

# Smooth Primal-Dual Coordinate Descent Algorithms for Nonsmooth Convex Optimization

A. Alacaoglu, V. Cevher, *O. Fercoq*, Q. Tran-Dinh

8 January 2019



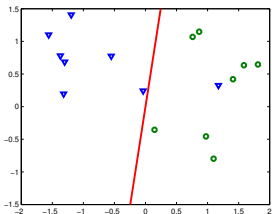
THE UNIVERSITY  
of NORTH CAROLINA  
at CHAPEL HILL

## Composite convex optimisation

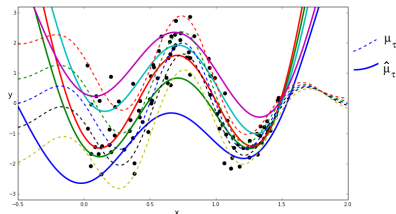
$$\min_{x \in \mathcal{X}} f(x) + g(x) + h(Ax)$$

- $f : \mathcal{X} \rightarrow \mathbb{R}$  is a differentiable convex function with  $L(\nabla_i f)$ -Lipschitz partial derivatives
- $g : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$  is a l.s.c. convex function with a simple proximal operator
- $h : \mathcal{Y} \rightarrow \mathbb{R} \cup \{+\infty\}$  is a l.s.c. convex function with a simple proximal operator
- $A : \mathcal{X} \rightarrow \mathcal{Y}$  is a linear map

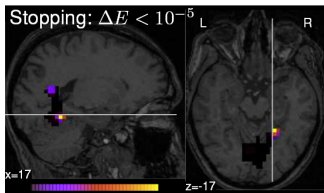
# Examples



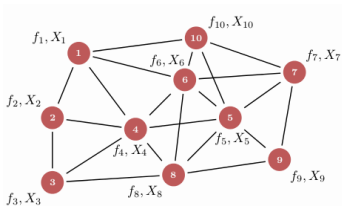
Support Vector Machines  
with intercept



Quantile regression



TV-regularized regression



©Mota

Distributed optimization

# Primal-dual algorithms

- Equivalent saddle point formulation  
if  $0 \in \text{relint}(\text{dom } h - A \text{ dom } g)$

$$\begin{aligned} \min_{x \in \mathcal{X}} f(x) + g(x) + h(Ax) \\ = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x) + g(x) + \langle Ax, y \rangle - h^*(y) \end{aligned}$$

where  $h^*(y) = \sup_{z \in \mathcal{Y}} \langle z, y \rangle - h(z)$  (Fenchel conjugate)

- Example: Vũ-Connat method

$$\begin{aligned} y_{k+1} &= \text{prox}_{\sigma h^*}(y_k + Ax_k) \\ x_{k+1} &= \text{prox}_{\tau g}(x_k - \tau(\nabla f(x_k) + A^\top(2y_{k+1} - y_k))) \end{aligned}$$

Splits  $f$ ,  $g$ ,  $h$  and  $A$ .

## Coordinate descent

- Updates 1 coordinate per iteration:
  - needs many iterations
  - efficient updates
  - allows longer step sizes
- Comparison gradient descent vs coordinate descent  
for  $g = h = 0$  and  $f(x) = \|Mx - b\|_2^2$

Algo	$x^{k+1} = x^k - \frac{1}{L(\nabla f)} \nabla f(x^k)$	$x_i^{k+1} = x_i^k - \frac{1}{L(\nabla_i f)} \nabla_i f(x^k)$
Parameter	$L(\nabla f) = \ M\ _2^2$	$L(\nabla_i f) = \ Me_i\ _2^2$
Complexity	$O(L(\nabla f)/k)$	$O(n \max_i L(\nabla_i f)/k)$
Cost 1 iter	$2 \text{nnz}(M)$	$2 \text{nnz}(Me_i)$
Total cost	$O(\text{nnz}(M)L(\nabla f)/k)$	$O(\text{nnz}(M) \max_i L(\nabla_i f)/k)$

## Previous works

[Hong *et al.*, 2014]

no convergence rate, “sufficiently small” step sizes

[Combettes, Pesquet, 2014]+[Bianchi, Hachem, Iutzeler, 2016]

no convergence rate, small step sizes

[Fercoq, Bianchi, 2015]

no convergence rate (add-on in 2018), longer step sizes

[Zhang, Xiao, 2017]

only strongly convex-concave lagrangians + all primal variables updated together

[Gao, Xu, Zhang, 2017] + [Chambolle, Ehrhardt, Richtárik, Schönlieb, 2017]

all dual variables updated together

## Idea of this work

Combine features from:

