ICML, August 6-11, 2017, Sydney

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Max Planck Institute for Intelligent Systems, Tübingen Campus

Approximate Steepest Coordinate Descent

sebastian.stich@epfl.ch, anant.raj@tuebingen.mpg.de, martin.jaggi@epfl.ch

The Setting

$$\min_{\mathbf{x}\in\mathbb{R}^n} [f(\mathbf{x}) := F(A\mathbf{x})]$$

Assumptions:

- $f\colon \mathbb{R}^n \to \mathbb{R}$ convex; $\forall \mathbf{x} \in \mathbb{R}^n, \gamma \in \mathbb{R}$, $i = 1\colon n$, $|\nabla_i f(\mathbf{x} + \gamma \mathbf{e}_i) - \nabla_i f(\mathbf{x})| \leq L\,|\gamma|$
- $A \in \mathbb{R}^{d\times n}$
- $d = \Omega(n)$

Coordinate Descent:

- $\mathbf{x}_+ = \mathbf{x} - \frac{1}{L}\nabla_i f(\mathbf{x})$
- $f(\mathbf{x}) - f\left(\mathbf{x} - \frac{1}{L}\nabla_i f(\mathbf{x})\right) \geq \underbrace{\frac{1}{2L}(\nabla_i f(\mathbf{x}))^2}_{:=\tau^{[i]}(\mathbf{x})}$

One step progress:

- $\tau(\mathbf{x}) := \mathbb{E}\left[|\tau^{[i]}(\mathbf{x})|\right]$

Complexity:

- $T(\nabla f(\mathbf{x})) = O(dn) = \Omega(n^2)$
- $T(\nabla_i f(\mathbf{x})) = O(d) = \Omega(n)$

Highlights

Steepest Coordinate Descent (SCD)

$$i_{\text{SCD}} = \operatorname*{argmax}_{i\in[n]} |\nabla_i f(\mathbf{x})|$$

- **Classic analysis:**

$$\tfrac{1}{n}\tau_{\text{SCD}}(\mathbf{x}) \leq \tau_{\text{UCD}}(\mathbf{x}) \leq \tau_{\text{SCD}}(\mathbf{x})$$

- **[Nutini et al. ICML15]**: If f is μ_1 strongly convex in the 1-norm[1], then $\frac{\mu_2}{n} \leq \mu_1 \leq \mu_2$

$$\tau_{\text{UCD}}(\mathbf{x}) \leq n \cdot \frac{\mu_1}{\mu_2} \cdot \tau_{\text{SCD}}(\mathbf{x})$$

[1] $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{\mu_p}{2}\|\mathbf{y} - \mathbf{x}\|_p^2$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Approximate Steepest CD (ASCD)

$$i_{\text{ASCD}} \sim_{\text{u.a.r.}} \mathcal{I}$$

- Maintain bounds $\boldsymbol{\ell} \leq |\nabla f(\mathbf{x})| \leq \mathbf{u}$
- Compute active set $\mathcal{I}$, $i_{\text{SCD}} \in \mathcal{I}$
- Special case $|\mathcal{I}| = 1 \Rightarrow \mathcal{I} = \{i_{\text{SCD}}\}$
- We give examples where the additional operations take only $O(n \log n)$ time.
- **Theorem:**

$$\tau_{\text{SCD}}(\mathbf{x}) \geq \tau_{\text{ASCD}}(\mathbf{x}) \geq \tau_{\text{UCD}}(\mathbf{x})$$

- The ratio $\frac{\tau_{\text{SCD}}(\mathbf{x})}{\tau_{\text{ASCD}}(\mathbf{x})}$ depends crucially on the quality of the bounds $\boldsymbol{\ell}, \mathbf{u}$.

Uniform Coordinate Descent (UCD)

$$i_{\text{UCD}} \sim_{\text{u.a.r.}} [n]$$

- **Theorem:** $\exists f$:

$$\tau_{\text{SCD}}(\mathbf{x}) \approx \tau_{\text{UCD}}(\mathbf{x})$$

- **Theorem:** $\exists f$ where $\tau_{\text{SCD}}(\mathbf{x}) \gg \tau_{\text{UCD}}(\mathbf{x})$ and

$$\tau_{\text{ASCD}}(\mathbf{x}) \approx \tau_{\text{SCD}}(\mathbf{x})$$

$\tau_{\text{SCD}}(\mathbf{x}) = \max_{i\in[n]} \tau^{[i]}(\mathbf{x}) = \frac{1}{L}\|\nabla f(\mathbf{x})\|_\infty^2$	$\geq$	$\tau_{\text{ASCD}}(\mathbf{x})$	$\geq$	$\tau_{\text{UCD}}(\mathbf{x}) = \mathbb{E}\left[\tau^{[i]}(\mathbf{x})\right] = \frac{1}{nL}\|\nabla f(\mathbf{x})\|_2^2$
Complexity: $T(\nabla f(\mathbf{x})) = O(dn)$		**Complexity:** $T(\nabla_i f(\mathbf{x})) + O(n\log n) = O(d + n\log n)$		**Complexity:** $T(\nabla_i f(\mathbf{x})) = O(d)$

Details

Safe bounds $\boldsymbol{\ell} \leq |\nabla f(\mathbf{x})| \leq \mathbf{u}$

- Trivial values $[\boldsymbol{\ell}]_i = 0$ and $[\mathbf{u}]_i = \infty$ are admissible, but more accurate bounds give better speed-up.
- Obtained through δ-gradient oracles $g_{ij}\colon \mathbb{R}^n \to \mathbb{R}$:

$$|\nabla_j f(\mathbf{x} + \gamma \mathbf{e}_i) - g_{ij}(\mathbf{x})| \leq |\gamma|\,\delta$$

