On Nesterov's Random Coordinate Descent Algorithms

Zheng Xu

University of Texas At Arlington

February 19, 2015

Unconstratined smooth minimization problem

Considering the following mathematical optimization problem

$$\min_{x \in \mathbb{R}^N} f(x) \tag{1}$$

where $f(x)$ is a convex and differentiable function on $\mathbb{R}^N$.

Full-Gradient Descent

When $f(x)$ is differentiable, it is favorable to apply first-order method like gradient descent method.
The basic step of the traditional full-gradient descent is

$$x^{k+1} = x^k + \alpha_k(-\nabla f(x^k)) \tag{2}$$

- $\{x^k\}$ is the optimization sequence generated by gradient descent algorithm
- $\alpha_k > 0$ is the step size. Usually given by some line search techniques.
- $\nabla f(x^k)$ is the gradient direction of $f(x)$ at x^k. Note the gradient direction at x^k is the steepest ascent direction at x^k. So $-\nabla f(x^k)$ is the steepest descent direction at x^k.

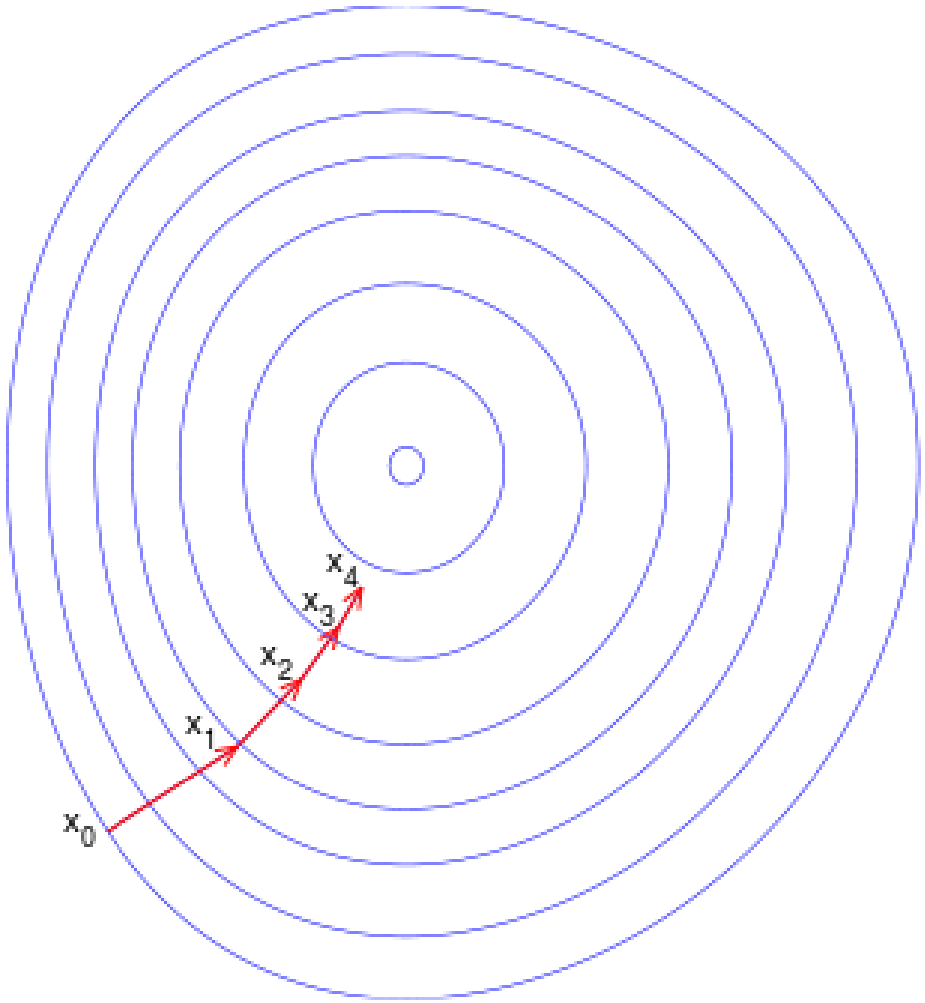

Figure: A basic example of $\mathbb{R}^2$

Huge-scale problem

- Full-gradient descent method works!
- But what if the N is incredibly large? e.g. $N = 1,000,000,000,000$
- Compute $\nabla f(x^k)$ will suffer from unfeasibly computational complexity.
- It is even impossible to store that large-scale data in a single computer!

For problem of this type, we adopt the term huge-scale problem.

