Asynchronous Stochastic Coordinate Descent: Parallelism and Convergence Properties (Ji Liu & Stephen J. Wright)

Formulations

- $\min_{x \in \Omega} f(x) + \lambda g(x),\ g(x) = \sum_{i=1}^{n} g_i(x_i)$

Update in each workers

- Choose i_k from $\{1, 2, \cdots, n\}$ with equal probabilities.
- $x_{i_k}^{k+1} \leftarrow prox_{g_{i_k}}^{\lambda_k}\left(x_{i_k}^{k} - \lambda_k \nabla_{i_k} f(\hat{x}_k)\right)$

Inconsistent read

Consistent read & Inconsistent read

Consistent read

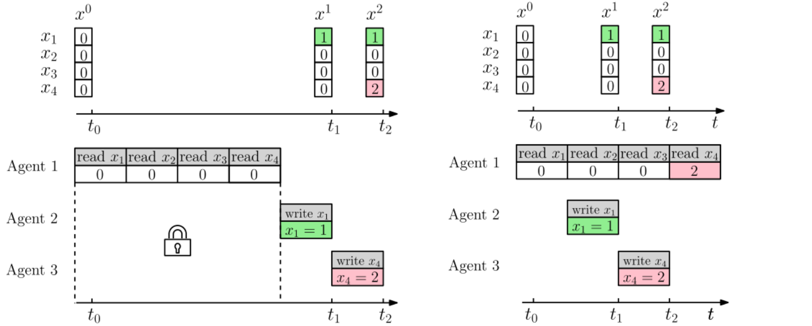
- Whenever a worker read, other workers can not write.

Inconsistent read

- Without a lock to when a worker read x.
- atomic coordinate update: a coordinate is not further broken to smaller components during an update; they are all updated at once.
- $\hat{x}_i^k$ is an ever-existed state of x_i among $x_i^k, \cdots, x_i^{k-\tau}$, that to say $\hat{x}_i^k = x_i^{\underline{d}}$, where $\underline{d} \in \{k, k-1, \cdots, k-\tau\}$
- $J_i(k) \subset \{k-1, \cdots, k-\tau\}$, $\hat{x}_i^k = x_i^k + \sum_{d \in J_i(k)} \left(x_i^d - x_i^{d+1}\right)$
- $J(k) := \bigcup_i J_i(k) \subset \{k-1, \cdots, k-\tau\}$, then

$$\hat{x}^k = x^k + \sum_{d \in J(k)} \left(x^d - x^{d+1}\right)$$

Consistent read & Inconsistent read



(a) Consistent read. While agent 1 reads x in memory, it acquires a global lock.

(b) Inconsistent read. No lock. Agent 1 reads $(0,0,0,2)^T$, a non-existing state of x.

Figure 1: Consistent read versus inconsistent read: A demonstration

Convergence rate

Theorem

Let ρ be a constant that satisfies $\rho > 1 + 4/\sqrt{n}$, and define the quantities θ, θ, and ϕ. Steplength parameter $\gamma > 0$ satisfies several bounds. Then

$$\mathbb{E}\left\|x^{k-1} - \hat{x}^{k}\right\|^2 \leq \rho \mathbb{E}\left\|x^{k} - \hat{x}^{k+1}\right\|^2$$

if the optimal strong convexity property holds,then

$$\mathbb{E}\left\|x^{k} - x^{*}\right\|^2 + 2\gamma\left(\mathbb{E}F\left(x^{k}\right) - F^{*}\right) \leq \mathcal{O}\left(C^{k}\right)$$

for general smooth convex function f, we have

$$\mathbb{E}F\left(x^{k}\right) - F^{*} \leq \mathcal{O}\left(\frac{1}{k}\right)$$

Looser assumption to delay

Async-Parallel Iteration with Arbitrary Delays asynchronously

Paper

On the Convergence of Asynchronous Parallel Iteration with Arbitrary Delays

Formulation and algorithm

AsySCD & AsySPCD.

