

# Accelerated, Parallel and Proximal Coordinate Descent

Olivier Fercoq

Joint work with P. Bianchi & P. Richtárik

24 October 2014

arXiv:1312.5799

# Outline

1. Randomized coordinate descent with non-uniform sampling
2. Accelerated, parallel and proximal coordinate descent

# Minimisation of composite functions

Minimise the composite function  $F$  for  $x \in \mathbb{R}^N$

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

- $f : \mathbb{R}^N \rightarrow \mathbb{R}$ , convex, differentiable, not strongly convex
- $\psi : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ , convex, separable

$$\psi(x) = \sum_{i=1}^n \psi_i(x^{(i)})$$

## Examples

- Dual of Support Vector Machines:  $\psi = I_{[0,1]^N}$

$$\min_{x \in [0,1]^N} \frac{1}{2\lambda N^2} \sum_{j=1}^m \left( \sum_{i=1}^N b_i A_{ji} x^{(i)} \right)^2 - \frac{1}{N} \sum_{i=1}^N x^{(i)}$$

- $L_1$ -regularised least squares:  $\psi = \lambda \|\cdot\|_1$

$$\min_{x \in \mathbb{R}^N} \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1$$

- $L_1$ -regularised  $L_1$  regression

$$\min_{x \in \mathbb{R}^N} \sum_{j=1}^m |e_j^T Ax - b_j| + \lambda \|x\|_1 \xrightarrow{\text{smoothing}} \sum_{j=1}^m f_{\mu}^j(e_j^T Ax - b_j) + \lambda \|x\|_1$$

- Dual Total Variation:  $\psi(x) = \sum_{i=1}^n I_{\{\|y\|_2 \leq \alpha\}}(x^{(i)})$

$$\min_{x \in \mathbb{R}^N} \frac{1}{2} \|b - \nabla^* x\|^2 + \psi(x)$$

## Coordinate descent

$f$  has Lipschitz continuous directional derivatives:

$$f(x + te_i) \leq f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$$

At each iteration:

1. Choose randomly a coordinate  $i$
2. Compute the update  $t \in \mathbb{R}$  that minimises the overapproximation of  $f(x + te_i)$
3. Update the variable  $x \leftarrow x + te_i$

Remarks :

- Many iterations are needed
- Each iteration is cheap
- Famous in machine learning

## Proximal coordinate descent

$$F = f + \psi:$$

$$F(x + te_i) \leq \psi(x + te_i) + f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$$

At each iteration:

1. Choose randomly a coordinate  $i$
2. Compute the update  $t \in \mathbb{R}$  that minimises the overapproximation of  $F(x + te_i)$
3. Update the variable  $x \leftarrow x + te_i$

Remarks :

- Deals with non-smooth regularisers
- Involves the proximity operator of  $\psi_i$ :  $\text{prox}_{\psi_i}$
- With uniform sampling, proof of convergence very similar to proximal gradient's

## Non-uniform probabilities

- Coordinate  $i$  selected with probability

$$p_i = \mathbf{P}(i = \hat{i})$$

- Convergence in [Tseng, Yun, 2009], [Bianchi et al. 2014]: no speed
- The expected decrease of the objective is

$$\mathbf{E} [f(x + te_{\hat{i}}) + \psi(x + te_{\hat{i}})] \leq f(x) + \psi(x)$$

$$+ \sum_{i=1}^n p_i (\psi_i(x^{(i)} + t) - \psi_i(x^{(i)})) + p_i \nabla_i f(x) h^{(i)} + p_i \frac{L_i}{2} (h^{(i)})^2$$

- Biased gradient estimation

# Algorithms

- [Nesterov, 2012]:  $\psi = 0$

$$x_{k+1} = x_k - \nabla_{i_{k+1}} f(x_k) e_{i_{k+1}}$$

$$\mathbf{E} [f(x - \nabla_{\hat{i}} f(x_k) e_{\hat{i}})] \leq f(x) + \sum_{i=1}^n \frac{p_i}{2L_i} (\nabla_i f(x))^2$$

