Random coordinate descent algorithms for huge-scale optimization problems

Ion Necoara

Automatic Control and Systems Engineering Depart.

University Politehnica Bucharest

Acknowledgement

Collaboration with

- Y. Nesterov, F. Glineur (Univ. Catholique Louvain)
- A. Patrascu, D. Clipici (Univ. Politehnica Bucharest)

Papers can be found at:

⇒ www.acse.pub.ro/person/ion-necoara

⇒ www.optimization-online.org

Outline

- Motivation
- Problem formulation
- Previous work
- Random coordinate descent alg. for smooth convex problems
- Random coordinate descent alg. for composite convex problems
- Random coordinate descent alg. for composite nonconvex problems
- Conclusions

Motivation

Recent “Big Data” applications:

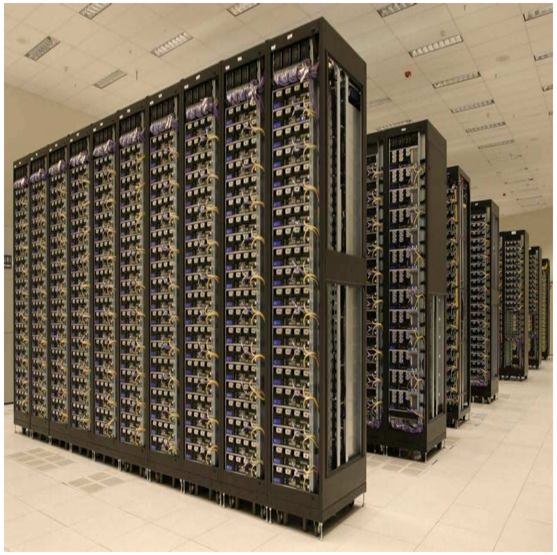
(a) **internet (e.g. PageRank)** (b) **support vector machine**
(c) **truss topology design** (d) **distributed control**
...gave birth to many huge-scale optimization problems (dimension of variables $n \approx 10^6 - 10^9$)

THE AVERAGE PERSON TODAY PROCESSES MORE DATA IN A SINGLE DAY THAN A PERSON IN THE 1500’S DID IN AN ENTIRE LIFETIME

LOOK TO THE LEFT, and you see Times Square at dusk. Look to the right, and you see the same location at midmorning. Internationally acclaimed photographer Stephen Wilkes’s time-altering image of New York’s Times Square is part of his body of work titled *Day to Night*.

The image was created by blending more than 1,400 separate photos taken over the course of 15 hours—a meticulous process that took him nearly three months.

PHOTO: STEPHEN WILKES



BUT matrices defining the optimization problem are very sparse!

Motivation

PageRank problem (Google ranking, network control, data analysis)

- Let $E \in \mathbb{R}^{n \times n}$ be adjacency matrix (column stochastic, sparse matrix)
- Find maximal unitary eigenvector satisfying $Ex = x$
- Number of variables (pages) $n \approx 10^6 - 10^9$

✓ Standard technique: power method ⇒ calculations of PageRank on supercomputers take about one week!

✓ Formulation as an optimization problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ex - x\|^2$$
$$\text{s.t. } e^T x = 1, \quad x \geq 0.$$

$\Rightarrow E$ has at most $p << n$ nonzeros on each row

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Z^T Z x + q^T x$$
$$\text{s.t. } a^T x = b, \quad l \leq x \leq u \qquad (\Rightarrow Z \text{ sparse!})$$

Motivation

Linear SVM problem

- Let $z_i \in \mathbb{R}^m \;\; i = 1, \dots, n$ be a set of training data points, $m << n$
- Two classes of data points z_i
- Find hyperplane $a^T y = b$ which separates data points z_i in two classes

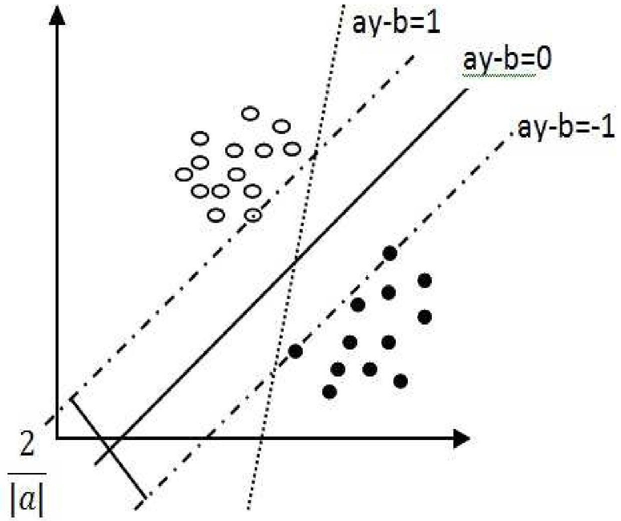
✓ Formulation as optimization problem:

$$\min_{a \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{2}\|a\|^2 + Ce^T \xi$$

s.t. $\alpha_i(a^T z_i - b) \geq 1 - \xi_i, \; \xi_i \geq 0 \;\; \forall i = 1, \dots, n$

$\Rightarrow \alpha_i \in \{-1, \; 1\}$ the id (label) of the class corresponding to z_i

$\Rightarrow n$ very big $\approx 10^6 - 10^9$ (many constraints)