Main contribution

Analyze the convergence of the algorithm and allow for arbitrarily large delays following a certain distribution. Main assumption on the delay is the boundedness of certain expected quantities(e.g., expected delay, variance of delay).

Main assumption and convergence rate

Assumption

- the reading $\hat{x}^k$ is consistent and delayed by j_k, namely, $\hat{x}^k = x^{k-j_k}$, and the delay follows an identical distributions:

$$Prob\,(j_k = t) = q_t, t = 0, 1, 2, \cdots, \forall k$$

- Assume

$$T := \mathbb{E}\,[j_k] < \infty$$

Theorem

If the stepsize is taken as $0 < \eta < \frac{1/L_c}{1+2\kappa T/\sqrt{m}}$, then

$$\lim_{k\to\infty} \mathbb{E} \left\| \nabla f \left(x^k \right) \right\| = 0$$

any limit point of $\left\{x^k\right\}_{k\geq 1}$ is almost surely a critical point.

Main assumption and convergence rate

Assumption

There is a constant $\delta > 1$ such that

$$M_\delta := \mathbb{E}\left[\sigma^{j_k}\right] < \infty$$

Lemma

Under assumptions above, we have that for any $1 < \rho \leq \delta$, if the stepsize satisfies

$$0 < \eta \leq \frac{(\mu - 1)\sqrt{m}}{\mu L_r (1 + M_\rho)}$$

then for all k,

$$\mathbb{E}\left\|\nabla f\left(x^k\right)\right\|^2 \leq \rho \mathbb{E}\left\|\nabla f\left(x^{k+1}\right)\right\|^2$$

$$\mathbb{E}\left\|\nabla f\left(x^{k+1}\right)\right\|^2 \leq \rho \mathbb{E}\left\|\nabla f\left(x^k\right)\right\|^2$$

Convergence rate for the convex smooth case

- For a certain $1 < \mu < \sigma$, define

$$N_\mu := \mathbb{E}\left[j_k \mu^{j_k}\right]$$

Take the stepsize above and also

$$0 < \eta \leq \frac{2/L_c}{M_\mu + \frac{\kappa(2N_\mu M_\mu + T)}{\sqrt{m}}}$$

- if f is convex, then

$$\mathbb{E}\left[f\left(x^{k+1}\right) - f^*\right] \leq \mathcal{O}\left(\frac{1}{k}\right)$$

- if f is strongly convex with constant μ, then

$$\mathbb{E}\left[f\left(x^{k+1}\right) - f^*\right] \leq (1 - 2\mu D)\,\mathbb{E}\left[f\left(x^{k}\right) - f^*\right]$$

Convergence results for the nonsmooth case

Assumption

Assume assumptions above, then for any $1 < \mu < \sigma$, it holds that

$$\gamma_{\mu,1} := \sum_{t=1}^{\infty} q_t \frac{\mu^{t/2} - 1}{\mu^{1/2} - 1} < \infty$$

$$\gamma_{\mu,2} := \left(\sum_{t=1}^{\infty} q_t t \frac{\mu^{t} - 1}{\mu^{1/2} - 1} \right)^{1/2} < \infty$$

Lemma

Stepsize is taken such that

$$0 < \eta \leq \frac{\left(1 - \mu^{-1}\sqrt{m} - 4\right)}{2L_r\left(1 + \gamma_{\mu,1} + \gamma_{\mu,2}\right)}$$

then for all $k \geq 1$

$$\mathbb{E}\left\|x^{k-1} - \bar{x}^{k}\right\|^2 \leq \rho \mathbb{E}\left\|x^{k} - \bar{x}^{k+1}\right\|^2$$

Convergence results for the nonsmooth case

- Adopt stepsize above and also

$$\eta \leq \frac{1}{L_c + \frac{2L_f\gamma_{\mu,2}^2}{m} + \frac{2L_f\gamma_{\mu,2}}{m}}$$