- [Qu, Richtárik, 2014]:  $\underline{p} = \min_{1 \leq i \leq n} p_i$ . For  $i = i_{k+1} \sim \hat{i}$ ,

$$x_{k+1}^{(i)} = (1 - \underline{p} p_i^{-1}) x_k^{(i)} + \underline{p} p_i^{-1} \text{prox}_{\frac{p_i}{L_i \underline{p}} \psi_i} \left( x_k^{(i)} - \frac{p_i}{L_i \underline{p}} \nabla_i f(x_k) \right)$$

- [Bianchi, Fercoq, 2014]: for  $i = i_{k+1} \sim \hat{i}$ ,

$$x_{k+1}^{(i)} = \text{prox}_{\frac{1}{L_i} \psi_i} \left( x_k^{(i)} - \frac{1}{L_i} \nabla_i f(x_k) \right)$$



# Convergence

## Theorem

If  $\forall k, i_{k+1} \sim \hat{i}$  with  $p_i = \mathbf{P}(i = \hat{i})$ ,  $\underline{p} = \min_{1 \leq i \leq n} p_i$  and

$$x_{k+1}^{(i)} = \begin{cases} \text{PROX}_{\frac{1}{L_i} \psi_i} \left( x_k^{(i)} - \frac{1}{L_i} \nabla_i f(x_k) \right) & i = i_{k+1} \\ x_k^{(i)} & i \neq i_{k+1} \end{cases}$$

$$\mathbf{E}[F(x_{k+1}) - F(x_*)]$$

$$\leq \frac{1}{k + \underline{p}^{-1}} \left( (\underline{p}^{-1} - 1)(F(x_0) - F(x_*)) + \frac{1}{2} \|x_0 - x_*\|_{L_{\underline{p}^{-1}}}^2 \right)$$

**Idea of proof:** Take the usual proof and replace at some point  $x_*^{(i)}$  by  $(1 - \underline{p} p_i^{-1}) x_k^{(i)} + \underline{p} p_i^{-1} x_*^{(i)}$ .

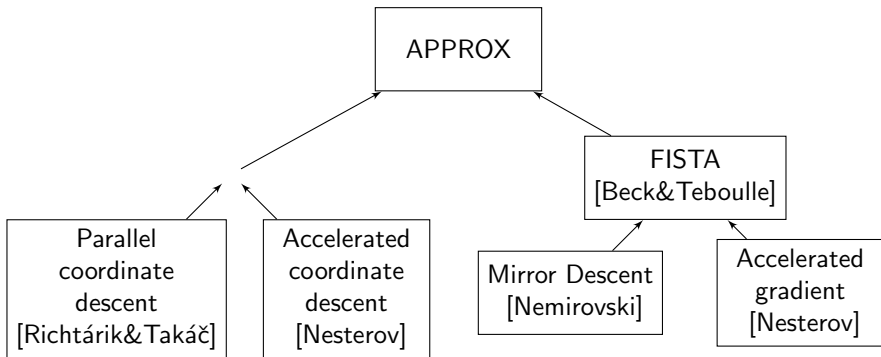
## Accelerated gradient

$k$  = iteration counter

Algorithm	Convergence	Assumptions
Sub-gradient	$\mathcal{O}(1/\sqrt{k})$	convex
Gradient	$\mathcal{O}(1/k)$	smooth, convex
Accelerated gradient [Nesterov, 1983]	$\mathcal{O}(1/k^2)$	smooth, convex
Gradient	$\mathcal{O}((1 - \kappa)^k)$	smooth, strongly convex
Accelerated gradient	$\mathcal{O}((1 - \sqrt{\kappa})^k)$	smooth, strongly convex

Similar convergence rates for coordinate descent

# Context



[O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent]

# APPROX: **A**ccelerated, **P**arallel and **PROX**imal coordinate descent

$x_0 \in \text{dom } \psi$ ,  $z_0 = x_0$ ,  $\theta_0 = \frac{\tau}{n}$  and  $\tau = \mathbf{E}[|\hat{S}|]$

**for**  $k \geq 0$  **do**

$$y_k = (1 - \theta_k)x_k + \theta_k z_k$$

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

$$z_{k+1} \leftarrow z_k$$

**for**  $i \in S_{k+1}$  **do**

$$z_{k+1}^{(i)} = \underset{z \in \mathbb{R}^{N_i}}{\text{argmin}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