Coordinate Descent

Recently, in [Ber99][Nes12][RT14], the old-age coordinate descent method is revisited to solve the huge-scale problems. The basic step of coordinate descent method is

$$x_{i_k}^{k+1} = x_{i_k}^{k} + \alpha_k(-\nabla_{i_k} f(x^k)) \tag{3}$$

where i_k is the component index of x to be updated in k-th step. The main idea of coordinate is updating only very few components of x in each iteration instead of full of x.
The benefit is that the iteration cost of coordinate descent is only $\mathcal{O}(1)$ or $\mathcal{O}(m)$ for $m \ll N$, while the full-gradient method usually requires at least $\mathcal{O}(N)$.

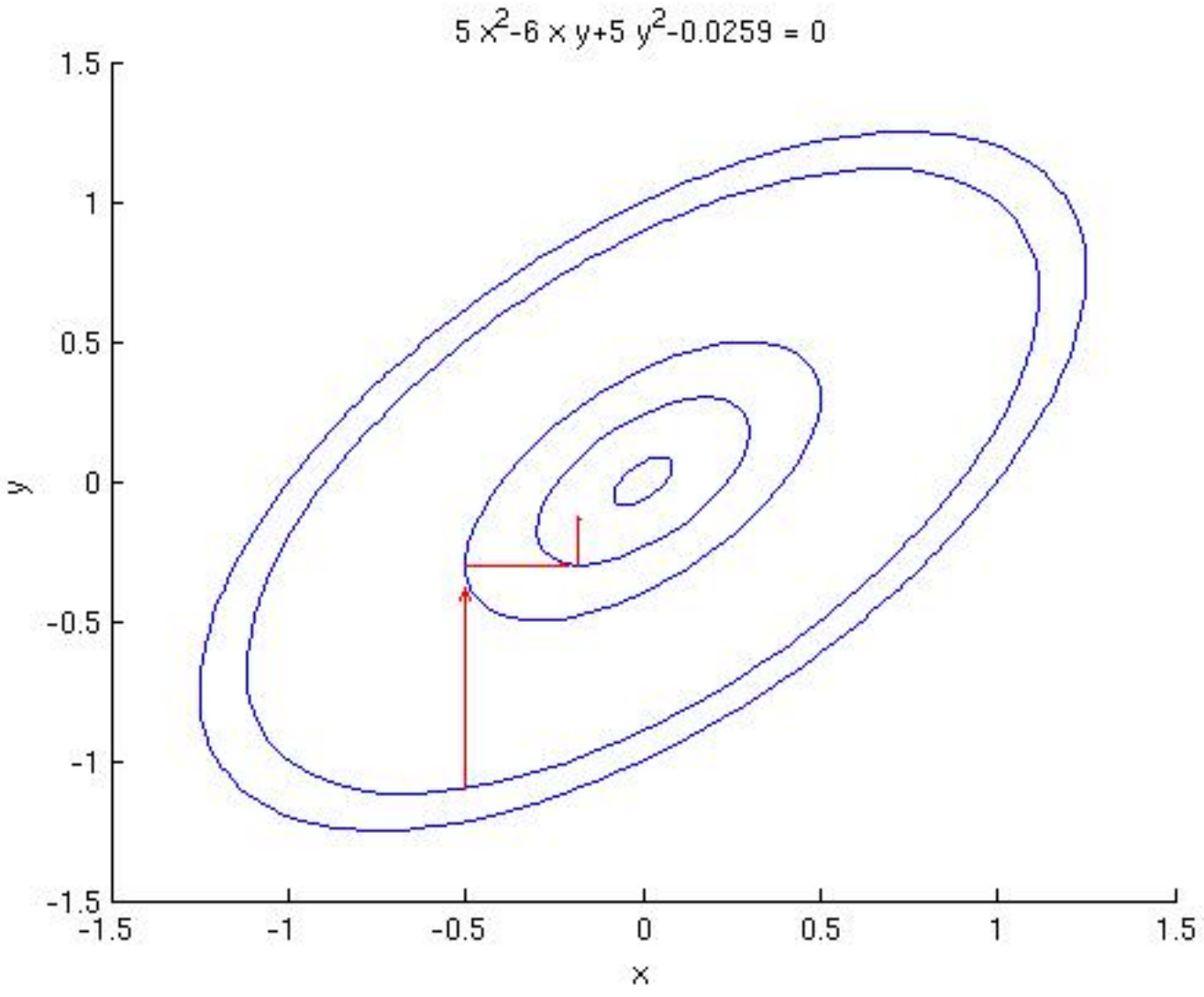

Figure: Example of Cyclic Coordinate Descent in $\mathbb{R}^2$

Does coordinate descent really work?

- Vague convergence theory: Theoretically, it's hard to prove the convergence of conventional cyclic coordinate search. It's even harder to estimate its convergence rate.
- Computational expensive line search: For coordinate descent methods, the line-search strategies based on the function values are too expensive.

To survive, in [Nes12], Yu. Nesterov propose a random coordinate descent with clear convergence theory and lower cost iteration.

