lions@epfl

EE-731: ADVANCED TOPICS IN DATA SCIENCES
LABORATORY FOR INFORMATION AND INFERENCE SYSTEMS
SPRING 2016

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

INSTRUCTOR: VOLKAN CEVHER

SCRIBER: JOEL CASTELLON AND DENYS LAMOTTE

VARIANCE REDUCTION TECHNIQUES AND COORDINATE DESCENT METHODS

Introduction: Coordinate descent methods (CD) have a long history in optimization and related fields. Mainly because they provide various practical advantages. First, they reduce to a sequence of easier optimization problems. In the simple version of CD this amounts to one dimensional optimization. Furthermore, coordinate descent methods are often simple to implement and their structure offers a favorable setup for parallel execution, an application that has drawn attention in recent years.

Outline: In this lecture, we first cover variance reduction techniques for stochastic gradient methods in section 1. We then mention the link with coordinate descent and delve into CD methods. This includes a description of the basic framework for CD in section 2 and main results in 3. Then we describe the class of functions we will work on in sections 4 and 9. We will also point out the link with classic methods in section 5. Finally, we survey applications and variants of CD such randomized, accelerated, proximal, parallel and primal-dual in sections in sections 6 – 13.

1 Proximal stochastic variance reduction (SPGD-VR)

We consider the following composite convex minimization problem:

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := \mathbb{E}[h(\mathbf{x},\theta)] + g(\mathbf{x})\}} \tag{1}$$

Where $f := \mathbb{E}[h(\mathbf{x},\theta)]$ and g are both proper, closed and convex. Let ∇f be L-Lipschitz continuous, g is possibly non-smooth and θ is a random vector whose distribution is supported on Θ.
Next, we consider the following instance of the problem:

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ F(\mathbf{x}) := \underbrace{\frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x})}_{f(\mathbf{x})} + g(\mathbf{x}) \Big\}$$

.
Here the solution set $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ is nonempty. Finally, recall that a prevalent choice (in SGD methods) of G is $G(\mathbf{x}^k, i_k) = \nabla f_{i_k}(\mathbf{x}^k)$ because computation of $\nabla f_{i_k}(\mathbf{x})$ is m times cheaper than $\nabla f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(\mathbf{x})$.

1.1 Algorithms and convergence

The technique consists in computing in a loop of q iterations the stochastic gradient $(\overline{G(\mathbf{x}^l, i_l)} = \nabla f_{i_l}(\mathbf{x}^l) - \nabla f_{i_l}(\mathbf{x}^k) + \nabla f(\overline{\mathbf{x}}^k))$ using the full-gradient computing at $\overline{\mathbf{x}}$ and just averaging after q iterations.

Proximal stochastic variance reduction (SPGD-VR)
1. Choose $\overline{\mathbf{x}}^0 \in \mathbb{R}^p, 0 \neq q \in \mathbb{N}$ and stepsize $\gamma > 0$.
2. For $k = 0, 1 \ldots,$ perform :
2a. $\nabla f(\overline{\mathbf{x}}^k) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(\overline{\mathbf{x}}), \mathbf{x}^0 = \overline{\mathbf{x}}^k$.
2b. For $l = 0, 1 \ldots, q-1$, perform :
$\begin{cases} \text{pick } i_l \in \{1,\ldots,m\} \text{ uniformly at random,} \\ \overline{G(\mathbf{x}^l, i_l)} = \nabla f_{i_l}(\mathbf{x}^l) - \nabla f_{i_l}(\mathbf{x}^k) + \nabla f(\overline{\mathbf{x}}^k), \\ \mathbf{x}^{l+}1 = prox_{\gamma h}(\mathbf{x}^l - \gamma\overline{G(\mathbf{x}^l, i_l)}). \end{cases}$
3. $\overline{\mathbf{x}}^{k+1} = \frac{1}{q}\sum_{l=1}^{q} \mathbf{x}^l$.

Theorem 1.1. *Mean of convergence of SPGD-VR. Nitanda, 2014*
Set $L_{max} = max_{1\leqslant i\leqslant m} L_i$, where L_i is Lipschitz constant of ∇f_i, and suppose that F is μ-strongly convex and that the stepsize satisfies :

$$\rho = \frac{1}{\mu\gamma(1-2L_{max}\gamma)q} + \frac{2L_{max}\gamma}{(1-2L_{max}\gamma)q} \leq 1 \tag{2}$$

Then,

$$\mathbb{E}\left[F(\overline{\mathbf{x}}^k) - F^*\right] \leqslant \rho^k \left(F(\overline{\mathbf{x}}^0) - F^*\right), \tag{3}$$

Proof. The proof is described in detail in [10]. □

To accelerate the algorithm we can choose s components instead of one for each iteration q and also add an acceleration step.