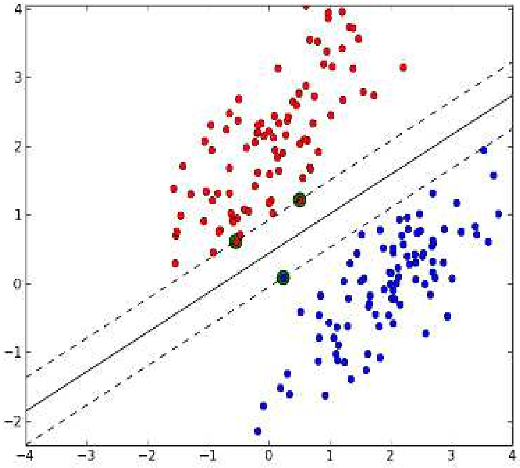


Motivation

The *dual* formulation for linear SVM:

$$\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^T(Z^TZ)x - e^Tx$$

$$\text{s.t.} \ \ \alpha^T x = 0, \quad 0 \le x \le Ce.$$



$\Rightarrow Z \in \mathbb{R}^{m\times n}, m << n$ depends on training points z_i (columns of Z are $\alpha_i z_i$)

or

$\Rightarrow Z \in \mathbb{R}^{n\times n}$ with sparse columns

Primal solution is recovered via: $a = \sum_i \alpha_i x_i z_i$ & $b = \sum_i (a^T z_i - \alpha_i)/n$

$$\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^TZ^TZx + q^Tx$$

$$\text{s.t.} \ \ a^T x = b, \quad l \le x \le u \qquad (\Rightarrow Z \text{ sparse!})$$

Motivation

State-of-the-art:

1. Second-order algorithms (Newton method, Interior point method):
$\Rightarrow$ solve at least one linear system per iteration

	Second-order methods
Complexity per iteration	$\approx \mathcal{O}(n^3)$
Worst-case no. of iterations	$\mathcal{O}(\ln\ln\frac{1}{\epsilon})/\mathcal{O}(\ln\frac{1}{\epsilon})$

where ϵ is the desired accuracy for solving the optimization problem

✓ Let $n = 10^8$, a standard computer with 2GHz processor takes:

10^7 years to finish only 1 iteration!

Motivation

2. First-order algorithms (Gradient method, Fast-Gradient method) perform at least one matrix-vector multiplication per iteration (in quadratic case)

	First-order methods
Complexity per iteration	$\approx \mathcal{O}(n^2)$
Worst-case no. of iterations	$\mathcal{O}(\frac{1}{\epsilon})/\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$

For $n = 10^8$, a standard computer with 2GHz processor takes 23.14 hours per iteration and 100 days to attain $\epsilon = 0.01$ accuracy!

Conclusion: for $n \approx 10^6 - 10^9$ we require algorithms with low complexity per iteration $\mathcal{O}(n)$ or even $\mathcal{O}(1)$!

$$\Downarrow$$

Coordinate Descent Methods

Problem formulation

$$F^* = \min_{x \in \mathbb{R}^n} \; F(x) \quad (= f(x) + h(x))$$

s.t. $a^T x = b$ (or even $Ax = b$) $\Rightarrow$ coupling constraints

Define decompositions:

- $n = \sum_{i=1}^{N} n_i, I_n = [E_1 \dots E_N]$ with $n \approx 10^6 - 10^9$
- $x = \sum_{i=1}^{N} E_i x_i \in \mathbb{R}^N$ and $x_{ij} = E_i x_i + E_j x_j, \quad x_i \in \mathbb{R}^{n_i}$

(i) $A \in \mathbb{R}^{m \times n}, \; m << n$

(ii) f has block-component Lipschitz continuous gradient, i.e.

$$\|\nabla_i f(x + E_i s_i) - \nabla_i f(x)\| \leq L_i \|s_i\| \quad \forall x \in \mathbb{R}^n, s_i \in \mathbb{R}^{n_i}, \; i = 1, \dots, N$$

(iii) h nonsmooth, convex and componentwise separable, i.e.

$h(x) = \sum_{i=1}^{n} h_i(x_i)$ $\Rightarrow$ e.g. : $h = 0$ or $h = 1_{[l,u]}$ or $h = \mu \|x\|_1 \dots$

Previous work - Greedy algorithms

Tseng (2009) developed coordinate gradient descent methods with greedy strategy

$$\min_{x \in \mathbb{R}^n} \; F(x) \quad (= f(x) + h(x))$$

s.t. $a^T x = b$ (or $Ax = b$)

Let $\mathcal{J} \subseteq \{1, \dots, N\}$ be set of indices at current iteration x, then define direction:

$$d_H(x; \mathcal{J}) = \arg \min_{s \in \mathbb{R}^n} f(x) + \langle \nabla f(x), s \rangle + \frac{1}{2} \langle Hs, s \rangle + h(x+s) \\ \text{s.t. } \; a^T s = 0, \quad s_j = 0 \quad \forall j \notin \mathcal{J}, \qquad (1)$$

where $H \in \mathbb{R}^{n \times n}$ is a positive definite matrix chosen at initial step of algorithm

Tseng & Yun, A Block-Coordinate Gradient Descent Method for Linearly Constrained Nonsmooth Separable Optimization, J. Opt. Theory Applications, 2009

