Doubly Random Parallel Stochastic Methods for Large-Scale Learning

Aryan Mokhtari, Alec Koppel, and Alejandro Ribeiro
University of Pennsylvania, Philadelphia, PA

Asilomar Conference on Signals, Systems, and Computers
Pacific Grove, CA, Nov, 9, 2016
Learning \Rightarrow \text{params } \mathbf{x}^* \in \mathbb{R}^p \text{ that minimize expected risk } F(\mathbf{x})

\Rightarrow f : \mathbb{R}^p \rightarrow \mathbb{R} \Rightarrow \text{convex loss, quantifies merit of statistical model}

\Rightarrow \theta \text{ is random variable representing data}

\mathbf{x}^* := \arg\min_{\mathbf{x}} F(\mathbf{x}) := \arg\min_{\mathbf{x}} \mathbb{E}_\theta [f(\mathbf{x}, \theta)]
Learning ⇒ params $\mathbf{x}^* \in \mathbb{R}^p$ that minimize expected risk $F(\mathbf{x})$

- $f : \mathbb{R}^p \to \mathbb{R}$ ⇒ convex loss, quantifies merit of statistical model

- $\theta$ is random variable representing data

Suppose $N$ i.i.d. samples $\theta_n$ of stationary dist. of $\theta$

- $f_n(\mathbf{x}) := f(\mathbf{x}, \theta_n)$ loss associated with $n$-th sample

$$\mathbf{x}^* := \arg\min_{\mathbf{x}} F(\mathbf{x}) := \arg\min_{\mathbf{x}} \frac{1}{N} \sum_{n=1}^{N} f_n(\mathbf{x})$$

Example problems:

- support vector machines
- logistic regression
- matrix completion
Learning $\Rightarrow$ params $\mathbf{x}^* \in \mathbb{R}^p$ that minimize expected risk $F(\mathbf{x})$

- $f : \mathbb{R}^p \rightarrow \mathbb{R}$ $\Rightarrow$ convex loss, quantifies merit of statistical model
- $\theta$ is random variable representing data

Suppose $N$ i.i.d. samples $\theta_n$ of stationary dist. of $\theta$

- $f_n(\mathbf{x}) := f(\mathbf{x}, \theta_n)$ loss associated with $n$-th sample

$$
\mathbf{x}^* := \arg\min_{\mathbf{x}} F(\mathbf{x}) := \arg\min_{\mathbf{x}} \frac{1}{N} \sum_{n=1}^{N} f_n(\mathbf{x})
$$

Example problems:
- support vector machines
- logistic regression
- matrix completion

Focus: feature dimension $p$ and sample size $N$ are huge-scale

$\Rightarrow$ e.g., $p = \mathcal{O}(N)$
Background

- Optimization for large $N$: stochastic approximation
  - stochastic first-order (SGD, SAG, SVRG, etc.)
  - stochastic quasi-Newton (RES, SQN, oLBFGS)

- Optimization for large $p$: block coordinate methods
  - block coordinate descent
  - stochastic coordinate descent

- Optimization for large $p$ and $N$
  - asynchronous block SGD w/ sparsity (Hogwild!)
  - This work: operate on random subsets of features & samples
  - no block separability in gradient computations as in Hogwild!
Recall the problem

\[ x^* := \arg\min_x F(x) := \arg\min_x \mathbb{E}_\theta[f(x, \theta)] \]

\( N \) is very large \( \Rightarrow \) can’t afford gradient or Newton methods
\( \Rightarrow \) solution: stochastic methods

Classically solved with stochastic gradient method

\[ x^{t+1} = x^t - \gamma^t \nabla_x f(x^t, \theta^t) \]

\( \Rightarrow \) descend using stochastic gradient rather than true gradient
\( \Rightarrow \) breaks bottleneck in \( N \) \( \Rightarrow \) operate on one sample at a time

Nice analytical properties for convex and strongly convex cases
\( \Rightarrow \) Converges sublinearly in mean, converges to optimum a.s.
Suppose the feature dimension $p = O(N)$. For this case:

⇒ Computational complexity per iteration $O(p)$ ⇒ very large!
⇒ Stochastic gradient update is computationally demanding

Focus: break bottleneck in $p$ in stochastic approx. methods

We do this by partitioning vector $\mathbf{x}$ into $B$ blocks of size $p_b$

⇒ block stochastic approximation on random subsets of blocks
⇒ executed by a collection of $I$ parallel processors

Results in a doubly stochastic parallel method (RAPSA)

Propose Quasi-Newton extension

Establish convergence properties comparable to SGD
Parallelizing over Features

- Break regressor $\mathbf{x}$ into $B$ distinct blocks $\mathbf{x}_b$ of size $p_b << p$
- Associate w/ each block an i.i.d. sample of random variable $\theta$:

$$
\begin{bmatrix}
\mathbf{x}_1 \\
\vdots \\
\mathbf{x}_B
\end{bmatrix} \leftrightarrow
\begin{bmatrix}
\theta_1 \\
\vdots \\
\theta_B
\end{bmatrix}
$$
Parallelizing over Features

