Parallel Stochastic Successive Convex Approximation Method for Large-Scale Dictionary Learning

Alec Koppel[†], Aryan Mokhtari[§], and Alejandro Ribeiro[*] Contact E-mail: alec.e.koppel.civ@mail.mil, aryanm@mit.edu, aribeiro@seas.upenn.edu

[†] CISD, U.S. Army Research Laboratory; [§] EECS, Massachusetts Institute of Technology; [*] Dept. of ESE, University of Pennsylvania

Large-Scale Parameter Estimation

- Supervised learning $\Rightarrow$ vector $\mathbf{x}^* \in \mathcal{X} \subset \mathbb{R}^p$ minimizing expected risk $F(\mathbf{x})$
- $f_n : \mathbb{R}^p \to \mathbb{R}$ $\Rightarrow$ non-convex loss associated to nth sample point.
- Examples:
  - $\Rightarrow$ Regression with non-linear transformation of feature space
  - $\Rightarrow$ Objective defined by training a neural network (deep learning)
- $G(\mathbf{x})$ convex non-differentiable, block separable,e.g.,$\|\mathbf{x}\|_1$ sparsity penalty

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}} F(\mathbf{x}) + G(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}} \frac{1}{N}\sum_{n=1}^{N} f_n(\mathbf{x}) + G(\mathbf{x})$$

- Example problems:
  - $\Rightarrow$ support vector machines,
  - $\Rightarrow$ logistic regression,
  - $\Rightarrow$ deep learning
- Both the feature dimension p and sample size N are huge-scale

Feature and Sample Parallel Learning

- Sample size $N \to \infty$ $\Rightarrow$ stochastic gradient
  - $\Rightarrow$ Workhorse for convex setting where $p << N$, can't handle $p = O(N)$
- **Key Novelty**: operate on subsets of both features and samples
- Main idea: partition classifier $\mathbf{x}$ into B distinct blocks of size $p_b << p$
  - $\Rightarrow$ Associate to block $\mathbf{x}_b \in \mathcal{X}_b$ distinct sub-vector of features $\theta \in \mathbb{R}^p$

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

- Rather than parallel stochastic gradient, SGD on only *some* blocks
- $I \ll B$ processors work in parallel by randomly choosing blocks
  - $\Rightarrow$ Processor P_i updates block $\mathbf{x}_b$ using stochastic subset of data.
- Merge advantages of stochastic coord. descent and stochastic approx.
- Prior work implemented this idea for convex stochastic programming
  - $\Rightarrow$ This work: **extension to non-convex composite** problems

Successive Convex Approximation

- Broad idea: replace non-convex part of objective with convex approximation
  - $\Rightarrow$ agrees with non-convex function at the gradient
- To develop doubly stoch. successive approx., many definitions required
- Define block-wise stochastic gradient

$$\nabla_{\mathbf{x}_b} f(\mathbf{x}, \Theta_b^t) = \frac{1}{L}\sum_{\theta \in \Theta_b^t} \nabla_{\mathbf{x}_b} f(\mathbf{x}, \theta).$$

  - $\Rightarrow$ Stochastic approx. of grad. of average cost w.r.t. block variables $\mathbf{x}_b$
  - $\Rightarrow$ Evaluated at a distinct data mini-batch Θ_b^t of size L
- Define block-wise instantaneous convex surrogate function $\tilde{f}_i(\mathbf{x}_b; \mathbf{x}, \theta)$
  - $\Rightarrow$ differentiable and convex with respect to $\mathbf{x}_i$
  - $\Rightarrow$ $\nabla_{\mathbf{x}_b}\tilde{f}_b(\mathbf{x}_b; \mathbf{x}, \theta)$ is equal to the gradient $\nabla_{\mathbf{x}_b} f(\mathbf{x}, \theta)$
  - $\Rightarrow$ $\nabla_{\mathbf{x}_b}\tilde{f}_i(\mathbf{x}_b; \mathbf{x}, \theta)$ is Γ-Lipschitz continuous
- Example surrogate: linearization of instantaneous cost

$$\tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \theta^t) = f(\mathbf{x}^t, \theta^t) + \nabla_{\mathbf{x}_b} f(\mathbf{x}^t, \theta^t)^T(\mathbf{x}_b - \mathbf{x}_i^t) + \frac{\tau_i}{2}\|\mathbf{x}_b - \mathbf{x}_b^t\|^2,$$

- Similarly, define the mini-batch sample surrogate as

$$\tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \Theta_i^t) = \frac{1}{L}\sum_{\theta \in \Theta_i^t} \tilde{f}_i(\mathbf{x}_b; \mathbf{x}^t, \theta).$$

