Universal Coordinate Descent

Olivier Fercoq

Joint work with Peter Richtárik

3-5 September 2014

Introduction

Problem

Minimize a convex composite function

$$\min_{x \in \mathbb{R}^d} [F(x) = f(x) + \psi(x)]$$

Hölder gradient

We say that f has Hölder (sub)gradient ($f \in$ Hölder(M, ν)) if

$$\|\nabla f(x) - \nabla f(y)\|_* \leq M \|x - y\|^{\nu}, \quad \forall x, y \in \mathbb{R}^d$$

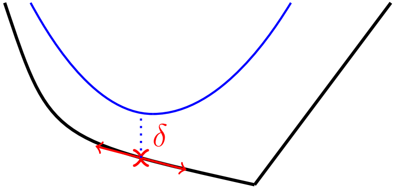
- $\nu = 0$: non-differentiable but bounded subgradients
- $\nu = 1$: differentiable with Lipschitz gradient

Universal gradient methods

- Yu. Nesterov. Universal gradient methods for convex optimization problems, CORE DP #26/2013

- If $f \in$ Hölder(M, ν), then for all $\delta > 0$ and $L \geq \left(\frac{1}{\delta}\right)^{\frac{1-\nu}{1+\nu}} M^{\frac{2}{1+\nu}}$:

$$f(x) \leq f(y) + \langle f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2 + \frac{\delta}{2}$$



- ν “discovered” via a line search

Block coordinate setting

The Problem

$$\min_{x=(x^{(1)},\ldots,x^{(n)})\in\mathbb{R}^d} \left[F(x) = f(x) + \psi(x)\right]$$

Blocks: $x^{(i)} \in \mathbb{R}^{n_i}, \qquad \sum_{i=1}^n n_i = d$

Assumptions

- ∇f **is block-Hölder:** For each block $i \in \{1, 2, \ldots, n\}$ and all $x \in \mathbb{R}^d$ and $h \in \mathbb{R}^{n_i}$,

$$\|\nabla_i f(x + U_i h) - \nabla_i f(x)\| \leq M_i \|h\|^\nu$$

- ψ **is block-separable:** $\psi(x) = \sum_{i=1}^n \psi_i(x^{(i)})$

Line search

Proposition

If $f \in$ *Hölder*(M, ν)*, then for any* $\delta > 0$ *and* $L \geq \left(\frac{1}{\delta}\right)^{\frac{1-\nu}{1+\nu}} M^{\frac{2}{1+\nu}}$ *and all* $x, y \in \mathbb{R}^d$*:*

(1) $f(x) \leq f(y) + \langle f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2 + \frac{\delta}{2}$

(2) $f(y) + \langle \nabla f(y), x - y\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_*^2 - \frac{\delta}{2} \leq f(x)$

(3) $\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_*^2 \leq \frac{L}{2}\|x - y\|^2 + \delta$

- *In fact,* (1) $\Longrightarrow$ (2) $\Longrightarrow$ (3)
- *Moreover, if* (3) *holds for* some L, x, y, δ, *then*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + L\|x - y\|^2 + \delta \qquad (*)$$

Line search: based on "(3) $\Longrightarrow$ (*)" for $f(h) \leftarrow f(x_k + U_i h)$

Universal Coordinate Descent (UCD)

Choose $x_0 \in \mathrm{dom}\,\psi$, $L_0^1, \dots, L_0^n > 0$, $\epsilon > 0$ and $\delta = \frac{\epsilon}{2n}$

For $k \geq 0$ do:

1. **Select block:** $i \in \{1, 2, \dots, n\}$ at random
2. **Line search:** Find the smallest integer s such that for

$$h \leftarrow \arg\min_{h \in \mathbb{R}^{n_i}} f(x_k) + \langle \nabla_i f(x_k), h \rangle + 2^{s-1} L_k^i \|h\|^2 + \psi_i(x_k + h)$$

and $x_+ := x_k + U_i h$, we have

$$\frac{1}{2} \frac{1}{2^{s-1} L_k^i} \|\nabla_i f(x_k) - \nabla_i f(x_+)\|^2 \leq \frac{2^{s-1} L_k^i}{2} \|x_k^{(i)} - x_+^{(i)}\|^2 + \delta$$