Previous work - Greedy algorithms

Algorithm (CGD):

$1.$ Choose set of indices $\mathcal{J}^k \subset \{1, \dots, N\}$ w.r.t. Gauss-Southwell rule

$2.$ Solve (1) with $x = x^k,\ \mathcal{J} = \mathcal{J}^k,\ H = H_k$ to obtain $d^k = d_{H_k}(x^k; \mathcal{J}^k)$

$3.$ Choose stepsize $\alpha^k > 0$ and set $x^{k+1} = x^k + \alpha^k d^k$

Procedure of choosing $\mathcal{J}^k$ (Gauss-Southwell rule):

(i) decompose projected gradient direction d^k into low-dimensional vectors

(ii) evaluate function (1) in each low-dim. vector

(iii) choose the vector with smallest evaluation and assign to $\mathcal{J}$ its support

$\Rightarrow$ Alg. (CGD) takes $\mathcal{O}(n)$ operations per iteration (for quadratic case & $A = a$)

$\Rightarrow$ An estimate for rate of convergence of objective function values is:

$$\mathcal{O}\left(\frac{nL\|x^0 - x^*\|^2}{\epsilon}\right), \qquad L = \max_i L_i$$

Recently Beck (2012) developed a greedy coordinate descent algorithm (approx. same complexity) for singly linear constrained models with h box indicator function

Previous work - Random algorithms

Nesterov (2010) derived complexity estimates of *random* coordinate descent methods

$$\min_{x \in Q} f(x)$$

$\Rightarrow Q = Q_1 \times \cdots \times Q_N$ convex $\Rightarrow h(x) = 1_Q(x)$
$\Rightarrow f$ convex and block-component Lipschitz gradient
$\Rightarrow a = 0$ (no coupling constraints)

Algorithm (RCGD):

1. Choose randomly and index i_k with respect to given probability p_{i_k}
2. Set $x^{k+1} = x^k + E_{i_k} \nabla_{i_k} f(x_k)$.

$\Rightarrow$ We can choose Lipschitz dependent probabilities $p_i = L_i / \sum_{i=1}^{N} L_i$

$\Rightarrow$ For structured cases (sparse matrices with $p << n$ number of nonzeros per row) has complexity per iteration $\mathcal{O}(p)$!

Previous work - Random algorithms

✓ An estimate for rate of convergence for the expected values of objective function for Nesterov's method (RCGD) is

$$\mathcal{O}\left(\frac{\sum_{i=1}^{N} L_i \|x^0 - x^*\|^2}{\epsilon}\right)$$

✓ Richtarik (2012), Lu (2012) extended complexity estimates of Nesterov's random coordinate descent method to composite case

$$\min_{x \in \mathbb{R}^n} \; F(x) \quad (= f(x) + h(x))$$

$\Rightarrow$ f convex and has block-component Lipschitz gradient
$\Rightarrow$ h nonsmooth, convex, block-separable

$$\Downarrow$$

parallel implementations & inexact implementations were also analyzed

Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM J. Opt., 2012

Random coordinate descent - smooth & constrained case

$$\min_{x\in\mathbb{R}^n} \ f(x)$$

s.t. $a^T x = b$ (or $Ax = b$)

$\Rightarrow$ f convex & has block-component Lipschitz gradient

$\Rightarrow$ communication via connected graph $G = (V, E)$

```
Algorithm (RCD) :  given $x^0$,  $a^T x^0 = b$
1. Choose randomly a pair $(i_k, j_k) \in E$ with probability $p_{i_k j_k}$
2. Set $x^{k+1} = x^k + E_{i_k} d_{i_k} + E_{j_k} d_{j_k}$,
```

$$d_{ij} = (d_i, d_j) = \arg \min_{s_{ij}\in\mathbb{R}^{n_i+n_j}} f(x) + \langle \nabla_{ij} f(x), s_{ij}\rangle + \frac{L_i + L_j}{2}\|s_{ij}\|^2$$

$$\text{s.t.} \ a_i^T s_i + a_j^T s_j = 0$$

each iteration requires approximately $\mathcal{O}(p)$ operations (quadratic case)!

✓ *Necoara, Nesterov & Glineur, A random coordinate descent method on large optimization problems with linear constraints, ICCOPT, 2013*

✓ *Necoara, Random coordinate descent algorithms for multi-agent convex optimization over networks, IEEE Trans. Automatic Control, 2013*

(RCD) smooth case - convergence rate

Characteristics:

- only 2 components (in E) of x are updated per iteration (distributed!)
- alg. (RCD) needs only 2 components of gradient $\Rightarrow$ complexity per iteration $\mathcal{O}(p)$!
- closed-form solution $\Rightarrow$ e.g. $d_i = -\frac{1}{L_i+L_j}\left(\nabla_i f(x) - \frac{a_{ij}^T \nabla_{ij} f(x)}{a_{ij}^T a_{ij}} a_i\right)$

Theorem 1 *Let x^k generated by Algorithm (RCD). Then, the following estimates for expected values of objective function can be obtained*

$$\mathcal{E}[f(x^k)] - f^* \leq \frac{\|x^0 - x^*\|^2}{\lambda_2(Q)k}$$

If additionaly, function f is σ-strongly convex, then

$$\mathcal{E}[f(x^k)] - f^* \leq \Big(1 - \lambda_2(Q)\sigma\Big)^k (f(x^0) - f^*)$$

where $Q = \sum_{(i,j)\in E} \frac{p_{ij}}{L_i+L_j}\left(I_{n_i+n_j} - \frac{a_{ij}a_{ij}^T}{a_{ij}^T a_{ij}}\right)$ *(Laplacian matrix of the graph)*