- Define mini-batch surrogate function gradient w.r.t. data subset Θ_b^t:

$$\nabla \tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \Theta_b^t) = \frac{1}{L}\sum_{\theta \in \Theta_b^t} \nabla \tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \theta).$$

DSSC: Doubly Stochastic Successive Convex Approximation

- Processor P_i chooses $b \in \{1, \ldots, B\}$ uniformly random, data subset Θ_b^t
  - $\Rightarrow$ Executes "proximal" update using surrogate function $\tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \Theta_b^t)$

$$\hat{\mathbf{x}}_b^{t+1} = \operatorname*{argmin}_{\mathbf{x}_b \in \mathcal{X}_b}\Big\{\rho^t \tilde{f}_b(\mathbf{x}_b; \mathbf{x}^t, \Theta_b^t) + (1-\rho^t)(\mathbf{d}_b^{t-1})^T(\mathbf{x}_b - \mathbf{x}_b^t) + g_i(\mathbf{x}_b) + \frac{\tau_b}{2}\|\mathbf{x}_b - \mathbf{x}_b^t\|^2\Big\},$$

  - $\Rightarrow$ τ_b behaves similarly to inverse of step-size of standard SGD method
  - $\Rightarrow$ $\mathbf{d}_b^t$ $\Rightarrow$ time-average stochastic surrogate gradients w.r.t. block b:

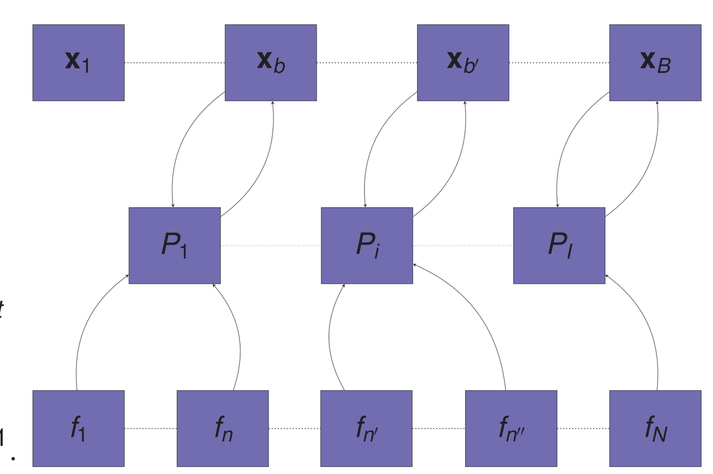
$$\mathbf{d}_b^t = (1-\rho^t)\mathbf{d}_b^{t-1} + \rho^t \nabla_{\mathbf{x}_b}\tilde{f}_b(\mathbf{x}_b^t; \mathbf{x}^t, \Theta_b^t),$$

  - $\Rightarrow$ ρ^t is gradient averaging momentum parameter
- Next block-variable $\Rightarrow$ convex combo. of last $\mathbf{x}_b^t$ and "proximal" $\hat{\mathbf{x}}_b^{t+1}$

$$\mathbf{x}_b^{t+1} = (1-\gamma^{t+1})\mathbf{x}_b^t + \gamma^{t+1}\hat{\mathbf{x}}_b^{t+1}.$$

  - $\Rightarrow$ γ^{t+1} is additional momentum/step-size parameter
- Update is done for each processor P_i **in parallel**

Parallel Architecture Implementation



- Iteration t, processor i picks
  - $\Rightarrow$ Block $b_i^t \in [B]$ at random,
  - $\Rightarrow$ Sample subset Θ_b^t
- Computes proximal step $\hat{\mathbf{x}}_b^{t+1}$
  - $\Rightarrow$ with surrogate $\tilde{f}(\mathbf{x}_b^t, \Theta_b^t)$
  - $\Rightarrow$ recursively avg. grad. $\mathbf{d}_b^t$
- Stabilized proximal update
  $\mathbf{x}_b^{t+1} = (1-\gamma^{t+1})\mathbf{x}_b^t + \gamma^{t+1}\hat{\mathbf{x}}_b^{t+1}$.
- Independent of processors $j \neq i$

- Allows for processors to operate on distinct data subset at same time
  - $\Rightarrow$ theoretical speedup comparable to number of processors I
  - $\Rightarrow$ allow for passing through training set I times faster
  - $\Rightarrow$ Amenable to implementation on GPU or HPC system

Technical Conditions

- The block-wise feasible sets $\mathcal{X}_i$ are convex and compact.
  - $\Rightarrow$ Common condition to guarantee optimization sequence is bounded