Accelerated mini-batch proximal stochastic variance reduction (Acc. MB SPGD-VR)
1. Choose $\overline{\mathbf{x}}^0 \in \mathbb{R}^p, 0 \neq q \in \mathbb{N}$ and stepsize $\gamma > 0$, accelerated stepsize $\beta = (1-\sqrt{\mu\gamma})/(1+\sqrt{\mu\gamma})$.
2. For $k = 0, 1 \ldots$, perform :
2a. $\nabla f(\overline{\mathbf{x}}^k) = \frac{1}{m}\sum_{i=1}^m \nabla f_i(\overline{\mathbf{x}}), \overline{\mathbf{x}} = \overline{\mathbf{x}}^k, \mathbf{x}^0 = y^1 = \overline{\mathbf{x}}$.
2b. For $l = 0, 1 \ldots, q-1$, perform :
$\begin{cases} \text{pick } I_l \subset \{1,\ldots,m\} : \text{mini-batch of size } s\ , \\ \overline{G(\mathbf{y}^l, I_l)} = \nabla f_{I_l}(\mathbf{y}^l) - \nabla f_{I_l}(\mathbf{x}^k) + \nabla f(\overline{\mathbf{x}}^k), \\ \mathbf{x}^{l+1} = prox_{\gamma h}(\mathbf{y}^l - \gamma \overline{G(\mathbf{y}^l, I_l)}), \\ \mathbf{y}^{l+1} = \mathbf{x}^{l+1} + \beta(\mathbf{x}^{l+1} - \mathbf{x}^l). \end{cases}$
3. $\overline{\mathbf{x}}^{k+1} = \mathbf{x}^q$.

Theorem 1.2. *Convergence of the mean for Acc. MB SPGD-VR*
Set $L_{max} = max_{1\leqslant i\leqslant m} L_i$, where L_i is Lipschitz constant of ∇f_i, and suppose that :

1. $0 \leq \gamma \leqslant \gamma_{max} = min\left\{\frac{(\alpha q)^2(m-1)^2\mu}{64(m-s)^2L_{max}^2}, \frac{1}{2L_{max}}\right\}$ *for some* $0 \leq \alpha \leq 1/8$.
2. $q \geqslant \frac{1}{(1-\alpha)\sqrt{\mu\gamma}} log\frac{1-\alpha}{\alpha}$.

Then,

$$\mathbb{E}\left[F(\overline{\mathbf{x}}^k) - F^*\right] \leqslant \rho^k \left(F(\overline{\mathbf{x}}^0) - F^*\right), \tag{4}$$

where $\rho = 2\alpha(2+\alpha)/(1-\alpha) \leq 1$.

Proof. Defining $\Phi_1(\mathbf{x}) = f(\mathbf{x}_1) + \frac{\mu}{2}\left\|\mathbf{x} - \mathbf{x}^0\right\|^2$
We have

$$\mathbb{E}\left[F(\overline{\mathbf{x}}^k) - F^*\right] \leqslant (1-\sqrt{\mu\gamma})^{k-1}(\Phi_1 - F)(\mathbf{x}^*) + \alpha(F(\mathbf{x}) - F(\mathbf{x}^*)) + \mathbb{E}\left[\sum_{l=1}^{k-1}(1-\sqrt{\mu\gamma})^{k-1-l}\alpha\sqrt{\mu\gamma}(F(\mathbf{x}^l) - F(\mathbf{x}^*))\right] \tag{5}$$

which is a lemma proved in [7]
We denote V_K the left part of the previous inequality and W_k the right one. We so have, for $k \geqslant 1$, $V_k \leqslant W_k$.
For $k \geqslant 2$,

$$W_k = (1-\sqrt{\mu\gamma})\left\{(1-\sqrt{\mu\gamma})^{k-2}(\Phi_1 - F)(\mathbf{x}^*) + \alpha V_1 + \sum_{l=1}^{k-2}(1-\sqrt{\mu\gamma})^{k-2-l}\alpha\sqrt{\mu\gamma}V_l\right\} + \alpha\sqrt{\mu\gamma}V_{k-1} + \alpha\sqrt{\mu\gamma}V_1 \leqslant (1-\sqrt{\mu\gamma}(1-\alpha))W_{k-1} + \alpha\sqrt{\mu\gamma}W_1 \tag{6}$$