- Break regressor $\mathbf{x}$ into $B$ distinct blocks $\mathbf{x}_b$ of size $p_b \ll p$
- Associate w/ each block an i.i.d. sample of random variable $\theta$:

$$
\begin{bmatrix}
\mathbf{x}_1 \\
\vdots \\
\mathbf{x}_B
\end{bmatrix}
\longleftrightarrow
\begin{bmatrix}
\theta_1 \\
\vdots \\
\theta_B
\end{bmatrix}
$$

- Collection of $I \ll B$ processors work in parallel
  - Each processor randomly chooses a block $\mathbf{x}_b$
- Rather than parallel SGD, gradient update on only *some* blocks
  - Processor $P_i$ updates block $\mathbf{x}_b$ w/ stochastic subset of data.
- Advantages of both stochastic coordinate descent and SGD
RAPSA: Random Parallel Stochastic Algorithm

- Processor $P_i$ picks block $b_i^t \in [B]$ at random, sample subset $\theta_i^t$
- Executes block SGD

\[
x_{b_i}^{t+1} = x_{b_i}^t - \gamma^t \nabla_{x_{b_i}} f(x^t, \theta_i^t), \quad b = b_i^t.
\]

- Block selection $\Rightarrow$ no processors operate on the same block
- Processors use shared memory on common time index.
Technical Conditions

- Instantaneous objective functions $f(x, \theta)$ ⇒ differentiable
- Average function $F(x) = \mathbb{E}_\theta[(x, \theta)]$ ⇒ $m$-strongly convex
- Average objective gradients $\nabla F(x)$ ⇒ $M$-Lipschitz continuous,
  ⇒ For all $x, \hat{x} \in \mathbb{R}^p$, it holds
  \[ \|\nabla F(x) - \nabla F(\hat{x})\| \leq M \|x - \hat{x}\|. \]
- Stochastic gradient has finite variance
  ⇒ for a constant $K$, all $x$, we have
  \[ \mathbb{E}_\theta[\|\nabla f(x^t, \theta^t)\|^2 | x^t] \leq K. \]
- Standard conditions in stochastic approximation literature
Convergence of RAPSA

Theorem
(i) The RAPSA sequence \( \{x^t\} \), with diminishing step-size rules \( \gamma^t = O(1/t) \) converges a.s. to optimal \( x^* \),

\[
\lim_{t \to \infty} \|x^t - x^*\|^2 = 0 \quad \text{a.s.}
\]

(ii) If step-size is such that \( \gamma^t := \gamma^0 T^0 / (t + T^0) \) and \( 2mr \gamma^0 T^0 > 1 \), then the error sequence \( \mathbb{E}[F(x^t) - F(x^*)] \) converges to null as \( O(1/t) \),

\[
\mathbb{E}[F(x^t) - F(x^*)] \leq \frac{C}{t + T^0},
\]

⇒ Constant \( C \) is defined as

\[
C = \max \left\{ \frac{rMK(\gamma^0 T^0)^2}{4mr \gamma^0 T^0 - 2}, T^0(F(x^0) - F(x^*)) \right\}.
\]
Convergence of RAPSA

- Almost sure convergence to optimum using diminishing step-size
- A.s. convergence to nbhd. of optimum w/ constant step-size
- Linear convergence on average to optimal objective
  \[ \Rightarrow \text{provided step-size is chosen as sufficiently small constant} \]
First-order stochastic approximation methods $\Rightarrow$ converge slowly

In stochastic setting, Newton’s method impractical
$\Rightarrow$ requires inverting Hessian $H = \nabla^2 F$, an $p \times p$ dim. matrix
$\Rightarrow$ Quasi-Newton methods approximate this Hessian inverse

We develop an online block-coordinate Quasi-Newton method
$\Rightarrow$ $\mathcal{I}$ processors execute stochastic approx. updates in parallel
Extension: Accelerated RAPSA

- First-order stochastic approximation methods ⇒ converge slowly
- In stochastic setting, Newton’s method impractical
  ⇒ requires inverting Hessian $H = \nabla^2 F$, an $p \times p$ dim. matrix
  ⇒ Quasi-Newton methods approximate this Hessian inverse
- We develop an online block-coordinate Quasi-Newton method
  ⇒ $I$ processors execute stochastic approx. updates in parallel
- Consider RAPSA update at processor $i \in \{1, \ldots, I\}$
  ⇒ $I$ selects block index $b^t_i \in \{1, \ldots, B\}$ uniformly at random
    \[ x^{t+1}_b = x^t_b - \gamma^t \nabla_{x_b} f(x^t, \Theta^t_i), \quad b = b^t_i. \]
- Modify stochastic descent step by “pre-conditioning” matrix $\hat{B}^t_b$
  ⇒ $\hat{B}^t_b \approx [\nabla^2_{x_b} F(x^t_b)]^{-1}$ in a certain sense
First-order stochastic approximation methods ⇒ converge slowly

