ARock: an asynchronous parallel coordinate update framework

Zhimin Peng[†], Yangyang Xu[‡], Ming Yan[†], Wotao Yin[†]

([†]UCLA Math, [‡]U.Waterloo DCO)

UCLA CAM Report 15-37

Int'l Symposium on Mathematical Programming — July 16, 2015

The fixed-point problem

- Operator $T : \mathcal{H} \to \mathcal{H}$, find $x \in \mathcal{H}$ such that

$$x = Tx$$

- it abstracts many problems:
    - convex optimization
    - statistical regression
    - optimal control
    - linear and nonlinear systems of equations
    - certain ordinary and partial differential equations

Krasnosel'skiĭ–Mann (KM) iteration

- **require**:
  1. T has a fixed point;
  2. nonexpansive operator T, that is

$$\|Tx - Ty\| \leq \|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

- **iteration**:

$$\begin{aligned} & x^{k+1} = (1 - \eta_k) x^k + \eta_k T x^k \\ \iff \quad & x^{k+1} = x^k - \eta_k \underbrace{(I - T)}_{S} x^k \end{aligned}$$

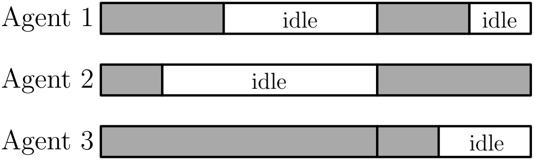
- **special cases:** gradient descent, proximal-point algorithm, prox-gradient, operator-splitting algorithms such as Douglas-Rachford and ADMM, ...

Sync-parallel coordinate update

- suppose $\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_m$
- agents i update $x_i \in \mathcal{H}_i$ in parallel, $i = 1, \ldots, m$:

$$\begin{array}{c} \text{agent } 1: \\ \text{agent } 2: \\ \vdots \\ \text{agent } m: \end{array} \qquad \begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \vdots \\ x_m^{k+1} \end{bmatrix} = \begin{bmatrix} x_1^{k} \\ x_2^{k} \\ \vdots \\ x_m^{k} \end{bmatrix} - \eta_k \begin{bmatrix} (Sx^k)_1 \\ (Sx^k)_2 \\ \vdots \\ (Sx^k)_m \end{bmatrix}$$

- **require**:
  1. x is a shared global variable
  2. each $(Sx)_i$ is much easier to compute than Sx
  3. synchronization after each iteration



Synchronous

(new iteration starts after the last agent finishes)

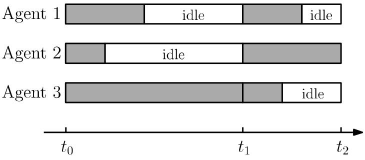
ARock: Async-parallel coordinate update

- suppose $\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_m$
- p agents, possibly $p \neq m$
- each agent **randomly picks** $i \in \{1, \ldots, m\}$ and updates x_i:

$$\begin{bmatrix} x_1^{k+1} \\ \vdots \\ x_i^{k+1} \\ \vdots \\ x_m^{k+1} \end{bmatrix} = \begin{bmatrix} x_1^{k} \\ \vdots \\ x_i^{k} \\ \vdots \\ x_m^{k} \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ \eta_k (S x^{k-d_k})_i \\ \vdots \\ 0 \end{bmatrix}$$

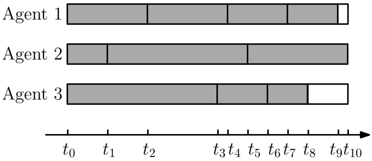
- $0 \leq d_k \leq \tau$
- **require**:$(Sx)_i$ is much easier to compute than Sx; also, proper η_k

Comparison: iteration is redefined



Synchronous

new iteration = all agents finish



Asynchronous

new iteration = any agent finishes

Brief history of async-parallel algorithms

- **1969** – a linear equation solver by Chazan and Miranker;
- **1978** – extended to the fixed-point problem by Baudet under the **absolute-contraction**[1] type of assumption.
- For 20–30 years, mainly solve linear, nonlinear and differential equations.
- **1989** – *Parallel and Distributed Computation: Numerical Methods* by Bertsekas and Tsitsiklis.

[1] An operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is absolute-contractive if $|T(x) - T(y)| \leq A|x - y|$, component-wise, where $|x|$ denotes the vector with components $|x_i|, \; i = 1, \ldots, n$, and $A \in \mathbb{R}^{n \times n}$ is a matrix with a spectral radius strictly less than 1.

Recent work

- **2013** – AsySCD for minimizing convex smooth functions by Liu et al.
- **2014** – AsySPCD for minimizing convex composite objective functions by Liu and Wright.
- **2015** – async-parallel randomized dual coordinate update for ridge regression problems by Hsieh et al.
- Other async-parallel / async-ADMM methods: Wei-Ozdaglar'13, Iutzeler et al'13, Zhang-Kwok'14, Hong'14.