Selection of probabilities

I. uniform probabilities:

$$p_{ij} = \frac{1}{|E|}$$

II. probabilities dependent on the Lipschitz constants L_i

$$p_{ij}^{\alpha} = \frac{L_{ij}^{\alpha}}{L^{\alpha}}, \qquad \text{where } L^{\alpha} = \sum_{(i,j)\in E} L_{ij}^{\alpha}, \;\; \alpha \geq 0.$$

III. optimal probabilities obtained from $\max_{Q\in\mathcal{M}} \lambda_2(Q) \Leftrightarrow$ SDP

$$[p_{ij}^{*}]_{(i,j)\in E} = \arg\max_{t,\, Q} \left\{ t: \quad Q + t\frac{aa^T}{a^T a} \succeq tI_n, \;\; Q \in \mathcal{M} \right\}.$$

$$\mathcal{M} = \{Q \in \mathbb{R}^{n\times n} : Q = \sum_{(i,j)\in E} \frac{p_{ij}}{L_{ij}} Q_{ij}, \; p_{ij} = p_{ji}, \; p_{ij} = 0 \text{ if } (i,j) \notin E, \; \sum_{(i,j)\in E} p_{ij} = 1\}.$$

Comparison with full projected gradient alg.

Assume:

$$a = e \text{ and Lipschitz dependent probabilities } p_{ij}^1 = \frac{L_i + L_j}{L^1}$$

then

$$Q = \frac{1}{\sum_{i=1}^n L_i} \left(I_n - \frac{1}{n} e e^T \right) \Rightarrow \lambda_2(Q) = \frac{1}{\sum_{i=1}^n L_i}$$

Alg. (RCD)	Alg. full projected grad.
iter. complexity $\mathcal{O}(p)$	iter. complexity $\mathcal{O}(n \cdot p)$
$\mathcal{E}[f(x^k)] - f^* \leq \frac{\sum_i L_i \|x^0 - x^*\|^2}{k}$	$f(x^k) - f^* \leq \frac{L_f \|x^0 - x^*\|^2}{k}$
	$\nabla^2 f(x) \leq L_f \cdot I_n$

Remark: maximal eigenvalue of a symmetric matrix can reach its trace!

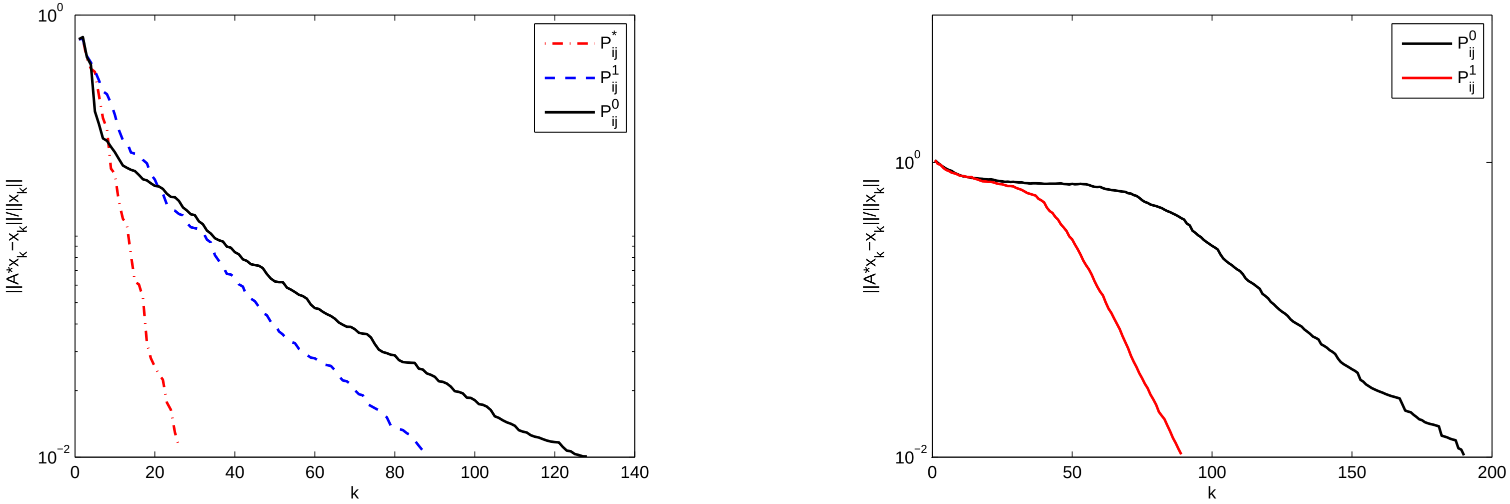
worst case: rate of convergence of (RCD) met. is the same as of full gradient met.!