- Smooth minimization of nonsmooth functions  
[Nesterov, 2005] + [Fercoq, Richtárik, 2013]
- Accelerated proximal coordinate descent  
[Fercoq, Richtárik, 2015]
- Accelerated smoothed gap reduction algorithm  
[Tran-Dinh, Fercoq, Cevher, 2017]

## Smoothing nonsmooth functions

- We define the smoothed function [Nesterov 2005]

$$h_\beta(z; \dot{y}) = \max_y \langle z, y \rangle - h^*(y) - \frac{\beta}{2} \|y - \dot{y}\|_y^2$$

- $h_\beta$  is differentiable wrt  $z$  and  $\nabla_z h_\beta$  is  $\frac{1}{\beta}$ -Lipschitz
- $h_\beta(Ax^k, \dot{y}) = \langle \dot{y}, Ax^k - c \rangle + \frac{1}{2\beta} \|Ax^k - c\|_{y,*}^2$



## Fundamental theorem

$$F = f + g ; S_{\beta}(x, y) = F(x) + h_{\beta}(Ax; y) - F(x^*) - h(Ax^*)$$

### Theorem (Nesterov)

*Suppose that  $L(h) < +\infty$ . Then*

$$F(x) + h(Ax) - F(x^*) - h(Ax^*) \leq S_{\beta}(x; y) + 2\beta L(h)$$

## Fundamental theorem

$$F = f + g ; S_{\beta}(x, \dot{y}) = F(x) + h_{\beta}(Ax; \dot{y}) - F(x^*) - h(Ax^*)$$

### Theorem (Nesterov)

Suppose that  $L(h) < +\infty$ . Then

$$F(x) + h(Ax) - F(x^*) - h(Ax^*) \leq S_{\beta}(x; \dot{y}) + 2\beta L(h)$$

### Theorem (Tran-Dinh, Fercoq, Cevher)

Consider  $h = \delta_{\{c\}}$

$$\begin{cases} \|Ax - c\|_{\mathcal{Y},*} \leq \beta \left[ \|y^* - \dot{y}\|_{\mathcal{Y}} + (\|y^* - \dot{y}\|_{\mathcal{Y}}^2 + 2\beta^{-1} S_{\beta}(x; \dot{y}))^{1/2} \right] \\ F(x) - F(x^*) \geq -\|y^*\|_{\mathcal{Y}} \|Ax - c\|_{\mathcal{Y},*} \\ F(x) - F(x^*) \leq S_{\beta}(x, \dot{y}) + \|y^*\|_{\mathcal{Y}} \|Ax - c\|_{\mathcal{Y},*} + \frac{\beta}{2} \|y^* - \dot{y}\|_{\mathcal{Y}}^2 \end{cases}$$

If  $\beta$  and  $S_{\beta}(x, \dot{y})$  are small, we have an approximate solution

# Idea of the algorithm

- Run the accelerated proximal coordinate descent method on  $F(x) + h_\beta(Ax; \dot{y})$   
 $\Rightarrow S_\beta(x, \dot{y})$  decreases
- Reduce  $\beta$  at each iteration  
 $\Rightarrow \beta$  decreases but  $S_\beta(x, \dot{y})$  increases

# SMooth, Accelerate, Randomize The Coordinate Descent (SMART-CD)

Choose  $\beta_1 > 0$ ,  $x^0 \in \mathcal{X}$  and probabilities  $q \in \mathbb{R}^n$

Set  $\tau_0 = \min_{1 \leq i \leq n} q_i$  and  $\bar{x}^0 = \tilde{x}^0 = x^0$

For  $k \in \{0, 1, \dots, k_{\max}\}$

$$\hat{x}^k = (1 - \tau_k)\bar{x}^k + \tau_k\tilde{x}^k$$

Select  $i_k \sim q$

$$B_{i_k}^k = L(\nabla_{i_k}(f + h_{\beta_{k+1}} \circ A))$$

$$\tilde{x}_{i_k}^{k+1} = \text{prox}_{\frac{\tau_0}{\tau_k B_{i_k}^k} g_{i_k}} \left( \tilde{x}_{i_k}^k - \frac{\tau_0}{\tau_k B_{i_k}^k} \nabla_{i_k}(f + h_{\beta_{k+1}} \circ A)(\hat{x}^k) \right)$$

$$\bar{x}^{k+1} = \hat{x}^k + \frac{\tau_k}{\tau_0}(\tilde{x}^{k+1} - \tilde{x}^k)$$

$$\tau_{k+1} \in (0, 1) \text{ solution to } \tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$$

$$\beta_{k+2} = \frac{\beta_{k+1}}{1 + \tau_{k+1}}$$

# SMooth, Accelerate, Randomize The Coordinate Descent (SMART-CD)

Choose  $\beta_1 > 0$ ,  $x^0 \in \mathcal{X}$  and probabilities  $q \in \mathbb{R}^n$