3. **Update:** Set $x_{k+1} \leftarrow x_+$ and $L_{k+1}^i \leftarrow 2^s L_k^i$

Complexity

Theorem

The **universal coordinate descent method** *satisfies:*

$$\mathbf{E}[F(\hat{x}_k) - F(x_*)] \leq \frac{n}{k} R_0^2(x_*) + \frac{4n}{k} \bar{\Lambda} \mathcal{R}_M(x_*)^2 + \frac{\epsilon}{2}$$
$$\leq \frac{n}{k} R_0^2(x_*) + \frac{4n^{\frac{2}{1+\nu}}}{k\epsilon^{\frac{1-\nu}{1+\nu}}} \bar{\Lambda} \mathcal{R}^2_{M^{\frac{2}{1+\nu}}}(x_*) + \frac{\epsilon}{2},$$

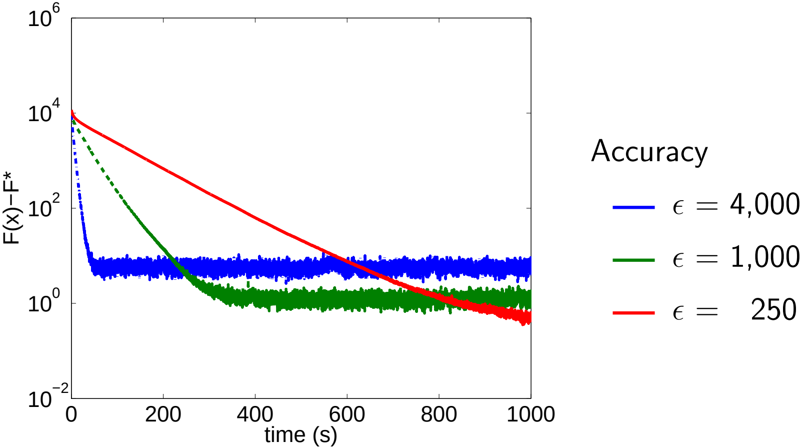
where $\mathcal{R}_M(x_*) = \sup_k \|x_k - x_*\|_M$ and $\bar{\Lambda} = \frac{1}{n} \sum_{i=1}^n \log_2(\frac{2M^i}{L_0^i})$.

$\Rightarrow$ # iterations needed to reach an ϵ-solution

- $\nu = 0$: $O(n^2/\epsilon^2)$
- $\nu = 1$: $O(n/\epsilon)$

Choice of ϵ

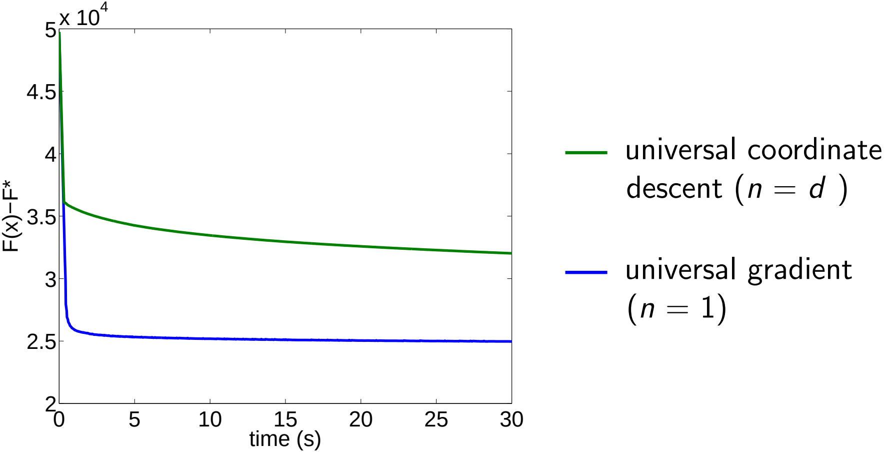
w8a dataset: $d = 300$, $m = 49749$. $F(x) = \|Ax - b\|_1 + \|x\|_1$



No convergence but ϵ-solutions

Subgradient-based coordinate descent

w8a dataset: $d = 300$, $m = 49749$. $F(x) = \|Ax - b\|_1 + \|x\|_1$



Coordinate descent not so efficient in this case: $O(n^2/\epsilon^2)$

Parallel coordinate descent

Studied in:

- P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming* 144(2):1–38, 2014.
- O. Fercoq and P. Richtárik. Smooth minimization of nonsmooth functions with parallel coordinate descent methods. *arXiv:1309.5885*, 2013
- P. Richtárik and M. Takáč. Distributed coordinate descent method for learning with big data. *arXiv:1310.2059*, 2013
- O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent. *arXiv:1312.5799*, 2013

Parallel coordinate descent

- Update several coordinates at the same time: random sampling $\hat{S}$
- **Theory** based on the concept of Expected Separable Overapproximation (ESO):

$$\mathbf{E}\left[f(x + h_{[\hat{S}]})\right] \leq f(x) + \frac{\mathbf{E}[|\hat{S}|]}{n}\left(\langle \nabla f(x), h\rangle + \frac{1}{2}\|h\|_v^2\right)$$