Since $0 \leq \sqrt{\mu\gamma}(1-\alpha) \leq 1$, the above inequality leads to

$$W_k = \left((1-(1-\alpha)\sqrt{\mu\gamma})^{k-1} + \frac{\alpha}{1-\alpha}\right)W_1 \tag{7}$$

From the strong convexity of g (and f), we can see

$$W_1 = (1+\alpha)(F(\mathbf{x}) - F(\mathbf{x}^*)) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \leqslant (2+\alpha)(F(\mathbf{x}) - F(\mathbf{x}^*)) \tag{8}$$

Thus, for $k \geqslant 2$, we have

$$V_k \leqslant W_k \leqslant \left((1-(1-\alpha)\sqrt{\mu\gamma})^{k-1} + \frac{\alpha}{1-\alpha}\right)(2+\alpha)(F(\mathbf{x}) - F(\mathbf{x}^*)) \tag{9}$$

Moreover, as $log(1-\mathbf{p}) \leqslant -p$ and $q \geqslant \frac{1}{(1-\alpha)\sqrt{\mu\gamma}} log\frac{1-\alpha}{\alpha}$, we have

$$log(1-(1-\alpha)\sqrt{\mu\gamma})^q \leqslant -q(1-\alpha)\sqrt{\mu\gamma} \leqslant -log\frac{1-\alpha}{\alpha}, \tag{10}$$

and so

$$(1-(1-\alpha)\sqrt{\mu\gamma})^q \leqslant \frac{\alpha}{1-\alpha} \tag{11}$$

which proves the theorem. □

We see that these theorems of convergence allows us to have a constant stepsize in the algorithm SPGD-VR and moreover we obtain a linear rate convergence (with ρ as linear rate).

1.2 Taxonomy of algorithms

After looking at these different algorithms to solve our example of convex composite minimization, i.e,

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x}) + g(\mathbf{x})\right\} \tag{12}$$

We now summarize the rate of convergence of each of the mentioned Stochastic Gradient variants in order to obtain an $\epsilon-$ solution. Where such an approximate solution x_ϵ is defined as follows:
$f(x_\epsilon) - f(x^*) \leq \epsilon[max_X f - min_X f]$. We remind the following facts:

- $f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x})$ with $f\mu$-strongly convex with L-Lipschitz continuous gradient.
- $\kappa = \frac{L}{\mu}$
- $s_0 = \frac{8\sqrt{\kappa m}}{\sqrt{2}\alpha(m-1)+8\sqrt{\kappa}}$ for $0 \leq \alpha \leqslant 1/8$

Table 1: Rate of convergence

Gradient descent	Acc. MB SPGD-VR	SPGD-VR	SPGD
Linear	Linear	Linear	Sublinear

Table 2: Complexity to obtain ϵ-solution

SPGD-VR	Acc. MB SPGD-VR	Acc. Prox. Grad.
$O((m+\kappa)log(1/\epsilon))$	$O((m+\kappa\frac{m-s}{m-1})log(1/\epsilon))$	$O((m\kappa)log(1/\epsilon))$

A few remarks :

- $s = 1$: Acc. MB SPGD-VR has the same per-iteration cost as SPGD-VR since the size of the mini-batch is zero and it takes only one components for each q iteration.
- $s = m$: Acc. MB SPGD-VR has the same per-iteration cost as Acc. Prox. Grad. since it takes all the functions estimating f at each q iteration.

1.3 Subset of rows vs. Subset of columns

For the moment we mainly focused on using a subset of rows insted of the full data at each iteration like in Figure 1. We managed to compute an unbiased estimate $G(\mathbf{x}^l, i_k)$ of the gradient using :

- a subset of data points : $(\mathbf{a}_{i_k}, \mathbf{b}_{i_k})$,
- the whole decision variable $\mathbf{x}^k$:

$$G(\mathbf{x}^l, i_k) = \mathbf{a}_{i_k}^T(< \mathbf{a}_{i_k}^T, \mathbf{x} > -\mathbf{b}_{i_k}). \tag{13}$$