Set  $\tau_0 = \min_{1 \leq i \leq n} q_i$  and  $\bar{x}^0 = \tilde{x}^0 = x^0$

For  $k \in \{0, 1, \dots, k_{\max}\}$

$$\hat{x}^k = (1 - \tau_k)\bar{x}^k + \tau_k \tilde{x}^k$$

Select  $i_k \sim q$

$$B_{i_k}^k = L(\nabla_{i_k} f) + \frac{\|A_{i_k}\|^2}{\beta_{k+1}}$$

$$y_k^* = \text{prox}_{\beta_{k+1}^{-1} h^*}(\dot{y} + \beta_{k+1}^{-1} A \hat{x}^k)$$

$$\tilde{x}_{i_k}^{k+1} = \text{prox}_{\frac{\tau_0}{\tau_k B_{i_k}^k} g_{i_k}}\left(\tilde{x}_{i_k}^k - \frac{\tau_0}{\tau_k B_{i_k}^k} (\nabla_{i_k} f(\hat{x}^k) + A_{i_k}^\top y_k^*)\right)$$

$$\bar{x}^{k+1} = \hat{x}^k + \frac{\tau_k}{\tau_0} (\tilde{x}^{k+1} - \tilde{x}^k)$$

$$\tau_{k+1} \in (0, 1) \text{ solution to } \tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$$

$$\beta_{k+2} = \frac{\beta_{k+1}}{1 + \tau_{k+1}}$$

## Implementation issues

Each iteration should be cheap

- $\nabla_{i_k} f(\hat{x}^k)$  computed using partial matrix-vector products that are stored and updated
- Full-dimensional operation  $\bar{x}^{k+1} = \hat{x}^k + \frac{\tau_k}{\tau_0}(\tilde{x}^{k+1} - \tilde{x}^k)$  treated by a change of variable as for accelerated coordinate descent
- $y_k^* = \text{prox}_{\beta_{k+1}^{-1} h^*}(\dot{y} + \beta_{k+1}^{-1} A \hat{x}^k)$ :  
we only compute the coordinates that are necessary  
(a few of them only if  $h^*$  is partially separable)

## Efficient implementation

Choose  $\beta_1 > 0$ ,  $x^0 \in \mathcal{X}$  and probabilities  $q \in \mathbb{R}^n$

Set  $\tau_0 = \min_{1 \leq i \leq n} q_i$  and  $\bar{x}^0 = \tilde{x}^0 = x^0$

For  $k \in \{0, 1, \dots, k_{\max}\}$

Select  $i_k \sim q$

$$B_{i_k}^k = L(\nabla_{i_k} f) + \frac{\|A_{i_k}\|^2}{\beta_{k+1}}$$

$$y_k^* = \text{prox}_{\beta_{k+1}^{-1} h^*}(\dot{y} + \beta_{k+1}^{-1}(c_k A \check{x}^k + A \tilde{x}^k))$$

$$\tilde{x}_{i_k}^{k+1} = \text{prox}_{\frac{\tau_0}{\tau_k B_{i_k}^k} g_{i_k}}\left(\tilde{x}_{i_k}^k - \frac{\tau_0}{\tau_k B_{i_k}^k} (\nabla_{i_k} f(c_k A \check{x}^k + A \tilde{x}^k) + A_{i_k}^\top y_k^*)\right)$$

$$\check{x}_{i_k}^{k+1} = \check{x}_{i_k}^k - \frac{1 - \tau_k / \tau_0}{c_k} (\tilde{x}_{i_k}^{k+1} - \tilde{x}_{i_k}^k)$$

$$\tau_{k+1} \in (0, 1) \text{ solution to } \tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$$

$$\beta_{k+2} = \frac{\beta_{k+1}}{1 + \tau_{k+1}}$$

$$c_{k+1} = (1 - \tau_{k+1}) c_k$$

## Speed of convergence

$$F = f + g, \tau_0 = \min_{1 \leq i \leq n} q_i \in O(1/n)$$

### Theorem

*SMART-CD has a  $O(n/k)$  convergence speed. If  $h = \delta_{\{c\}}$ ,*

$$\mathbb{E}(\|A\bar{x}^k - c\|_*) \leq \frac{\beta_1}{\tau_0(k-1)+1} \left[ \|y^* - \dot{y}\| + (\|y^* - \dot{y}\|^2 + 4\beta_1^{-1}C^*)^{1/2} \right]$$