ARock contributions

- An async-parallel framework for nonexpansive $T \quad \Rightarrow \quad$ applications:
  - (smooth and nonsmooth) function minimization, recovers AsySPCD
  - async-Jacobi for linear equations
  - distributed and decentralized optimization
  - operator splitting algorithms ...
- Convergence:
  - almost sure convergence of x^k to $x^* \in \operatorname{Fix} T$
  - linear convergence (when S is strongly monotone)

Convergence results

Recall: $p_{\min} = \min_i p_i > 0$, m is # coordinates, τ is the maximum delay.

Theorem (Weak convergence)

Assume that T is nonexpansive and has a fixed point. Let $(x^k)_{k\geq 0}$ be the sequence generated by ARock with the step sizes $\eta_k \in [\eta_{\min}, \frac{mp_{\min}}{2\tau\sqrt{p_{\min}}+1})$, $\forall k$. Then, with probability one, $(x^k)_{k\geq 0}$ weakly converges to a fixed point of T.

Theorem (Linear convergence rate)

If S is quasi-μ-strongly monotone if $\langle x - y, Sx - Sy\rangle \geq \mu\|x - y\|^2$ for any $x \in \mathcal{H}$ and $y \in \mathrm{zer}S := \{y \in \mathcal{H} : Sy = 0\}$, then with certain fixed step size

$$\mathbb{E}\|x^k - x^*\|^2 \leq c^k \cdot \|x^0 - x^*\|^2, \text{ with } c < 1.$$

Sketch of proof

- The key inequality:

$$\mathbb{E}\left(\|\mathbf{x}^{k+1}-\mathbf{x}^*\|_M^2 \,\middle|\, \mathcal{X}^k\right) \leq \|\mathbf{x}^k-\mathbf{x}^*\|_M^2 - c\,\|\bar{x}^{k+1}-x^k\|^2$$

  where

  - $c = c(\eta_k, m, \tau)$
  - $\mathbf{x}^k = (x^k, x^{k-1}, \ldots, x^{k-\tau}) \in \mathcal{H}^{\tau+1},\ k \geq 0$
  - $\mathbf{x}^* = (x^*, x^*, \ldots, x^*) \in \mathbf{X}^* \subseteq \mathcal{H}^{\tau+1}$
  - M is a positive definite matrix.

Sketch of proof

- apply the Robbins-Siegmund theorem;
- prove weakly convergence clustering points are fixed-points;
- apply a Theorem from [Combettes, Pesquet 2014].

Applications and numerical results

Minimizing composite functions

- **require**: convex smooth function g and convex (possibly nonsmooth) function f
- **proximal map**: $\mathbf{prox}_{\gamma f}(y) = \arg\min f(x) + \frac{1}{2\gamma}\|x-y\|^2$.

$$\underset{x}{\text{minimize}}\ f(x) + g(x) \iff x = \underbrace{\mathbf{prox}_{\gamma f} \circ (I - \gamma \nabla g)}_{T} x.$$

- ARock will be very fast given
  - easy-to-compute $\nabla_{x_i} g(x)$
  - either separable or easy-to-compute f (e.g., ℓ_1 and $\ell_{1,2}$)

Example: sparse logistic regression

- n features, N labeled samples
- each sample $a_i \in \mathbb{R}^n$ has its label $b_i \in \{1, -1\}$
- ℓ_1 regularized logistic regression:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \lambda \|x\|_1 + \frac{1}{N} \sum_{i=1}^{N} \log\Big(1 + \exp(-b_i \cdot a_i^T x)\Big), \tag{1}$$

- compare sync-parallel and ARock (async-parallel) on two datasets:

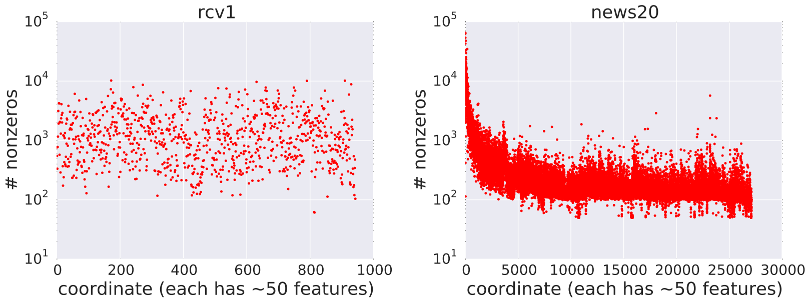
Name	N (#samples)	n (# features)	# nonzeros in $\{a_1, \ldots, a_N\}$
rcv1	20,242	47,236	1,498,952
news20	19,996	1,355,191	9,097,916

Speedup tests

- implemented in C++ and OpenMP.
- 32 cores shared memory machine.