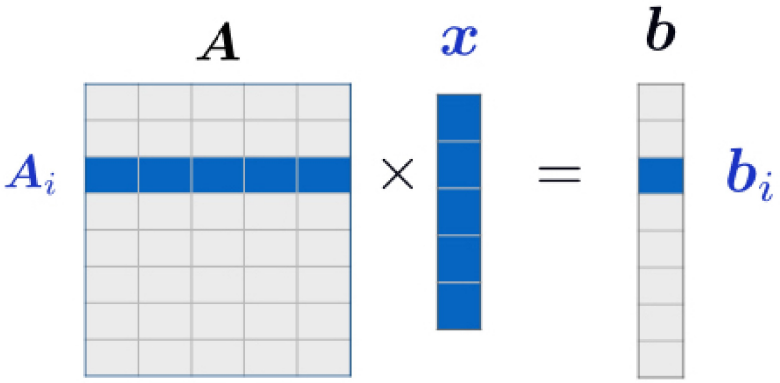


Figure 1: Example of Least squares. In the SGD setting, we want to sample by rows. Intuitively, the 'row geometry' yields a distribution for the output variable with the features (**x**) as the support. The idea of SGD is to sample from that distribution.

However, we can think about using a subset of columns at each iteration like in Figure 2. If we denote the basis vectors by $\mathbf{e}_i$, and the corresponding directional derivatives by ∇_i. We let $\mathbf{a}_i$ represent the ith column of matrix A and we consider the following unbiased estimate :

$$G(\mathbf{x}^l, i_k) = p\nabla_{i_k} f(\mathbf{x}^k)\mathbf{e}_{i_k} = p < \mathbf{a}_{i_k}, \mathbf{a}_{i_k}\mathbf{x}^k_{i_k} - \mathbf{b} > \mathbf{e}_{i_k}. \tag{14}$$

We so compute an unbiased estimate $G(\mathbf{x}^l, i_k)$ of the gradient using :

- a subset of columns ($\mathbf{a}_{i_k}$) and the whole measurement vector **b**,
- and only the chosen coordinates of decision variable : $\mathbf{x}^k_{i_k}$.

$$G(\mathbf{x}^l, i_k) = \mathbf{a}^T_{i_k}(< \mathbf{a}^T_{i_k}, \mathbf{x} > -\mathbf{b}_{i_k}). \tag{15}$$

Contrary to the unbiased estimate of gradient using rows which is full, in this case $G(\mathbf{x}^l, i_k)$ is sparse because only coordinates chosen by i_k are nonzero. Hence, we update these coordinates only.

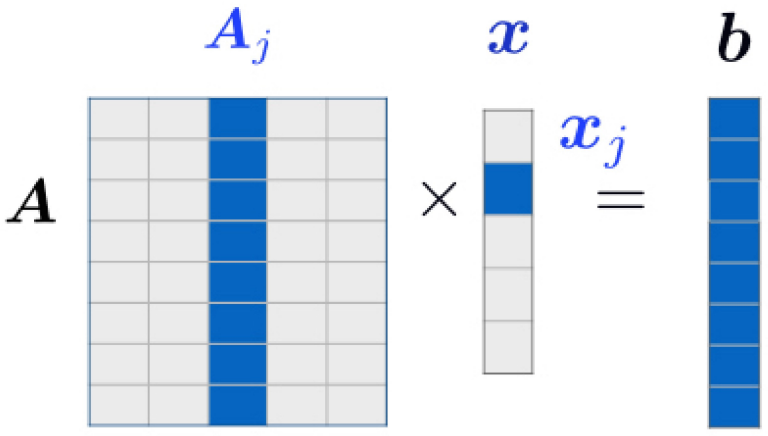


Figure 2: Example of Least squares. In the CD setting, we want to sample from the columns that span a lower dimensional space where most of the variance is present .

2 Basic coordinate descent framework

2.1 Relationship with stochastic gradient method

Randomized CD methods can be viewed as a special case of stochastic gradient methods, in which $G(\mathbf{x}^k, i_k) = p\nabla_{i_k} f(\mathbf{x}^k)\mathbf{e}_{i_k}$, where i_k is chosen uniformly random from $\{1, \ldots, p\}$, since,

$$\mathbb{E}[G(\mathbf{x}^k, i_k)] = p\mathbb{E}[\nabla_{i_k} f(\mathbf{x}^k)\mathbf{e}_{i_k}] = \sum_{i=1}^{p} \nabla_{i_k} f(\mathbf{x}^k)\mathbf{e}_{i_k} = \nabla f(\mathbf{x}^k).$$

Nonetheless, a proper theoretical analysis for CD is required because of the following distinctions. First, coordinate descent provides a descent lemma, so by properly choosing the step-size, we can guarantee $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$. Although, in some cases, variance of the gradient estimates can be characterized. As a simple example, variance shrinks to zero as we converge to $\mathbf{x}^\star$ in unconstrained smooth convex minimization. As we will see, coordinate descent is more than unbiased estimates. Theoretical analysis for CD shows that, properly constructed biased estimates may be favorable, this is because CD methods can take advantage of geometrical properties like the directional Lipschitz constants.