$$\mathbb{E}(F(\bar{x}^k) - F(x^*)) \geq -\|y^*\| \mathbb{E}(\|A\bar{x}^k - c\|_*)$$

$$\mathbb{E}(F(\bar{x}^k) - F(x^*)) \leq \frac{(1-\tau_0)C^*}{\tau_0(k-1)+1} + \frac{\beta_1\|y^* - \dot{y}\|^2}{2(\tau_0(k-1)+1)} + \|y^*\| \mathbb{E}(\|A\bar{x}^k - c\|_*)$$

where  $C^* = (1 - \tau_0)(F_{\beta_0}(x^0) - F(x^*)) + \sum_{i=1}^n \frac{\tau_0 B_i^0}{2q_i} \|x_i^* - \tilde{x}_i^0\|_i^2$

Same kind of results for Lipschitzian  $h$



## Sketch of proof

- $\bar{x}^k$  is a convex combination of the  $\tilde{x}^l$ 's ( $l \leq k$ )
- Denote  $\hat{F}_{\beta_k}^k = f(\bar{x}^k) + \sum_{l=0}^k \gamma^{k,l} g(\tilde{x}^l) + h_{\beta_k}(A\bar{x}^k; y)$   
If  $\beta_{k+1}(1 + \tau_k) - \beta_k \geq 0$ , then

$$\begin{aligned} & \mathbb{E}(\hat{F}_{\beta_{k+1}}^{k+1} - F^*) + \mathbb{E}\left[\sum_{i=1}^n \frac{\tau_k^2 B_i^k}{2q_i \tau_0} \|x_i^* - \tilde{x}_i^{k+1}\|_{(i)}^2\right] \\ & \leq (1 - \tau_k) \mathbb{E}(\hat{F}_{\beta_k}^k - F^*) + \mathbb{E}\left[\sum_{i=1}^n \frac{\tau_k^2 B_i^k}{2q_i \tau_0} \|x_i^* - \tilde{x}_i^k\|_{(i)}^2\right] \end{aligned}$$

- Choose  $\tau_k$  so that  $\tau_k^2 B_i^k \leq (1 - \tau_k) \tau_{k-1} B_i^{k-1}$
- Show that  $\tau_k \in O(1/k)$ ,  $\beta_k \in O(1/k)$  and  $\frac{\tau_k^2}{\beta_k} \in O(1/k)$

# Restart

Like accelerated gradient methods, one can restart SMART-CD

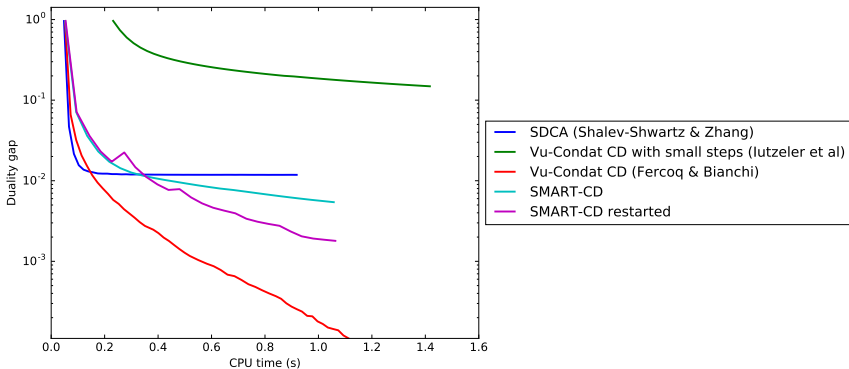
$$\left\{ \begin{array}{l} \tilde{x}^k \leftarrow \bar{x}^k, \\ \dot{y} \leftarrow y_{\beta_{k+1}}^*(\bar{x}^k; \dot{y}), \\ \beta_{k+1} \leftarrow \beta_1, \\ \tau_k \leftarrow \tau_0, \end{array} \right.$$

→ Analysis is not that easy

# Experiment on Support Vector Machines

SVM problem on RCV1 dataset  $m = 20,242$  and  $n = 47,236$

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2\lambda} \|X^\top D(y)\alpha\|^2 - e^\top \alpha + I_{[0,c]}(\alpha) + I_{y^\perp}(\alpha)$$



## Distributed training for logistic regression 1/2

- Data distributed across  $M$  machines,  $R(w) = \sum_{i=1}^p |w_i|$

$$\min_{w \in \mathbb{R}^{p+1}} \sum_{m=1}^M \sum_{j=1}^{n_m} \log(1 + \exp(-y_j x_j^\top w)) + R(w)$$