Block stochastic gradients have bounded stochastic approx. error

$$\mathbb{E}\|\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \theta^t) - \nabla_{\mathbf{x}_i} F(\mathbf{x}^t)\|^2 \mid \mathcal{F}^t < \infty.$$

  - $\Rightarrow$ Standard stochastic approximation condition
  - $\Rightarrow$ Typically bounded by constant plus a term proportional to $\|\mathbf{x}^t\|^2$
- The sequences γ^t and ρ^t are chosen such that
  - $\Rightarrow$ $\lim_{t\to\infty}\gamma^t = 0$, $\sum_{t=0}^{\infty}\gamma^t = \infty$, $\sum_{t=0}^{\infty}(\gamma^t)^2 < \infty$,
  - $\Rightarrow$ $\lim_{t\to\infty}\rho^t = 0$, $\sum_{t=0}^{\infty}\rho^t = \infty$, $\sum_{t=0}^{\infty}(\rho^t)^2 < \infty$,
  - $\Rightarrow$ $\lim_{t\to\infty}\gamma^t/\rho^t = 0$.
- Attenuating step-size rules for both
  - $\Rightarrow$ momentum parameter γ^t
  - $\Rightarrow$ gradient averaging parameter ρ^t
  - $\Rightarrow$ $\rho^t \to 0$ means in limit $\mathbf{d}_b^t$ is true block gradient

Convergence Result

- **Lemma:** *The DSSC sequence* $\{\mathbf{x}^t\}$ *with attenuating step-size parameters* $\rho^t, \gamma^t > 0$ *satisfies the stochastic descent relation*

$$\mathbb{E}[V(\mathbf{x}^{t+1}) \mid \mathcal{F}^t] \leq V(\mathbf{x}^t) - r\gamma^{t+1}\left[\frac{\tau}{2} - \frac{\Gamma\gamma^{t+1}}{2} - \rho^t\Gamma\right]\|\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t\|^2 + r\gamma^{t+1}\left(\nabla F(\mathbf{x}^t) - \mathbf{d}^t\right)^T(\hat{\mathbf{x}}^{t+1} - \mathbf{x}^t),$$

  - $\Rightarrow$ $\mathcal{F}^t$ $\Rightarrow$ filtration measuring the random variables up to step t
  - $\Rightarrow$ Expectation is taken with respect to randomized block selection $\mathcal{S}^t$.
  - $\Rightarrow$ Descent $\Rightarrow$ key to establishing convergence of stochastic methods

Theorem: *The DSSC sequence* $\{\mathbf{x}^t\}$ *converges to along a subsequence to stationary point of the non-convex regularized expected risk.*

- Almost sure convergence to optimum using diminishing step-size
- One of the first w.p.1 results for non-convex problems
- Stationarity is best one may hope for in generic non-convex problem

Simple Non-Convex Example: Perturbed Linear Estimation

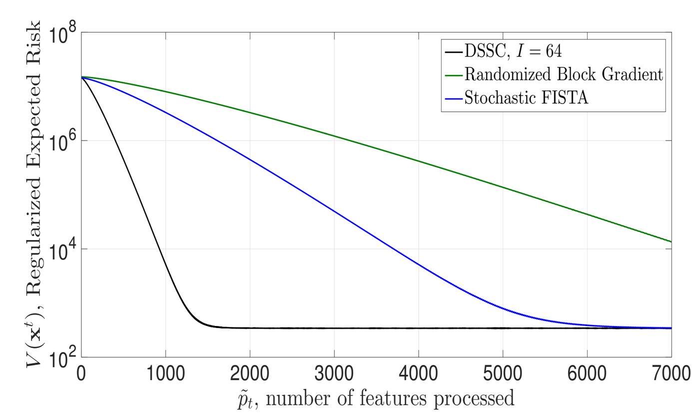
- Consider a *perturbed* linear regression problem
  - $\Rightarrow$ defined by an *indefinite* obs. matrix with an ℓ_1 regularization
- Observations $\mathbf{z}_n \in \mathbb{R}^q$ are collected
  - $\Rightarrow$ Noisy linear transformations $\mathbf{z}_n = \mathbf{H}_n\mathbf{x} + \mathbf{w}_n$ of unknown signal $\mathbf{x} \in \mathbb{R}^p$
  - $\Rightarrow$ $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 I_q)$ is a Gaussian random variable.
- For N samples, locally optimal $\mathbf{x}^*$ $\Rightarrow$ λ-regularized least squares estimate