However:

- (RCD) method has cheap iteration
- (RCD) method has more chances to accelerate

Numerical tests (I) - Google problem

- Google problem: $\min_{e^T x = 1} \|Ex - x\|^2$
- accuracy $\epsilon = 10^{-3}$, full iterations: $x^0, x^{\frac{n}{2}}, x^n, \cdots, x^{\frac{kn}{2}} \cdots$



Equivalent number of full iterations versus $\|Ex^k - x^k\|/\|x^k\|$

Left: $n = 30$ using probabilities p^0_{ij}, p^1_{ij} and p^*_{ij}

Right: $n = 10^6$ using probabilities p^0_{ij} and p^1_{ij}

Random coordinate descent - composite case

$$\min_{x \in \mathbb{R}^n} f(x) + h(x)$$

s.t. $a^T x = b$

$\Rightarrow$ f convex with block-component Lipsch. gradient

$\Rightarrow$ h convex, nonsmooth and separable: $h(x) = \sum_{i=1}^{n} h_i(x_i)$

(e.g. $h = 1_{[l,u]}$ or $h = \mu \|x\|_1$...)

Algorithm (CRCD): $a^T x^0 = b$

1. Choose randomly a pair (i_k, j_k) with probability $p_{i_k j_k}$
2. Set $x^{k+1} = x^k + E_{i_k} d_{i_k} + E_{j_k} d_{j_k}$,

$$d_{ij} = (d_i, d_j) = \arg \min_{s_{ij} \in \mathbb{R}^{n_i + n_j}} f(x) + \langle \nabla_{ij} f(x), s_{ij} \rangle + \frac{L_i + L_j}{2} \|s_{ij}\|^2 + h(x + s_{ij})$$

$$\text{s.t. } a_i^T s_i + a_j^T s_j = 0$$

each iteration requires approximately $\mathcal{O}(p)$ operations (quadratic case)!

Random coordinate descent - composite case

Characteristics:

- only 2 components of x are updated per iteration
- alg. (CRCD) needs only 2 components of the gradient and is using only 2 functions h_i & h_j of h
- if $N = n$ and h is given by ℓ_1 norm or indicator function for box, then the direction d_{ij} can be computed in closed form
- if $N < n$ and h is coordinatewise separable, strictly convex and piece-wise linear/quadratic with $\mathcal{O}(1)$ pieces (e.g. h given by ℓ_1 norm), then the direction d_{ij} can be computed in linear-time (i.e. $\mathcal{O}(n_i + n_j)$ operations).
- the complexity of choosing randomly a pair (i, j) with a uniform probability distribution requires $\mathcal{O}(1)$ operations

(CRCD) composite case - convergence rate

Theorem 2 *Let x^k be generated by Algorithm (CRCD) and $L = \max_i L_i$. If the index pairs are selected with uniform distribution, then we have*

$$\mathcal{E}[F(x^k)] - F^* \leq \frac{N^2 L \|x^0 - x^*\|^2}{k}.$$

If additionaly, function f is σ-strongly convex, then

$$\mathcal{E}[F(x^k)] - F^* \leq \left(1 - \frac{2(1-\gamma)}{N^2}\right)^k (F(x^0) - F^*),$$

where γ is defined by:

$$\gamma = \begin{cases} 1 - \frac{\sigma}{8L}, & if \ \sigma \leq 4L \\ \frac{2L}{\sigma}, & otherwise. \end{cases}$$

Necoara & Patrascu, Random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints, Computational Opt. Appl., 2013

Arithmetic complexity - comparison

$\Rightarrow N = n$ (scalar case) & sparse QP

$\Rightarrow R^2 = \|x^* - x^0\|^2$

$\Rightarrow L_f \leq \sum_i L_i$ & $L = \max_i L_i$ & $L_{av} = \frac{\sum_i L_i}{n}$

metoda	grad. Lipsc.	model	complexity per iteration
GM (Nesterov)	$\mathcal{O}(\frac{L_f R^2}{\epsilon})$	h & a	full gradient - $\mathcal{O}(n)$
CGM (Tseng)	$\mathcal{O}(\frac{nLR^2}{\epsilon})$	h & a	partial gradient - $\mathcal{O}(n)$
RCGM (Nesterov)	$\mathcal{O}(\frac{nL_{av}R^2}{\epsilon})$	h & $a = 0$	partial gradient - $\mathcal{O}(1)$
RCD	$\mathcal{O}(\frac{nL_{av}R^2}{\epsilon})$	$h = 0$ & a	partial gradient - $\mathcal{O}(1)$
CRCD	$\mathcal{O}(\frac{n^2L_{av}R^2}{\epsilon})$	h & a	partial gradient - $\mathcal{O}(1)$

- our methods RCD & CRCD have usually better ($N < n$) or comparable ($N = n$) arithmetic complexity than (or with) existing methods
- adequate for parallel or distributed architectures
- robust and have more chances to accelerate (due to randomness)
- easy to implement (closed-form solution)

Numerical tests (II) - SVM problem

Data set	n/m	(CRCD) full-iter/obj/time(s)	(CGD) iter/obj/time(s)
a7a	16100/122 $(p = 14)$	11242/-5698.02/2.5	23800/-5698.25/21.5
a9a	32561/123 $(p = 14)$	15355/-11431.47/7.01	45000/-11431.58/89.0
w8a	49749/300 $(p = 12)$	15380/-1486.3/26.3	19421/-1486.3/27.2
ijcnn1	49990/22 $(p = 13)$	7601/-8589.05/6.01	9000/-8589.52/16.5
web	350000/254 $(p = 85)$	1428/-69471.21/29.95	13600/-27200.68/748
covtyp	581012/54 $(p = 12)$	1722/-337798.34/38.5	12000/-24000/480
test1	$2.2 \cdot 10^6/10^6$ $(p = 50)$	228/-1654.72/51	4600/-473.93/568
test2	$10^7/10^3$ $(p = 10)$	500/-508.06/142.65	502/-507.59/516,66

real test problems taken from LIBSVM library

Our alg. (CRCD) - by a factor of 10 faster than (CGD) method (Tseng)!