- Vector $v = (v_1, \ldots, v_n)$ depends on $\hat{S}$ and f
- The dependence of v on

$$\tau := \mathbf{E}[|\hat{S}|]$$

determines the **parallelization speedup**

Partial separability

Definition

Assume $f : \mathbb{R}^d \to \mathbb{R}$ is partially separable. That is,

$$f(x) = \sum_{j=1}^{m} f_j(x),$$

where f_j depends on blocks $i \in C_j \subseteq \{1, 2, \ldots, n\}$ only and

$$\omega = \max_j |C_j| \qquad \text{(degree of partial separability)}$$

- Note that $1 \leq \omega \leq n$

Deterministic Separable Overapproximation

Proposition (Non-quadratic DSO)

If $f : \mathbb{R}^d \to \mathbb{R}$ *is convex and partially separable of degree* ω*, then for any* $S \subseteq \{1, \ldots, d\}$ *and* $x, h \in \mathbb{R}^d$*,*

$$f\left(x + \sum_{i \in S} U_i h^{(i)}\right) - f(x) \leq \frac{1}{\omega_S} \sum_{i \in S} \left(f(x + \omega_S U_i h^{(i)}) - f(x)\right),$$

where $\omega_S = \max_j |C_j \cap S| \leq \min(\omega, |S|)$

Line search:

Estimate the curvature of $\phi_i(h^{(i)}) = f(x + \omega_S U_i h^{(i)}) - f(x)$ *independently* for all $i \in S$

Expected Separable Overapproximation

Proposition (Non-quadratic ESO)

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and partially separable of degree ω and $\hat{S}$ be a τ-nice sampling and let

$$\pi_j = \binom{\omega}{j}\binom{n-\omega}{\tau-j}\Big/\binom{n}{\tau}$$

For all $x, h \in \mathbb{R}^d$:

$$\mathbf{E}\left[f(x+h_{[\hat{S}]}) - f(x)\right] \le \sum_{i=1}^{n} \sum_{j=1}^{\min(\omega,\tau)} \frac{\pi_j}{\omega}\left(f(x+jU_i h^{(i)}) - f(x)\right)$$

Simpler ESO

Proposition

The ESO can be further bounded by the following separable function involving a logarithmic number of terms per block:

$$\mathbf{E}\left[f(x + h_{[\hat{S}]}) - f(x)\right] \leq \frac{\tau}{n} \sum_{i=1}^{n} \tilde{f}_x^i(h^{(i)}),$$

where

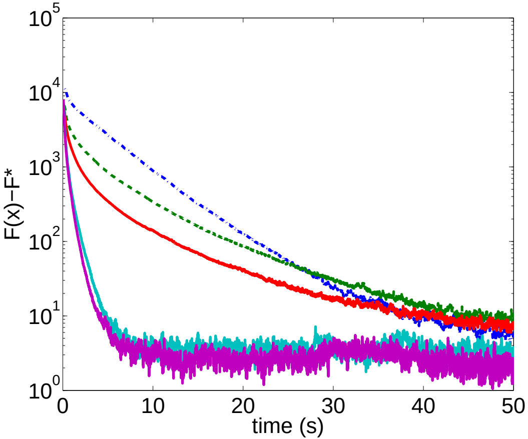
$$\tilde{f}_x^i(h) := \sum_{s=0}^{\lfloor \log_2(\min(\omega,\tau)) \rfloor} \frac{Q_s}{2^{s+1}-1} \left(f(x + (2^{s+1}-1)U_i h^{(i)}) - f(x) \right)$$

$$Q_s = \frac{n}{\omega\tau} \sum_{j=2^s}^{2^{s+1}-1} j\pi_j.$$

Parallelization speedup

w8a dataset: $d = n = 300,\ m = 49,749$

$$F(x) = \|Ax - b\|_1 + \|x\|_1, \quad \epsilon = 4,000$$



processors

- $\tau = 1$
- $\tau = 2$
- $\tau = 4$
- $\tau = 8$
- $\tau = 16$

Acceleration

- For smooth functions rate $O(1/\sqrt{\epsilon})$ instead of $O(1/\epsilon)$
- If nonsmooth: no improvement, rate remains $O(1/\epsilon^2)$
- Arguments of line search and acceleration combine well
- Algorithm a little more complex