- define $\Phi\left(x^k\right) = \mathbb{E}\left\|x^k - x^*\right\|^2 + 2\eta\mathbb{E}\left[F\left(x^k\right) - F^*\right]$
- if F is convex, then

$$\mathbb{E}\left[F\left(x^k\right) - F^*\right] \leq \frac{m\Phi\left(x^0\right)}{2\eta\left(m+k\right)}$$

- if F is strongly convex, then

$$\Phi\left(x^k\right) \leq \left(1 - \frac{\eta\mu}{m\left(1+\eta\mu\right)}\right)^k \Phi\left(x^0\right)$$

Poisson Distribution

- Treat the asynchronous reading and writing as a queueing system.
- Suppose the algorithm runs on a system with $p+1$ processors, which have the same speed of reading and writing, then the delay j_k follows the Poisson distribution with parameter p, i.e., for all k,
$$Prob\left(j_k = t\right) = \frac{p^t e^{-p}}{t!}, t = 0, 1, \cdots,$$
- if the processors have different computing power, j_k would follow Poisson distribution with a parameter being the speed ratio of the other p processors to the p_k-th one.
- $T = \mathbb{E}\left[j_k\right] = p$, $S = \mathbb{E}\left[j_k^2\right] = p\left(p+1\right)$
$M_\mu = \mathbb{E}\left[\mu^{j_k}\right] = e^{p(\mu-1)}$, $N_\mu = \mathbb{E}\left[j_k \mu^{j_k}\right] = \mu p e^{p(\mu-1)}$
$\gamma_{\mu,1} = \frac{e^{p\left(\sqrt{\mu}-1\right)}-1}{\sqrt{\mu}-1}$, $\gamma_{\mu,2} = \sqrt{\frac{\mu p e^{p(\mu-1)}-p}{1-\mu^{-1}}}$

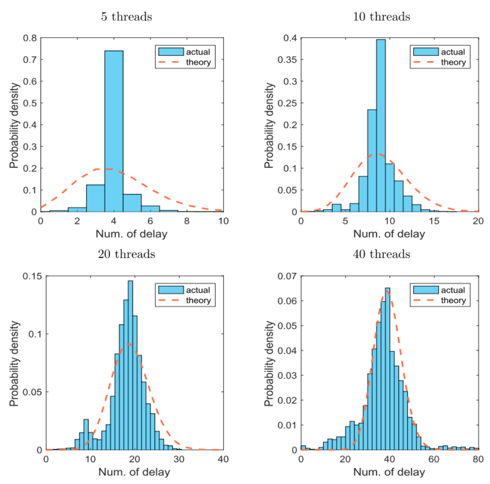


Figure 2: Delay distribution behaviors of the algorithm for solving LASSO. The tested problem has 20, 000 coordinates, and it was running with 5, 10, 20, and 40 threads.

AdaDelay

Introduction

Paper

Delay Adaptive Distribution Stochastic Optimization(AISTATS 16)

Problem

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

Algorithm

$$x_{t+1} := x_t - \alpha(t, \tau_t) \nabla f_{i_t}(\hat{x}_k)$$

Motivation

The server takes larger update steps when it obtains gradients from infrequent contributors, and smaller ones with gradients from frequent contributors

Adapt delay

Adagrad

- Motivation: It adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters.
- Iteration: $\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii}+\epsilon}} \cdot g_{t,i}$

  $G_t \in \mathbb{R}^{d \times d}$ here is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradient w.r.t. θ_i up to time step t

Adadelay

Stepsize:

$$\alpha\left(t, \tau_t\right) = \left(L + \eta\left(t, \tau_t\right)\right)^{-1}$$

$$\eta\left(t, \tau_t\right) = c\sqrt{t + \tau_t}$$

Conclusion

Conclusion

- It's very important to consider inconsistent read in CD.
- Bounded delay assumption can be loosed in many way.
- We can adapt delay to stepsize to speed up asynchronous parallel.