Improved Assumption

In [Nes12], it is further assumed the objective function $f(x)$ is Lipschitz continuous for every *partial gradient*. i.e.

$$\|\nabla_i f(x + \Delta x) - \nabla_i f(x)\|_{(i)} \leq L_i \|\Delta x\| \tag{4}$$

Roughly speaking, it is assumed $f(x)$ is strongly smooth on every component of $x \in \mathbb{R}^N$

Basic Idea of Random Coordinate Descent

The basic idea of random coordinate descent is adopting the random index selection instead of cyclic index selection, which is

$$i_k = \mathcal{R}_\alpha \tag{5}$$

$$x_{i_k}^{k+1} = x_{i_k}^k + \alpha_k(-\nabla_{i_k} f(x^k)) \tag{6}$$

where $\mathcal{R}_\alpha$ is a random number generator that is subject to some distribution with parameter α.

The $\mathcal{R}_\alpha$

$\mathcal{R}_\alpha \in \mathbb{N}$ is a special random counter, generating integer number with probability

$$p_\alpha^{(i)} = L_i^\alpha \cdot \left[\sum_{j=1}^{n} L_j^\alpha\right]^{-1} \tag{7}$$

where L_i is the Lipschitz constant of the *partial gradient* $\nabla_i f(x)$. As a special case, when $\alpha = 0$, $\mathcal{R}_0$ is a uniform random number generator.

Does it converge?

Denote random variable $\xi_k = i_0, i_1, \ldots, i_k$, we have the following convergence theorem of random coordinate descent method with improved assumption.

Theorem

For any $k \geq 0$ we have

$$\mathbb{E}_{\xi_k} f(x^k) - f^* \leq \frac{2}{k+4} \left[\sum_{j=1}^{n} L_j^{\alpha} \right] \cdot R_{1-\alpha}^2(x_0) \tag{8}$$

where $R_\beta(x_0) = \max_x \{\max_{x_* \in X^*} \|x - x_*\|_\beta\}$

How fast does it converge?

Let first discuss when $\alpha = 0$, we now have

$$\mathbb{E}_{\xi_k} f(x^k) - f^* \leq \frac{2n}{k+4} R_1^2(x_0) \tag{9}$$

And the worst-case convergence rate of full-gradient descent method is

$$f(x^k) - f^* \leq \frac{n}{k} R_1^2 \tag{10}$$

So the convergence rate of random coordinate descent is proportional to that one of full-gradient method.

How fast does it converge? - Continued

Let first discuss when $\alpha = \frac{1}{2}$, denote

$$D_\infty(x_0) = \max_x \left\{ \max_{y \in X*} \max_{1 \le i \le N} |x^{(i)} - y^{(i)}| : f(x) < f(x_0) \right\} \quad (11)$$

Under this terminology system, we have

$$\mathbb{E}_{\xi_k} f(x^k) - f^* \le \frac{2}{k+4} \left[\sum_{j=1}^{n} L_j^{1/2} \right] \cdot \cdot D_\infty^2(x_0) \quad (12)$$

However, there exists some cases that the random coordinate descent method can work while the full-gradient method cannot in the sense that $\sum_{j=1}^{n} L_j^{1/2}$ can be bounded when $L = \max\{L_i\}$ cannot.

How fast does it converge? - Continued

Let first discuss when $\alpha = 1$, we now have

$$\mathbb{E}_{\xi_k} f(x^k) - f^* \leq \frac{2}{k+4} \left[\sum_{j=1}^{n} L_j \right] \cdot R_0^2(x_0) \tag{13}$$

While the worst-case convergence rate of full-gradient descent method is

$$f(x^k) - f^* \leq \frac{\max_i \{L_i\}}{k} R_1^2 \tag{14}$$

Still, the convergence rate of random coordinate descent is proportional to that one of full-gradient method.

Summary of Random Coordinate Descent Method

To sum up, the random coordinate descent method

- It has very cheap iteration cost.
- It adopts the random coordinate selection to ensure convergence
- Surprisingly enough, its convergence rate is almost same as the worst-case convergence rate of full-gradient descent method.

Summary

Motivation

- Full-gradient method is not favorable to solve *huge-scale* problem.
- Conventional coordinate descent method lacks theoretical justification

Random Coordinate Descent

- Random Coordinate Descent adopts random coordinate index selection
- Its convergence rate is almost same as the worst-case convergence rate of full-gradient method.
- The computational cost of each iteration of random coordinate descent method is much cheaper!

Future topic

In the future, we will discuss

- The convergence property when f is strongly convex
- Minimizing $f(x)$ with constrain
- Accelerating random coordinate descent method to $\mathcal{O}(\frac{1}{k^2})$

Thanks!


Dimitri P Bertsekas.
Nonlinear programming.
Athena scientific Belmont, 1999.


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


Peter Richtárik and Martin Takáč.
Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function.
Mathematical Programming, 144(1-2):1–38, 2014.