In stochastic setting, Newton’s method impractical
⇒ requires inverting Hessian $\mathbf{H} = \nabla^2 F$, an $p \times p$ dim. matrix
⇒ Quasi-Newton methods approximate this Hessian inverse

We develop an online block-coordinate Quasi-Newton method
⇒ $\mathcal{I}$ processors execute stochastic approx. updates in parallel

Accelerated RAPSA (ARAPSA): processor $i \in \{1, \ldots, I\}$
⇒ selects block index $b_i^t \in \{1, \ldots, B\}$ uniformly at random

$$x_{b}^{t+1} = x_{b}^{t} - \gamma^{t} \hat{\mathbf{B}}_{b}^{t} \nabla_{x_b} f(x^{t}, \Theta_{i}^{t}) , \quad b = b_{i}^{t}.$$ 

$\hat{\mathbf{B}}_{b}^{t} \approx [\nabla^2_{x_b} F(x_{b}^{t})]^{-1}$ is block Hessian inverse approximation
Hessian Approximation

- $\hat{H}_b^t$ is block Hessian approximation, $\hat{B}_b^t = [\hat{H}_b^t]^{-1}$
- Specify this matrix by considering gradient and var. variations

$$v_b^t = x_b^{t+1} - x_b^t, \quad \hat{r}_b^t = \nabla_{x_b} f(x_b^{t+1}, \Theta_b^t) - \nabla_{x_b} f(x_b^t, \Theta_b^t).$$

- True Hessian $H_b^t$ associated w/ block var. $x_b$
  $\Rightarrow$ has inverse which satisfies secant condition $H_b^t^{-1}v_b^t = \hat{r}_b^t$
  $\Rightarrow$ “shouldn’t change much” $\Rightarrow$ measured via differential entropy

$$\hat{H}_b^{t+1} = \arg\min \text{tr}(\hat{H}_b^t)^{-1}Z) - \log \det(\hat{H}_b^t)^{-1}Z) - n$$
  s. t. $Zv_b^t = \hat{r}_b^t, \quad Z \succeq 0$

- Secant condition interpretation
  $\Rightarrow$ stoch. grad. of quad. approx. of objective is similar over time
Block Online BFGS

- Block variant of Online BFGS $\Rightarrow$ approximate Hessian inverse
- Derived as closed-form solution of opt. prob. on previous slide

\[
\hat{H}_{b}^{t+1} = \hat{H}_{b}^{t} + \frac{\hat{r}_{b}(\hat{r}_{b}^{T})}{(v_{b}^{T})^{T}\hat{r}_{b}} - \frac{\hat{B}_{b}^{t}v_{b}(v_{b}^{T})^{T}\hat{H}_{b}^{t}}{(v_{b}^{T})^{T}\hat{H}_{b}^{t}v_{b}}
\]

$\Rightarrow$ apply Sherman-Morrison matrix inversion Lemma to the result

\[
[\hat{H}_{b}^{t+1}]^{-1} = \hat{B}_{b}^{t+1} = [Z_{b}^{t}]^{T}\hat{B}_{b}^{t}Z_{b}^{t} + \rho_{b}^{t}v_{b}(v_{b}^{T})^{T}
\]

with scalar $\rho_{b}^{t}$ and matrix $Z_{b}^{t}$ defined as

\[
\rho_{b}^{t} = \frac{1}{(v_{b}^{T})^{T}\hat{r}_{b}}, \quad Z_{b}^{t} = I_{p_{b}} - \rho_{b}^{t}r_{b}(v_{b}^{T})^{T}
\]

- $\hat{B}_{b}^{t+1}$ depends on $\hat{B}_{b}^{u}$ for $u < t + 1$

$\Rightarrow$ at time $t + 1$, must recurse over all $u < t + 1$ to compute $\hat{B}_{b}^{t+1}$

$\Rightarrow$ Let’s truncate update at $t + 1$ to only past $\tau$ iterations
BoL-BFGS: Block Online Limited Memory BFGS

- **\( \tau \)** ⇒ **memory** for block online Limited Memory BFGS (oL-BFGS)
- ⇒ use past \( \tau \) pairs of curvature information \( \{ v_b^u, r_b^u \}_{u=t-\tau}^{t-1} \)
- ⇒ Approximate matrix \( \hat{B}_b^t \) is computed by initializing as

\[
\hat{B}_{b}^{t,0} := \eta_b^t I, \quad \eta_b^t := \frac{(v_b^{t-1})^T \hat{r}_b^{t-1}}{\| \hat{r}_b^{t-1} \|_2},
\]