**end for**

$$x_{k+1} = y_k + \frac{n}{\tau} \theta_k (z_{k+1} - z_k)$$

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

**end for**

# Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \text{dom } \psi$ ,  $z_0 = x_0$ ,  $\theta_0 = \frac{\tau}{n}$  and  $\tau = \mathbf{E}[|\hat{S}|]$

**for**  $k \geq 0$  **do**

$$y_k = (1 - \theta_k)x_k + \theta_k z_k$$

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

$$z_{k+1} \leftarrow z_k$$

**for**  $i \in S_{k+1}$  **do**

$$z_{k+1}^{(i)} = \underset{z \in \mathbb{R}^{N_i}}{\text{argmin}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

**end for**

$$x_{k+1} = y_k + \frac{n}{\tau} \theta_k (z_{k+1} - z_k)$$

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

**end for**

# Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \text{dom } \psi$ ,  $z_0 = x_0$ ,  $\theta_0 = \frac{\tau}{n}$  and  $\tau = \mathbf{E}[|\hat{S}|]$

**for**  $k \geq 0$  **do**

$$y_k = (1 - \theta_k)x_k + \theta_k z_k$$

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

$$z_{k+1} \leftarrow z_k$$

**for**  $i \in S_{k+1}$  **do**

$$z_{k+1}^{(i)} = \underset{z \in \mathbb{R}^{N_i}}{\text{argmin}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

**end for**

$$x_{k+1} = y_k + \frac{n}{\tau} \theta_k (z_{k+1} - z_k)$$

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

**end for**

# Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \text{dom } \psi$ ,  $z_0 = x_0$ ,  $\theta_0 = \frac{\tau}{n}$  and  $\tau = \mathbf{E}[|\hat{S}|]$

**for**  $k \geq 0$  **do**

$$y_k = (1 - \theta_k)x_k + \theta_k z_k$$

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

$$z_{k+1} \leftarrow z_k$$

**for**  $i \in S_{k+1}$  **do**

$$z_{k+1}^{(i)} = \underset{z \in \mathbb{R}^{N_i}}{\text{argmin}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

**end for**

$$x_{k+1} = y_k + \frac{n}{\tau} \theta_k (z_{k+1} - z_k)$$

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

**end for**

# Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \text{dom } \psi$ ,  $z_0 = x_0$ ,  $\theta_0 = \frac{\tau}{n}$  and  $\tau = \mathbf{E}[|\hat{S}|]$

**for**  $k \geq 0$  **do**

$$y_k = (1 - \theta_k)x_k + \theta_k z_k$$

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

$$z_{k+1} \leftarrow z_k$$

**for**  $i \in S_{k+1}$  **do**

$$z_{k+1}^{(i)} = \underset{z \in \mathbb{R}^{N_i}}{\text{argmin}} \left\langle \nabla_i f(y_k), z - y_k^{(i)} \right\rangle + \frac{n\theta_k\beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

**end for**

$$x_{k+1} = y_k + \frac{n}{\tau}\theta_k(z_{k+1} - z_k)$$

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

**end for**

Communication



# Efficient implementation

Idea of [Lee & Sidford, 2013]

**for**  $k \geq 0$  **do**

Generate a random set of coordinates  $S_{k+1} \sim \hat{S}$

**for**  $i \in S_{k+1}$  **do**

$$t_k^{(i)} = \arg \min_{t \in \mathbb{R}^{N_i}} \langle \nabla_i f(\theta_k^2 u_k + \tilde{z}_k), t \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|t\|_{(i)}^2 + \psi_i(\tilde{z}_k^{(i)} + t)$$

$$\tilde{z}_{k+1}^{(i)} = \tilde{z}_k^{(i)} + t_k^{(i)}$$

$$u_{k+1}^{(i)} = u_k^{(i)} - \frac{1 - \frac{n}{\tau} \theta_k}{\theta_k^2} t_k^{(i)}$$