Universal Accelerated Coordinate Descent

Choose $x_0 \in \mathrm{dom}\psi$; $L_0^1, \ldots, L_0^n > 0$; $\epsilon > 0$. Set $z_0 = x_0$ and $\theta_0 = \frac{\tau}{n}$.
For $k \geq 0$ do:

1. $y_k = (1 - \theta_k)x_k + \theta_k z_k$
2. Pick a subset of τ blocks $\hat{S}$, uniformly a random
3. In parallel for $i \in \hat{S}$:

   3.1 Find the smallest integer s such that for

$$z_+^{(i)} = \arg\min_{z \in \mathbb{R}^{n_i}} \langle \nabla_i f(y_k), z \rangle + \frac{n\theta_k 2^s L_k^i}{2\tau} \|z - z_k^{(i)}\|^2 + \psi_i(z)$$

$$h_k^{(i)} = \frac{n\theta_k}{\tau}(z_+^{(i)} - z_k^{(i)})$$

   we have $\frac{1}{2}\frac{1}{2^{s-1}L_k^i}\|\nabla_i f(y_k) - \nabla_i \tilde{f}_{y_k}^i(h_k^{(i)})\|^2 \leq \frac{2^{s-1}L_k^i}{2}\|h_k^{(i)}\|^2 + \frac{\epsilon\theta_k}{2\tau}$

   3.2 Set $L_{k+1}^i = 2^s L_k^i$ and $x_{k+1}^{(i)} = y_k^{(i)} + h_k^{(i)}$

4. $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$

Complexity

Theorem

The universal accelerated coordinate descent method satisfies

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau+2n)^2}\left(\left(1-\frac{\tau}{n}\right)(F(x_0)-F(x_*)) + \frac{1}{2}R^2_{L_0}(x_*)\right) + \left(\frac{2n}{k\tau+2n}\right)^{\frac{1+3\nu}{1+\nu}} \frac{(2n)^{\frac{1-\nu}{1+\nu}}\beta_\nu C}{\epsilon^{\frac{1-\nu}{1+\nu}}} \mathcal{R}^2_{M^{\frac{2}{1+\nu}}}(x_*) + \frac{\epsilon}{2}$$

where $\mathcal{R}_M(x_*) = \sup_k \|x_k - x_*\|_M,\ C = 2\left(\bar{\Lambda} + \frac{1-\nu}{1+\nu}\log\left(\frac{k\tau+2n}{\epsilon}\right)\right)$

$$\beta_\nu = \sum_{j=1}^{n} \frac{n\pi_j}{\tau\omega} j^{1+\nu},\ \bar{\Lambda} = \frac{\tau}{n}\sum_{i=1}^{n} \log_2 \frac{4\beta_\nu (M_i)^{\frac{2}{1+\nu}}}{L_0^i}$$

Accelerated Adaboost

- Semi-supervised classification
- A: matrix of characteristics
  b: labels
- Adaboost = greedy coordinate descent for

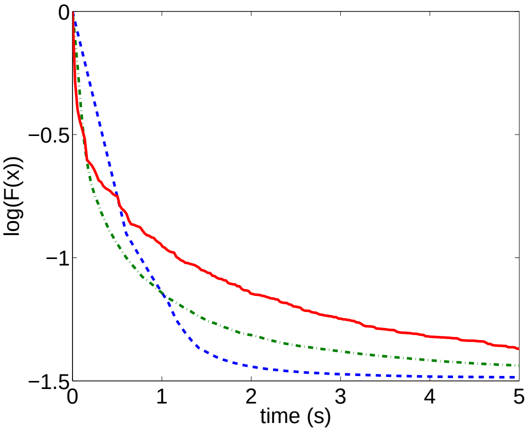
$$F(x) = \frac{1}{m} \sum_{j=1}^{m} \exp(b_j \sum_{i=1}^{n} A_{j,i} x_i)$$

- Differentiable but not globally Lipschitz

Numerical experiment

w8a dataset: $d = 300$, $m = 49749$

$F(x) = \frac{1}{m} \sum_{j=1}^{m} \exp(b_j \sum_{i=1}^{n} A_{j,i} x_i)$, $\epsilon = 0$, 16 processors



Coordinate descent algorithm:

- Exact greedy
- Line search
- Line search + Acceleration

Conclusion

Summary:

- Universal coordinate descent algorithms: accelerated, parallel, proximal
- Versatile line search
- Coordinates tackled independently

Open questions:

- Improve the theoretical bound on nonsmooth problems from $O(n^2/\epsilon^2)$ to $O(n/\epsilon^2)$
- Decrease the curvature estimate on the run
- Use non-quadratic ESO with exact minimization (for smooth functions)