#cores	rcv1				news20			
	Time (s)		Speedup		Time (s)		Speedup	
	async	sync	async	sync	async	sync	async	sync
1	122.0	122.0	1.0	1.0	591.1	591.3	1.0	1.0
2	63.4	104.1	1.9	1.2	304.2	590.1	1.9	1.0
4	32.7	83.7	3.7	1.5	150.4	557.0	3.9	1.1
8	16.8	63.4	7.3	1.9	78.3	525.1	7.5	1.1
16	9.1	45.4	13.5	2.7	41.6	493.2	14.2	1.2
32	4.9	30.3	**24.6**	**4.0**	22.6	455.2	**26.1**	**1.3**

Sparsity pattern and load imbalance



- each dot gives the # nonzeros in each coordinate (about 50 features)
- left: range of # nonzero: $10^2 - 10^4$
- right: range of # nonzero: $10^{1.8} - 10^5$
- larger ratio $\Rightarrow$ worse load balance

More applications

- Peaceman-Rachford splitting operator
- Douglas-Rachford splitting operator
- Davis-Yin's three operator splitting
- Parallel/distributed ADMM
- Decentralized ADMM

Async-parallel decentralized ADMM

- **a graph of connected agents**: $G = (V, E)$.
- decentralized consensus optimization problem:

$$\begin{aligned} &\underset{x_i \in \mathbb{R}^d, i \in V}{\text{minimize}} \; f(\mathbf{x}) := \textstyle\sum_{i \in V} f_i(x_i) \\ &\text{subject to } x_i = x_j, \; \forall (i,j) \in E \end{aligned}$$

- **ADMM reformulation**: constraints $x_i = y_{ij}, \; x_j = y_{ij}, \quad \forall (i,j) \in E$
- apply ARock
  - version 1: nodes asynchronously activate
  - version 2: edges (and nodes of each edge) asynchronously activate
  - both versions: each agent keeps f_i private and talks to its neighbors

notation:

- $N(i)$ all edges of agent i, $N(i) = L(i) \cup R(i)$
- $L(i)$ neighbors j of agent i, $j < i$
- $R(i)$ neighbors j of agent i, $j > i$

Algorithm 1: ARock for the decentralized consensus problem

Input : each agent i sets $x_i^0 \in \mathbb{R}^d$, dual variables $z_{e,i}^0$ for $e \in E(i)$, $K > 0$.

while $k < K$, *any activated agent* i **do**

  receive $\hat{z}_{li,l}^k$ from neighbors $l \in L(i)$ and $\hat{z}_{ir,r}^k$ from neighbors $r \in R(i)$;

  update local $\hat{x}_i^k$, $z_{li,i}^{k+1}$ and $z_{ir,i}^{k+1}$ according to **(??)–(??)**, respectively;

  send $z_{li,i}^{k+1}$ to neighbors $l \in L(i)$ and $z_{ir,i}^{k+1}$ to neighbors $r \in R(i)$.

$$\hat{x}_i^k \in \arg\min_{x_i} f_i(x_i) + \Big(\sum\nolimits_{l \in L(i)} \hat{z}_{li,l}^k + \sum\nolimits_{r \in R(i)} \hat{z}_{ir,r}^k \Big) x_i + \tfrac{\gamma}{2} |E(i)| \|x_i\|^2, \quad \text{(2a)}$$

$$z_{ir,i}^{k+1} = z_{ir,i}^k - \eta_k ((\hat{z}_{ir,i}^k + \hat{z}_{ir,r})/2 + \gamma \hat{x}_i^k), \quad \forall r \in R(i), \quad \text{(2b)}$$

$$z_{li,i}^{k+1} = z_{li,i}^k - \eta_k ((\hat{z}_{li,i}^k + \hat{z}_{li,l})/2 + \gamma \hat{x}_i^k), \quad \forall l \in L(i). \quad \text{(2c)}$$

Two ways to model $\hat{x}^k$

definitions: let $x^0, ..., x^k, ...$ be the states of x in the memory

1. $\hat{x}^k$ is called **consistent** if $\hat{x}^k = x^{k-d_k}$ for some $0 \le d_k \le \tau$.
2. $\hat{x}^k$ is called **inconsistent** if $\hat{x}^k \ne x^j$ for every $j \le k$.

ARock allows both consistent and inconsistent read.

Thank you!

Acknowledgements: NSF DMS

Reference: Zhimin Peng, Yangyang Xu, Ming Yan, Wotao Yin. UCLA CAM 15-37.

Website: `http://www.math.ucla.edu/~wotaoyin/arock`