Numerical tests (III) - Chebyshev center problem

Chebyshev center problem: given a set of points $z^1, \dots, z^n \in \mathbb{R}^m$, find the center z_c and radius r of the smallest enclosing ball of the given points

Applications: pattern recognition, protein analysis, mechanical engineering

Formulation as an optimization problem:

$$\begin{aligned} &\min_{r, z_c} \; r \\ &\text{s.t.:} \quad \|z^i - z_c\|^2 \leq r \quad \forall i = 1, \dots, n, \end{aligned}$$

where r is the radius and z_c is the center of the enclosing ball.

Dual problem:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \|Zx\|^2 - \sum_{i=1}^{n} \|z^i\|^2 x_i + \mathbf{1}_{[0,\infty)}(x) \qquad (2) \\ &\text{s.t.} \quad e^T x = 1, \end{aligned}$$

where Z contains the given points z^i as columns

Numerical tests (III) - Chebyshev center problem

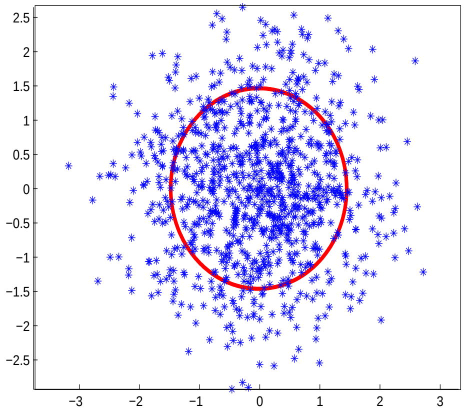
Simple recovery of primal optimal solution from dual x^*:

$$r* = \left(-\|Zx^*\|^2 + \sum_{i=1}^{n} \|z^i\|^2 x_i^* \right)^{1/2}, \qquad z_c^* = Zx^*. \tag{3}$$

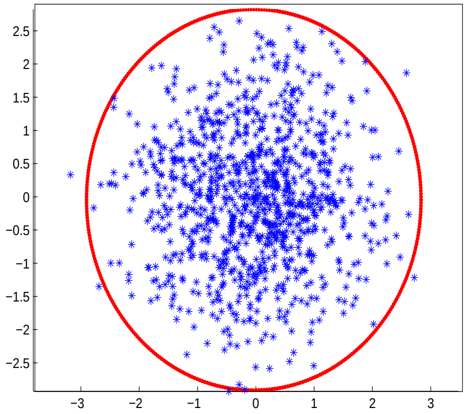
Two sets of numerical experiments:

- all alg. start from $x^0 = e_1$: observe that Tseng's algorithm has good performance and Gradient Method is worst
- starting from $x^0 = e/n$: observe that Gradient Method has good performance and Tseng is worst
- algorithm (CRCD) is very robust w.r.t. starting point x^0

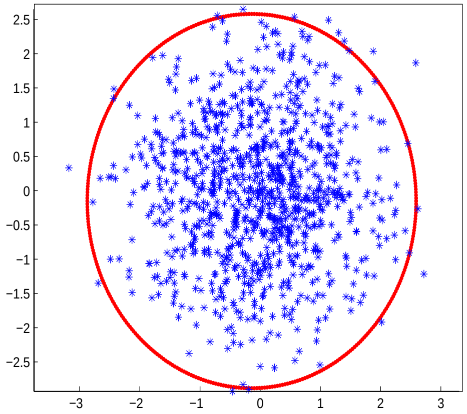
Numerical tests (III) - Chebyshev center problem



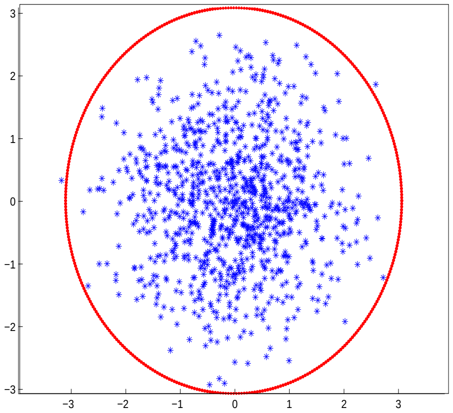
$x^0 = e/n$ (CGD)



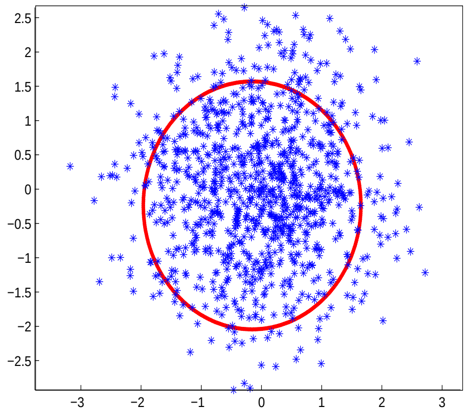
$x^0 = e/n$ (GM)