**end for**

**end for**

# Iteration complexity


## Theorem

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

# Iteration complexity

## Theorem

*randomised algorithm*


$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

# Iteration complexity

## Theorem

*randomised algorithm*

*# blocks*

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

# Iteration complexity

## Theorem

*randomised algorithm*

*# blocks*

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

*iteration counter*

# Iteration complexity

## Theorem

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

*randomised algorithm* (points to  $\mathbf{E}$ )  
*# blocks* (points to  $4n^2$ )  
*# processors* (points to  $\tau$ )  
*iteration counter* (points to  $k$ )

# Iteration complexity

## Theorem

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

*randomised algorithm* (points to  $\mathbf{E}$ )  
*# blocks* (points to  $4n^2$ )  
*depends of  $\tau$*  (points to  $\beta$ )  
*# processors* (points to  $\tau$ )  
*iteration counter* (points to  $k$ )

## Parallel coordinate descent

- We need more general overapproximations to compute the updates:
  - $f(x + te_i) \leq f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$   
 $\hookrightarrow$  not valid with several coordinates
  - $f(x + h) \leq f(x) + \langle \nabla f(x), h \rangle + \frac{L(\nabla f)}{2} \|h\|^2$   
 $\hookrightarrow$  too weak
- [Richtárik, Takáč, 2012]  
 Theory for **partially separable** functions

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

$f_j$  depends on  $i \in C_j$  only

$\omega_j = |C_j| \leq \omega$  for all  $j$



## Separable overapproximation

Proposition [Richtárik, Takáč]

If  $f$  is **partially separable**, then for all  $S \subseteq \{1, \dots, n\}$  and  $x, h \in \mathbb{R}^N$

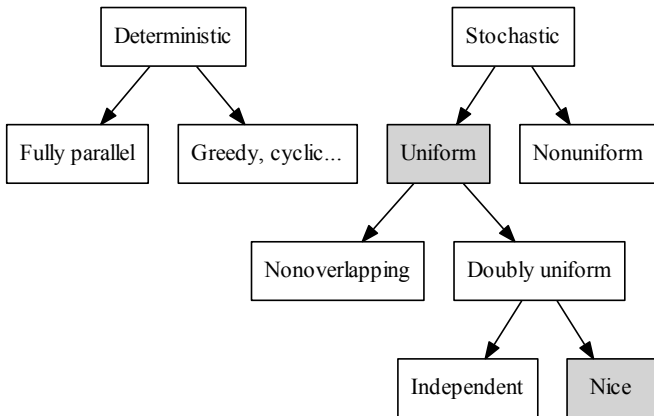
$$f(x + h_{[S]}) \leq f(x) + \langle \nabla f(x), h \rangle + \frac{\omega_S}{2} \|h\|_L^2$$

where

$$h_{[S]}^i = \begin{cases} h^i & i \in S \\ 0 & i \notin S \end{cases}$$

$$\omega_S = \min(|S|, \omega)$$

# Sampling



Uniform sampling  
 $\mathbf{P}(i \in \hat{S}) = \text{constant}$

$\tau$ -nice sampling  
 if  $|S| = \tau$ ,  
 $\mathbf{P}(\hat{S} = S) = \frac{1}{\binom{n}{\tau}}$

# Expected Separable Overapproximation (ESO)

Proposition [Richtárik, Takáč]

If  $\hat{S}$  is a  $\tau$ -nice sampling, then for all  $x, h \in \mathbb{R}^N$

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

where

$$\beta = 1 + \frac{(\omega - 1)(\tau - 1)}{\max\{1, n - 1\}}$$

Example:  $n = 10^6, \omega = 10^4, \tau = 10^3$

ESO:  $\beta \approx 11 \Rightarrow \frac{\tau}{\sqrt{\beta}} \approx 300$

whereas  $\omega_S = \min(\omega, \tau) = \tau \Rightarrow \frac{\tau}{\sqrt{\omega_S}} \approx 30$

## Are the iterates feasible?

Assume  $\psi = I_C$  where  $C$  is a convex set

$$z_{k+1}^{(i)} = \arg \min_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \nu_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

so  $z_{k+1} \in C$ , for all  $k$ .

$$x_{k+1} = y_k + \frac{n}{\tau} \theta_k (z_{k+1} - z_k) = (1 - \theta_k) x_k + \frac{n}{\tau} \theta_k z_{k+1} + \theta_k \left(1 - \frac{n}{\tau}\right) z_k$$

$\theta_k \left(1 - \frac{n}{\tau}\right) \leq 0$ : not a convex combination...