- Approx. Hessian inverse ⇒ \( \tau \) recursive applications of update

\[
\hat{B}_{b}^{t,u+1} = (\hat{Z}_{b}^{t-\tau+u})^T \hat{B}_b^t (\hat{Z}_b^{t-\tau+u}) + \hat{\rho}_b^{t-\tau+u} (v_b^{t-\tau+u}) (v_b^{t-\tau+u})^T,
\]

- Matrices \( \hat{Z}_{b}^{t-\tau+u} \), constant \( \hat{\rho}_b^{t-\tau+u} \) for \( u = 0, \ldots, \tau - 1 \) defined as

\[
\hat{\rho}_b^{t-\tau+u} = \frac{1}{(v_b^{t-\tau+u})^T \hat{r}_b^{t-\tau+u}} \quad \text{and} \quad \hat{Z}_b^{t-\tau+u} = I - \hat{\rho}_b^{t-\tau+u} \hat{r}_b^{t-\tau+u} (v_b^{t-\tau+u})^T.
\]
Logistic Regression Example

- \( \mathbf{z} \in \mathbb{R}^p \) ⇒ feature vector encoding image pixel intensities
  ⇒ label \( y \in \{-1, 1\} \) ⇒ whether image contains digit 0 or 8
- Learning a hand-written digit detector ⇒ logistic regression
  ⇒ \( \mathbf{x} \in \mathbb{R}^p \) ⇒ relate samples \( \mathbf{z}_n \in \mathbb{R}^p \) to labels \( y_n \in \{-1, 1\} \)
- ERM problem associated with training set \( \mathcal{T} = \{ (\mathbf{z}_n, y_n) \}_{n=1}^N \)
  ⇒ Find \( \mathbf{x}^* \) as \( \ell_2 \) regularized maximum likelihood estimate

\[
\mathbf{x}^* := \arg\min_{\mathbf{x} \in \mathbb{R}^p} \frac{\lambda}{2} \| \mathbf{x} \|^2 + \frac{1}{N} \sum_{n=1}^N \log(1 + \exp(-y_n \mathbf{x}^T \mathbf{z}_n)) ,
\]

⇒ Logistic transformation of odds ratio for label being \(-1\) or \(1\)
- We use an \( N = 1.76 \times 10^4 \) subset of MNIST with labels 0 and 8
  ⇒ Feature vectors \( \mathbf{z}_n \in \mathbb{R}^p \) are \( p = 28^2 = 784 \) pixel images
Logistic Regression Example

- RAPSA on binary subset of MNIST
  - hybrid step-size $\gamma^t = \min(10^{-2.5}, 10^{-2.5} \tilde{T}_0/t)$, $\tilde{T}_0 = 525$
  - no mini-batching $L = 1$
  - block size $p_b = p/4$

- Define $\tilde{p}_t = prtL$
  - no. of features processed per iteration

- Performance w.r.t. prop. of $x$ updated
  - faster when full $x$ is used with iteration $t$
  - faster with fewer entries of $x$ with $\tilde{p}_t$
Consider a test set of size $\tilde{N} = 5.88 \times 10^3$

Classification accuracy $\approx 95\%$

$\Rightarrow$ across different values of $B$

$\Rightarrow$ using fewest entries of $x$ is best

Now we fix $B = 64$

$\Rightarrow$ 1/4 of $x$ is updated per iteration

mini-batch size $L = 10$, step-size $\epsilon = 10^{-1}$

“Accelerated” RAPSA $\approx 3 \times$ RAPSA rate

$\Rightarrow$ ARAPSA $\Rightarrow$ block-wise oL-BFGS
Conclusions

- Classic stochastic approximation \( \Rightarrow \) can’t handle \( p = \mathcal{O}(N) \)
- RAPSA breaks bottleneck in \( p \)
  - Operates on random subsets of samples and features
- Can be implemented on a parallel computing architecture
  - Requires shared memory
- Convergence of RAPSA
  - Under standard technical conditions
- Benefits of both stochastic coordinate descent and SGD
  - Demonstrated utility on a standard parameter estimation task
  - Quasi-Newton extension \( \Rightarrow \) empirically superior convergence
A. Mokhtari, A. Koppel, and A. Ribeiro, “Doubly Random Parallel Stochastic Methods for Large Scale Learning,” American Control Conference, July. 2016.

A. Mokhtari, A. Koppel, and A. Ribeiro, “A Class of Parallel Doubly Stochastic Algorithms for Large-Scale Learning,” Journal of Machine Learning Research (Submitted), June. 2016. [Preprint on ArXiv]

http://seas.upenn.edu/~akoppel/