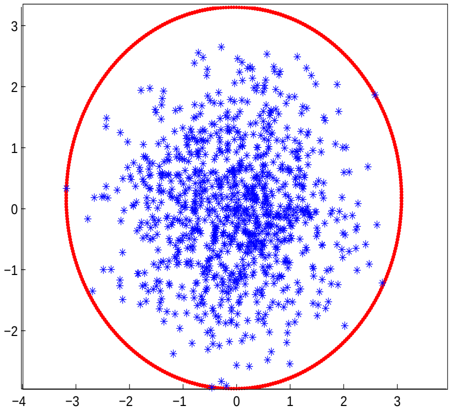
$x^0 = e/n$ (CRCD)



$x^0 = e_1$



$x^0 = e_1$



$x^0 = e_1$

Random coordinate descent - nonconvex & composite

$$\min_{x\in\mathbb{R}^n} \; f(x) + h(x)$$

$\Rightarrow$ f nonconvex with block-component Lip. gradient & $a = 0$ (no coupling constraints)
$\Rightarrow$ h is proper, convex and block separable $\Rightarrow$ e.g. : $h = 0$ or $h = 1_{[l,u]}$ or $h = \mu\|x\|_1$...

If $x^* \in \mathbb{R}^n$ is a local minimum, then the following relation holds

$$0 \in \nabla f(x^*) + \partial h(x^*) \qquad \text{(stationary points)}$$

Algorithm (NRCD):

1. Choose randomly an index i_k with probability p_{i_k}

2. Compute $x^{k+1} = x^k + E_{i_k} d_{i_k}$

$$d_i = \arg\min_{s_i\in\mathbb{R}^{n_i}} f(x) + \langle\nabla_i f(x), s_i\rangle + \frac{L_i}{2}\|s_i\|^2 + h(x+s_i).$$

Each iteration is cheap, complexity $\mathcal{O}(p)$, where $p << n$ (even closed-form solution)!

Patrascu & Necoara, Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization, submitted J. Global Opt., 2013

(NRCD) nonconvex & composite - convergence rate

We introduce the following map (here $L = [L_1 \dots L_N]$ and $D_L = \mathrm{diag}(L)$):

$$d_L(x) = \arg\min_{s \in \mathbb{R}^n} f(x) + \langle \nabla f(x), s\rangle + \frac{1}{2}\|s\|_L^2 + h(x+s)$$

We define *optimality measure*: $M_1(x, L) = \|D_L \cdot d_L(x)\|_L^*$, where $\|s\|_L^2 = s^T D_L s$ and $\|u\|_L^*$ its dual norm (observe that $M_1(x, L) = 0 \iff x$ stationary point)

Theorem 3 *Let the sequence x^k be generated by Algorithm (NRCD) using the uniform distribution, then the following statements are valid:*

- *(i) The sequence of random variables $M_1(x^k, L)$ converges to 0 a.s. and the sequence $F(x^k)$ converges to a random variable $\bar{F}$ a.s.*
- *(ii) Any accumulation point of the sequence x^k is a stationary point*

Moreover, in expectation

$$\min_{0 \le l \le k} \mathcal{E}\left[\left(M_1(x^l, L)\right)^2\right] \le \frac{2N\left(F(x^0) - F^*\right)}{k} \qquad \forall k \ge 0$$

Random coordinate descent - nonconvex & constrained

$$\min_{x\in\mathbb{R}^n} f(x) + h(x)$$

s.t. $a^T x = b$,

$\Rightarrow$ f nonconvex with block-component Lip. gradient

$\Rightarrow$ h is proper, convex and separable

If x^* is a local minimum, then there exists a scalar λ^* such that:

$$0 \in \nabla f(x^*) + \partial h(x^*) + \lambda^* a \quad \text{and} \quad a^T x^* = b.$$

Algorithm (NCRCD):

1. Choose randomly a pair (i_k, j_k) with probability $p_{i_k j_k}$
2. Compute $x^{k+1} = x^k + E_{i_k} d_{i_k} + E_{j_k} d_{j_k}$,

$$d_{ij} = (d_i, d_j) = \arg \min_{s_{ij}\in\mathbb{R}^{n_i+n_j}} f(x) + \langle \nabla_{ij} f(x), s_{ij}\rangle + \frac{L_i + L_j}{2}\|s_{ij}\|^2 + h(x+s_{ij})$$

$$\text{s.t. } a_i^T s_i + a_j^T s_j = 0$$

Each iteration is cheap, complexity $\mathcal{O}(p)$ (even closed-form solution)!