# Feasibility

## Lemma

Let  $\{x_k, z_k\}_{k \geq 0}$  be the iterates of APPROX

Then for all  $k \geq 0$ ,

$$x_k = \sum_{l=0}^k \gamma_k^l z_l$$

where  $\gamma_k^0, \gamma_k^1, \dots, \gamma_k^k \geq 0$  et  $\sum_{l=0}^k \gamma_k^l = 1$

$\gamma_k^l$  is defined recursively

# Supermartingale inequality

$$\hat{F}_k = f(x_k) + \sum_{l=0}^k \gamma_k^l \psi(z_l) \geq F(x_k)$$

$$\begin{aligned} \mathbf{E} \left[ \frac{1 - \theta_{k+1}}{\theta_{k+1}^2} (\hat{F}_{k+1} - F(x_*)) + \frac{\beta n^2}{2\tau^2} \|x_* - z_{k+1}\|_L^2 \mid \mathcal{S}_{k+1} \right] \\ \leq \frac{1 - \theta_k}{\theta_k^2} (\hat{F}_k - F(x_*)) + \frac{\beta n^2}{2\tau^2} \|x_* - z_k\|_L^2 \end{aligned}$$

# Iteration complexity

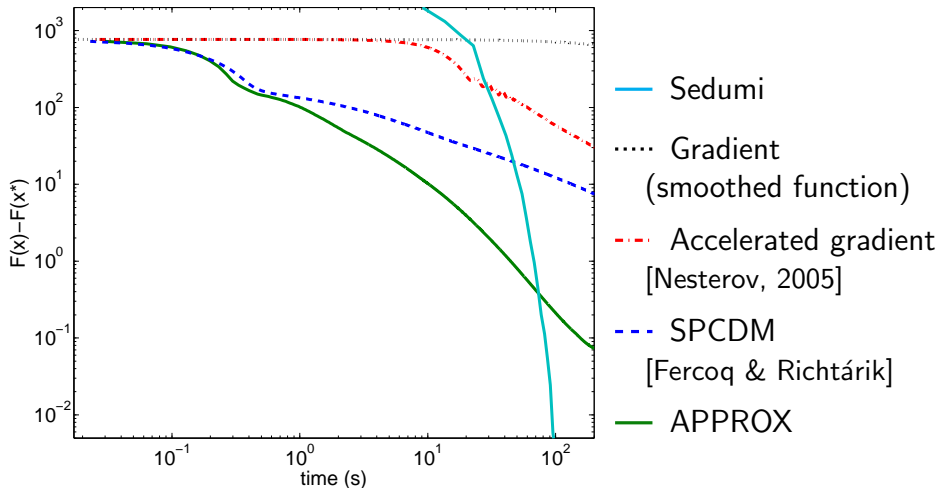
## Theorem

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2} \left( F(x_0) - F(x_*) + \frac{\beta}{2} \|x_0 - x_*\|_L^2 \right)$$

*randomised algorithm* (points to  $\mathbf{E}$ )  
*# blocks* (points to  $4n^2$ )  
*depends of  $\tau$*  (points to  $\beta$ )  
*# processors* (points to  $\tau$ )  
*iteration counter* (points to  $k$ )

# $L_1$ Regression

Dorothea :  $m=800$ ,  $N=100\ 000$ ,  $\omega=6\ 061$ ,  $\epsilon = 0,1$ ,  $\tau=4$