- **Principal example:** $f(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2$. Then

$$\nabla_j f(\mathbf{x} + \gamma \mathbf{e}_i) - \nabla_i f(\mathbf{x}) = \gamma \langle \mathbf{a}_i, \mathbf{a}_j\rangle$$

Active set $\mathcal{I}$

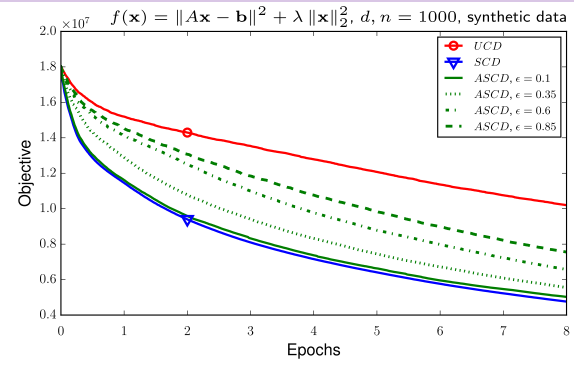
- **Example:** Consider the intervals $I_i = [[\boldsymbol{\ell}]_i, [\mathbf{u}]_i]$:

$$I_1 = [0,2] \quad I_2 = [1,4] \quad I_3 = [2,3] \quad I_4 = [3,4]$$

$$\mathcal{I} := \operatorname*{argmin}_{\mathcal{I}\subseteq[n]} \left\{ [\mathbf{u}]_i^2 < \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} [\boldsymbol{\ell}]_i^2, \forall i \notin \mathcal{I} \right\}$$

- **Theorem:** $\mathcal{I}$ can be computed in $O(n\log n)$ time, $i_{\text{SCD}} \in \mathcal{I}$ and $\tau_{\text{ASCD}}(\mathbf{x}) \geq \tau_{\text{UCD}}(\mathbf{x})$.

ϵ-accuracy of scalar product oracle



Scalar product approximation

- **Oracle:** $S(i,j)\colon [n]\times[n] \to \mathbb{R}$ with $T(S(i,j)) = O(\log n)$ and

$$|S(i,j) - \langle \mathbf{a}_i, \mathbf{a}_j\rangle| \leq \epsilon\,\|\mathbf{a}_i\|\,\|\mathbf{a}_j\|$$

- The bounds $\boldsymbol{\ell}, \mathbf{u}$ can be updated in $O(n\log n)$
- **Examples:**
  - $S(i,j) \equiv 0$ for $\epsilon = 1$
  - Low-dimensional embeddings (Johnson-Lindenstrauss)
  - Use caching techniques to compute and store *some* important scalar products exactly ($\epsilon = 0$ for cache-hit, $\epsilon = 1$ for cache-miss).

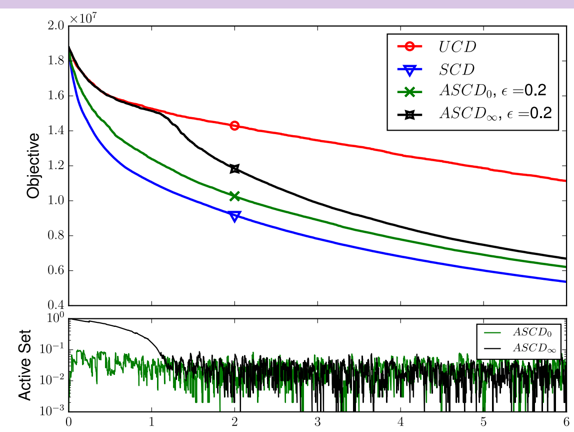
The Algorithm: ASCD

1. Initialize $\mathcal{I}_0 = [n]$, $\boldsymbol{\ell}_0 = \mathbf{0}$, $\mathbf{u}_0 = \boldsymbol{\infty}$
2. For $t \geq 0$ repeat:
   1. Pick $i_t \sim_{\text{u.a.r.}} \mathcal{I}_t$
   2. Compute $\nabla_{i_t} f(\mathbf{x}_t)$
   3. Update $\boldsymbol{\ell}_{t+1}, \mathbf{u}_{t+1}$ with scalar product oracle S
   4. Update $\mathcal{I}_{t+1}(\boldsymbol{\ell}_{t+1}, \mathbf{u}_{t+1})$

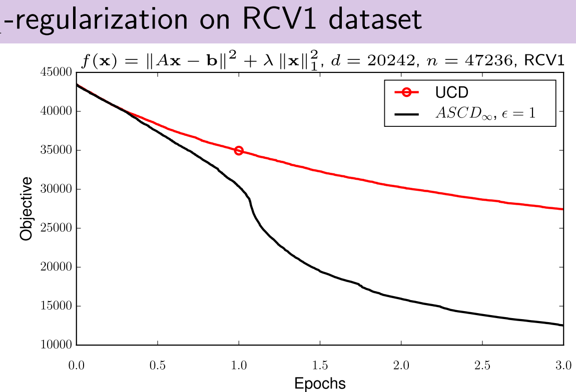
Total complexity:

- Each iteration $T(\nabla_{i_t} f(\mathbf{x}_t)) + O(n\log n) = O(d + n\log n)$
- If $d = \Omega(n)$, ASCD is only $O(\log n)$ more expensive than UCD, but can attain the iteration complexity of SCD!

No-initialization vs. initialization $\boldsymbol{\ell}_0 = \mathbf{u}_0 = \nabla f(\mathbf{x}_0)$



ℓ_1-regularization on RCV1 dataset



Open Problems

- more general/more accurate gradient oracles
- good and efficient scalar product oracles $S(i,j)$
- non-uniform sampling from $\mathcal{I}$
- similar technique for SGD setting?