- Equivalent problem

$$\min_{\mathbf{w} \in \mathbb{R}^{(p+1) \times M}} \sum_{m=1}^M \left( \sum_{j=1}^{n_m} \log(1 + \exp(-y_j x_j^\top w_m)) + \frac{1}{M} R(w_m) \right) + I_C(\mathbf{w})$$

where  $C = \{\mathbf{w} \in \mathbb{R}^{(p+1) \times M} : w_1 = \dots = w_M\}$

## Distributed training for logistic regression 2/2

- master / slave implementation

For  $k \in \mathbb{N}$

For  $m \in \{1, \dots, M\}$

Select at random  $i_m^k \in \{0, \dots, p\}$

$$(z_{m,i_m^k}^*)^k = \frac{1}{\beta_{k+1}} (\tilde{w}_{m,i_m^k} - (P_C(\tilde{w}))_{i_m^k}) + \frac{c_k}{\beta_{k+1}} (\check{w}_{m,i_m^k} - (P_C(\check{w}))_{i_m^k})$$

$$B_{m,i_m^k}^k = L(\nabla_{i_m^k} f_m) + \frac{1}{\beta_{k+1}}$$

$$\tilde{w}_{m,i_k}^{k+1} = \text{prox}_{\frac{\tau_0}{\tau_k B_{m,i_k}^k} R_{i_m^k}^k} \left( \tilde{w}_{m,i_m^k}^k - \frac{\tau_0}{\tau_k B_{m,i_m^k}^k} (\nabla_{i_m^k} f_m(\hat{w}_m^k) + (z_{m,i_m^k}^*)^k) \right)$$

$$\check{w}_{m,i_k}^{k+1} = \check{w}_{m,i_k}^k - \frac{1-\tau_k/\tau_0}{c_k} (\tilde{w}_{m,i_k}^{k+1} - \tilde{w}_{m,i_k}^k)$$

Update  $P_C(\tilde{w}) = \frac{1}{M} \sum_{m=1}^M \tilde{w}_m$  and  $P_C(\check{w}) = \frac{1}{M} \sum_{m=1}^M \check{w}_m$

$\tau_{k+1} \in (0, 1)$  solution to  $\tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$

$$\beta_{k+2} = \frac{\beta_{k+1}}{1+\tau_{k+1}}$$

## Distributed training for logistic regression 2/2

- master / slave implementation – communicated values

For  $k \in \mathbb{N}$

For  $m \in \{1, \dots, M\}$

Select at random  $i_m^k \in \{0, \dots, p\}$

$$(z_{m,i_m^k}^*)^k = \frac{1}{\beta_{k+1}} (\tilde{w}_{m,i_m^k} - (P_C(\tilde{w}))_{i_m^k}) + \frac{c_k}{\beta_{k+1}} (\check{w}_{m,i_m^k} - (P_C(\check{w}))_{i_m^k})$$

$$B_{m,i_m^k}^k = L(\nabla_{i_m^k} f_m) + \frac{1}{\beta_{k+1}}$$

$$\tilde{w}_{m,i_k}^{k+1} = \text{prox}_{\frac{\tau_0}{\tau_k B_{i_m^k}^k} R_{i_m^k}^k} \left( \tilde{w}_{m,i_m^k}^k - \frac{\tau_0}{\tau_k B_{m,i_m^k}^k} (\nabla_{i_m^k} f_m(\hat{w}_m^k) + (z_{m,i_m^k}^*)^k) \right)$$

$$\check{w}_{m,i_k}^{k+1} = \check{w}_{m,i_k}^k - \frac{1-\tau_k/\tau_0}{c_k} (\tilde{w}_{m,i_k}^{k+1} - \tilde{w}_{m,i_k}^k)$$

Update  $P_C(\tilde{w}) = \frac{1}{M} \sum_{m=1}^M \tilde{w}_m$  and  $P_C(\check{w}) = \frac{1}{M} \sum_{m=1}^M \check{w}_m$

$\tau_{k+1} \in (0, 1)$  solution to  $\tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$

$$\beta_{k+2} = \frac{\beta_{k+1}}{1+\tau_{k+1}}$$

# Conclusion

## Summary

- Designed a primal-dual coordinate descent method
- Rate in  $O(n/k)$
- Good performance in practice

## Related work

- Speed of convergence of Vu-Condat CD
- Smooth gap reduction schemes based on Frank-Wolfe and the stochastic gradient method