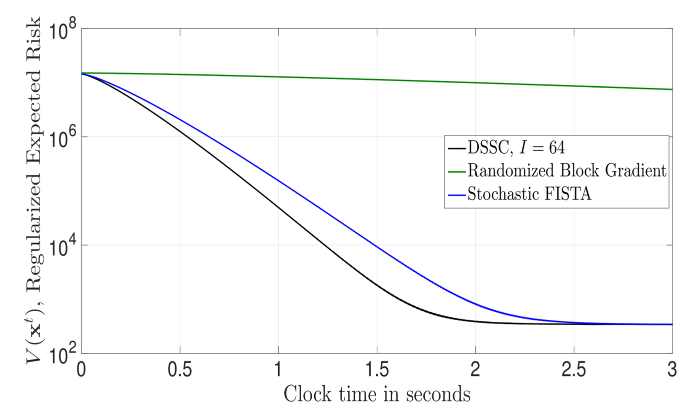
$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{H}_n\mathbf{x} - \mathbf{z}_n\|^2 - c\|\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1.$$

  - $\Rightarrow$ ℓ_1 regularization is introduced to sparsify the regressor
  - $\Rightarrow$ admits a block decomposition.
  - $\Rightarrow$ Non-convex due to term $-c\|\mathbf{x}\|^2$, provided $c > \lambda_{\max}(\mathbf{H}^T\mathbf{H})$.
- $F(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{H}_n\mathbf{x} - \mathbf{z}_n\|^2 - c\|\mathbf{x}\|^2$ and $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$.

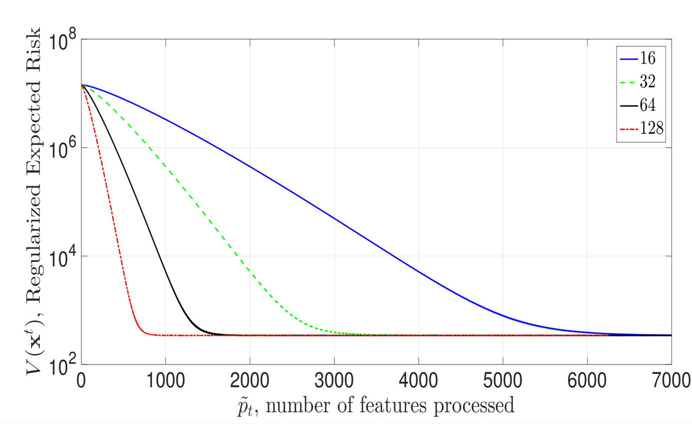
Results for Linear Estimation

- Fix $q = 1$, $p = 1024$, $N = 10^4$.
- Obs. matrices $\mathbf{H}_n \in \mathbb{R}^{q \times p}$
  - $\Rightarrow$ p-dimensional Gaussians
- True signal $\mathbf{x} \in \pm\{1, \ldots, p\}/p$
- Noise variance $\sigma^2 = 10^{-2}$.
- $I = 16$ Processors, 64 blocks
  - $\Rightarrow$ update $1/4$ of $\mathbf{x}$ per step
- Step-size $\rho^t = 5t^{-1/5000}$
  - $\Rightarrow$ momentum $\gamma^t = 2t^{-1/1000}$
- Surrogate function $\Rightarrow$ linearization
- Algorithm Comparison:
  - $\Rightarrow$ FISTA $\Rightarrow$ update all of $\mathbf{x}$
  - $\Rightarrow$ Random coord. descent
  - $\Rightarrow$ with deterministic gradients
- DSSC outperforms competitors
  - $\Rightarrow$ in terms of number of features processed
  - $\Rightarrow$ convergence w.r.t. clock time





- DSSC: vary number of processors I
  - $\Rightarrow$ $I \in \{16, 32, 64, 128\}$
- Larger I $\Rightarrow$ fewer features
  - $\Rightarrow$ faster w.r.t. no. of features
- Speedup $\approx$ problem conditioning



Dictionary Learning

- Collection of signals $\{\mathbf{z}_n\} \subset \mathbb{R}^m$ for $n = 1, \ldots, N$
  - $\Rightarrow$ Represent signals $\mathbf{z}_n$ as combos. of k basis elements $\{\mathbf{d}_l\}_{l=1}^k$
  - $\Rightarrow$ basis elements unknown $\Rightarrow$ learn from data
  - $\Rightarrow$ Group basis elements into dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$
- Denote the coding (coefficients) of $\mathbf{z}_t$ w.r.t. $\mathbf{D}$ as $\alpha_t \in \mathbb{R}^k$.
- Dictionary learning problem $\Rightarrow$ find coding α dictionary $\mathbf{D}$
  - $\Rightarrow$ $\mathbf{z}_t$ is close to its dictionary representation $\mathbf{D}\alpha$ for all t.
  - $\Rightarrow$ Introduce a loss function $f(\alpha_t, \mathbf{D}; \mathbf{z}) = \|\alpha\mathbf{D} - \mathbf{z}\|_2^2$
- When no. basis elements is greater than the feature dimension,
  - $\Rightarrow$ want signal encoding to be sparse $\Rightarrow$ incentivized via ℓ_1 regularizer