(NCRCD) nonconvex & constrained - convergence rate

We introduce the following map:

$$d_{\bar{T}}(x) = \arg \min_{s \in \mathbb{R}^n:\, a^T s = 0} f(x) + \langle \nabla f(x), s \rangle + \frac{1}{2} \|s\|_{\bar{T}}^2 + h(x+s).$$

We define the *optimality measure*: $M_2(x,T) = \|D_T \cdot d_{NT}(x)\|_T^*$, where $T_i = \frac{1}{N}\sum_j L_{ij}$ (observe that $M_2(x,T) = 0 \iff x$ stationary point)

Theorem 4 *Let the sequence x^k be generated by Algorithm (NCRCD) using the uniform distribution, then the following statements are valid:*

- *(i) The sequence of random variables $M_2(x^k, T)$ converges to 0 a.s. and the sequence $F(x^k)$ converges to a random variable $\bar{F}$ a.s.*
- *(ii) Any accumulation point of the sequence x^k is a stationary point*

Moreover, in expectation

$$\min_{0 \le l \le k} \mathcal{E}\left[\Big(M_2(x^l, T)\Big)^2\right] \le \frac{N\left(F(x^0) - F^*\right)}{k} \quad \forall k \ge 0.$$

Numerical tests (IV) - eigenvalue complementarity problem

Eigenvalue complem. prob. (EiCP): given matrices $A, B \in \mathbb{R}^{n\times n}$, find $\lambda \in \mathbb{R}$ & $x \neq 0$

$$\begin{cases} w = (\lambda B - A)x, \\ w \geq 0,\ x \geq 0,\ w^T x = 0 \end{cases}$$

Applications of EiCP: optimal control, stability analysis of dynamic systems, electrical networks, quantum chemistry, chemical reactions, economics...

If A, B are symmetric, then we have *symmetric (EiCP)*. Symmetric (EiCP) is equivalent with finding a stationary point of a *generalized Rayleigh quotient* on the simplex:

$$\min_{x\in\mathbb{R}^n} \frac{x^T A x}{x^T B x} \quad \text{s.t.: } e^T x = 1,\ x \geq 0.$$

Equivalent *nonconvex logarithmic* formulation (for $A, B \geq 0$, with $a_{ii}, b_{ii} > 0 \Rightarrow$ e.g. stability of positive dynamical systems):

$$\max_{x\in\mathbb{R}^n} f(x) \quad \left(= \ln x^T A x - \ln x^T B x\right)$$

$$\text{s.t.: } e^T x = 1,\ x \geq 0 \qquad \Rightarrow h(x) = 1_{[0,\infty)}(x)$$

$\Rightarrow$ Perron-Frobenius theory for A irreducible and $B = I_n$ implies global maximum!

Numerical tests (IV) - eigenvalue complementarity problem

$\Rightarrow$ Compare with DC (Difference of Convex functions) algorithm (Thi et al. 2012), equivalent in some sense with Projected Gradient method

$\Rightarrow$ Hard to estimate Lipschitz parameter μ in DC alg., but crucial for convergence of DC

$$\max_{x:\, e^T x = 1,\ x \geq 0} \left(\frac{\mu}{2} \|x^2\| + \ln x^T A x - \ln x^T B x \right) - \left(\frac{\mu}{2} \|x^2\| \right)$$

	DC				NCRCD		
n	μ	CPU (sec)	Iter	F^*	CPU (sec)	Iter	F^*
$7.5 \cdot 10^5$	$0.01n$	0.44	1	3.11	37.59	38	177.52
	n	0.81	2	143.31			
	$1.43n$	72.80	181	177.52			
	$50n$	135.35	323	177.54			
10^6	$0.01n$	0.67	1	3.60	49.67	42	230.09
	n	1.30	2	184.40			
	$1.43n$	196.38	293	230.09			
	$50n$	208.39	323	230.11			
10^7	$0.01n$	4.69	1	10.83	49.67	42	230.09
	n	22.31	2	218.88			
	$1.45n$	2947.93	325	272.37			
	$50n$	2929.74	323	272.38			

Conclusions

- usually *full* first/second-order methods are inefficient for huge-scale optimization
- for sparse problems coordinate descent methods are adequate for their low complexity per iteration ($\mathcal{O}(p)$)
- randomized coordinate descent methods have simple strategy for choosing the working set - $\mathcal{O}(1)$ operations for index choice
- usually randomized methods outperform greedy methods
- we provide rates of convergence and arithmetic complexities for randomized coordinate descent methods
- randomized methods are easy to implement and adequate for modern parallel and distributed architectures

References

- A. Beck, *The 2-coordinate descent method for solving double-sided simplex constrained minimization problems*, Technical Report, 2012.

- Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization 22(2), 341–362, 2012.

- P. Richtarik and M. Takac, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, Series A, DOI 10.1007/s10107-012-0614-z, 2012.

- P. Tseng and S. Yun, *A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization*, Mathematical Programming, 117, 387–423, 2009.

- P. Tseng and S. Yun, *A Block-Coordinate Gradient Descent Method for Linearly Constrained Nonsmooth Separable Optimization*, Journal of Optimization Theory and Applications, 140, 513–535, 2009.

References

- I. Necoara, Y. Nesterov and F. Glineur, *A random coordinate descent method on large optimization problems with linear constraints*, Technical Report, University Politehnica Bucharest, 2011, `http://acse.pub.ro/person/ion-necoara`.

- I. Necoara, *Random coordinate descent algorithms for multi-agent convex optimization over networks*, IEEE Transactions on Automatic Control, 58(7), 1-12, 2013.

- I. Necoara and A. Patrascu, *A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints*, Computational Optimization and Applications, in press, 2013, `http://acse.pub.ro/person/ion-necoara/`.

- A. Patrascu and I. Necoara, *Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization*, submitted to Journal of Global Optimization, 2013.