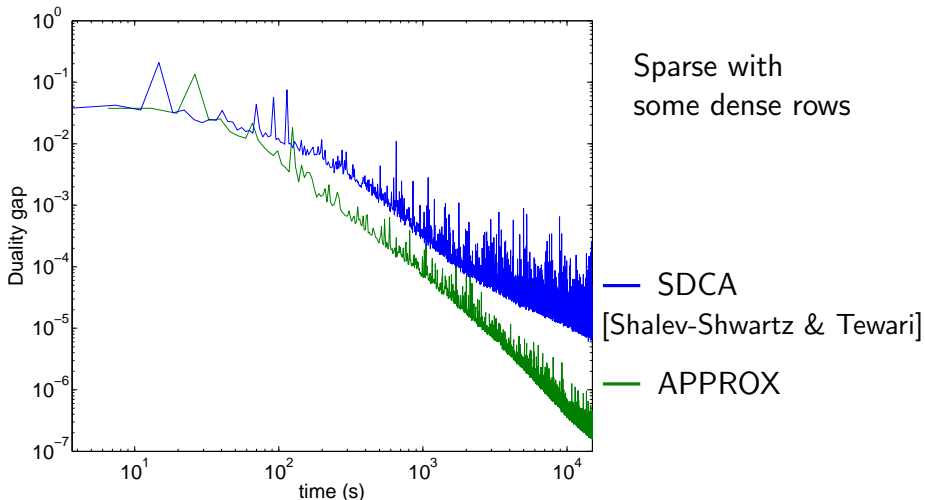


Comparison of Algorithms for  $F(x) = \|Ax - b\|_1 + \|x\|_1$



# SVM Dual

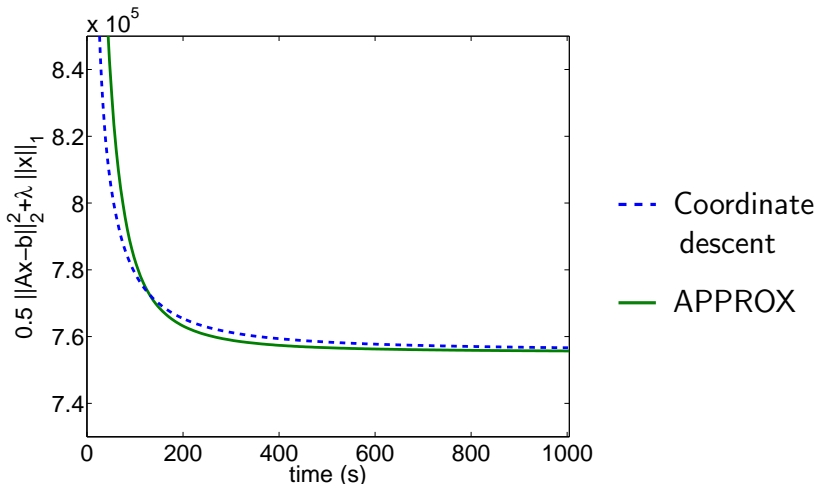
Malicious URL:  $m = 2,396,130$ ,  $N = 3,231,961$



Acceleration of Stochastic Dual Coordinate Ascent (SDCA)

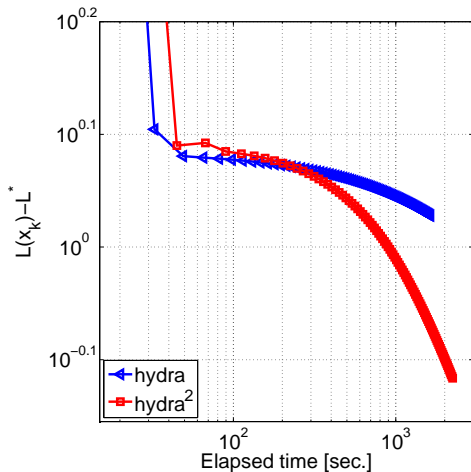
# Lasso

KDDB:  $m = 29,890,095$ ,  $N = 19,264,097$ ,  $\omega = 75$



Quick decrease in the beginning :  $8.3 \cdot 10^8 \rightarrow 8.5 \cdot 10^5$

# Large scale Lasso



Synthetic problem:

5TB

$m = 5e6$

$N = 5e10$

$\bar{\omega} = 6e5$

Resolution:

256 × 24 processors of  
Archer supercomputer

Sampling designed for  
distributed computation

[Fercoq, Qu, Richtárik, Takáč, Fast Distributed Coordinate Descent  
for Non-Strongly Convex Losses]

# Conclusion

- Summary
  - 1<sup>st</sup> accelerated, parallel and proximal coordinate descent method
  - Applications to machine learning
  - Non-uniform probabilities
- Perspectives
  - Primal-dual algorithm
  - Relax the i.i.d. sampling assumption