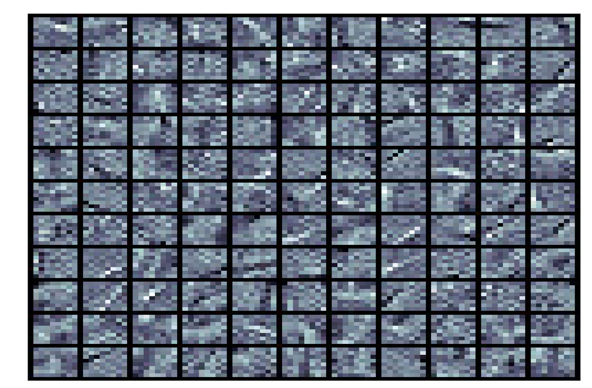
$$(\mathbf{D}^*, \alpha^*) := \operatorname*{argmin}_{\mathbf{D}\in\mathcal{D}, \alpha\in\mathcal{A}} \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{D}\alpha - \mathbf{z}_n\|_2^2 + \zeta_1\|\alpha_t\|_1.$$

- $\mathcal{D} := \{\mathbf{D} \in \mathbb{R}^{m \times k} : \|\mathbf{d}_l\| \leq 1,$ for all $l\}$ $\Rightarrow$ $m \times k$ matrices w/ unit col. norm
  - $\Rightarrow$ Eliminate the scale ambiguity of the bilinear term $\mathbf{D}\alpha$
  - $\Rightarrow$ $\mathcal{A} \subset \mathbb{R}^k$ $\Rightarrow$ compact subset from which sparse codes are learned
  - $\Rightarrow$ Bilinear term makes this formulation non-convex.

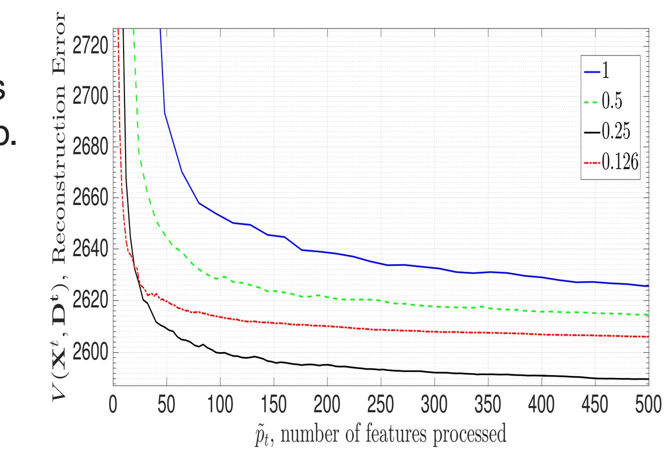
Results on Robot Image Feed

- We want to learn dictionary online
  - $\Rightarrow$ via robot image stream
  - $\Rightarrow$ for e.g., occlusion removal
- Images collected on Husky robot
  - $\Rightarrow$ prototypical ground vehicle

- Learned dict. from robot images
  - $\Rightarrow$ Consisting of image patches
  - $\Rightarrow$ via concrete, grass pathes
- Dict. atoms $\Rightarrow$ terrain features



- Vary no. of coord. per iteration
- Faster learning with less features
- Validates trend in linear reg. prob.
  - $\Rightarrow$ Problem w/ empirical data



Conclusions

- Classical stochastic approximation methods $\Rightarrow$ can't handle $p = O(N)$
- DSSC breaks bottleneck in p, first time for non-convex problems
  - $\Rightarrow$ Operates on random subsets of samples and features
- Can be implemented on a parallel computing architecture
- No coordination among distinct computing nodes required
- Established convergence of DSSC
  - $\Rightarrow$ Via use of recursive averaging of convex surrogate gradients
  - $\Rightarrow$ Under standard technical conditions
- Benefits of both stochastic coordinate descent and stochastic gradient
- Promising results both on numerical experiments
  - $\Rightarrow$ Corroborated on unsupervised learning problem: dictionary learning