2.2 Unconstrained smooth minimization

We assume f is differentiable and the solution set is nonempty and bounded. Then the basic coordinate descent algorithm is as follows:

Choose $\mathbf{x}^0 \in \mathbb{R}^p$.
For $k = 0, 1, \ldots$ perform:
  Choose $i_k \in \{1, \ldots, p\}$
  Choose step-size γ_k
  Update $\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla_{i_k} f(\mathbf{x}^k)\boldsymbol{e}_{i_k}$

Despite the simplicity of the basic framework, there are many variants relying on different approaches for the selection of coordinates (i_k) and step size (γ_k).

2.3 Some variants of coordinate descent methods

Theory for coordinate descent methods moves rapidly, especially in recent years due to practical interest. We present above some of the main variants. While some of them have a theoretical analysis as support, others are merely motivated by practice.

As an overview of recent research on Coordinate recent we mention synchronous and asynchronous variants. First, we have the broad category of *one-at-a-time strategies* which follow the same setup of section 2.2. This means that after a coordinate update we use the updated iterate to calculate the next coordinate update. In contrast, *all-in-one strategies* do not use iterates that were generated before one complete cycle is complete. This is analogous to the difference between Gauss-Seidel and Jacobi methods in solving linear system (see 3).Theoretical support exists for the former strategy, while the latter is mainly adequate for parallel implementations of CD methods. Note, however, that *all-in-one strategies* may not converge.[11]

With respect to coordinate selection, we mention three basic strategies. First, the simplest strategy is the *cyclic* one e.g. $i_{k+1} = i_k + 1$ mod p+1. Second, we have have the *essentially cyclic* strategy where we touch each coordinate i at least once in each p iterations. Finally, randomized selection plays an important role in the literature (see [6]), in the simplest case we select i_k at random and independently at each iteration.

Concerning the step size selection. We can use a short step, γ_k is prescribed by global knowledge about properties of f. As in many other methods, line search chooses γ_k to approximately minimize f along coordinate direction i_k. And when possible, exact line search chooses γ_k to exactly minimize f along i_k.

3 Prior work on coordinate descent

Convergence of coordinatewise minimization for solving linear systems, the Gauss-Seidel method is a classic topic, see [5] for example or [2] for a modern treatment with respect to CD. An example is presented in section 5.

Nesterov in [6] considers randomized coordinate descent for smooth functions and shows that it achieves a rate $O(1/\epsilon)$ under Lipschitz gradient condition, and a rate $O\log(1/\epsilon)$ under strong convexity. Richtarik and Takac [9] extend and simplify these results, considering smooth plus separable functions, where now each coordinate descent update applies a proximal operator.

Beck and Tetruashvili [1] study cyclic coordinate descent for smooth functions in general. They show that it achieves a rate of $O(1/\epsilon)$ under a Lipschitz gradient condition, and a rate $O\log(1/\epsilon)$ under strong convexity. They also extend these results to a constrained setting with projections.

Finally, the general case of smooth plus separable function is not well-understood with respect to cyclic coordinate descent. It is also a question as to whether these two should behave similarly, and whether the aforementioned results are tight.

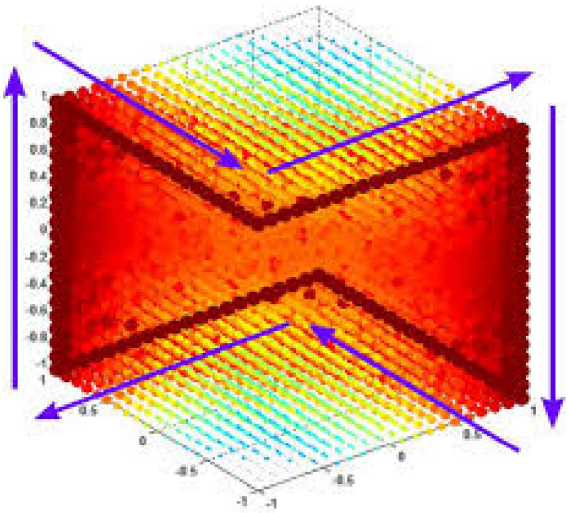


Figure 3: Red dots indicate the trajectory of CD when $\mathbf{x}^0$ is a vertex but not on either of the solutions. Figure from [11].

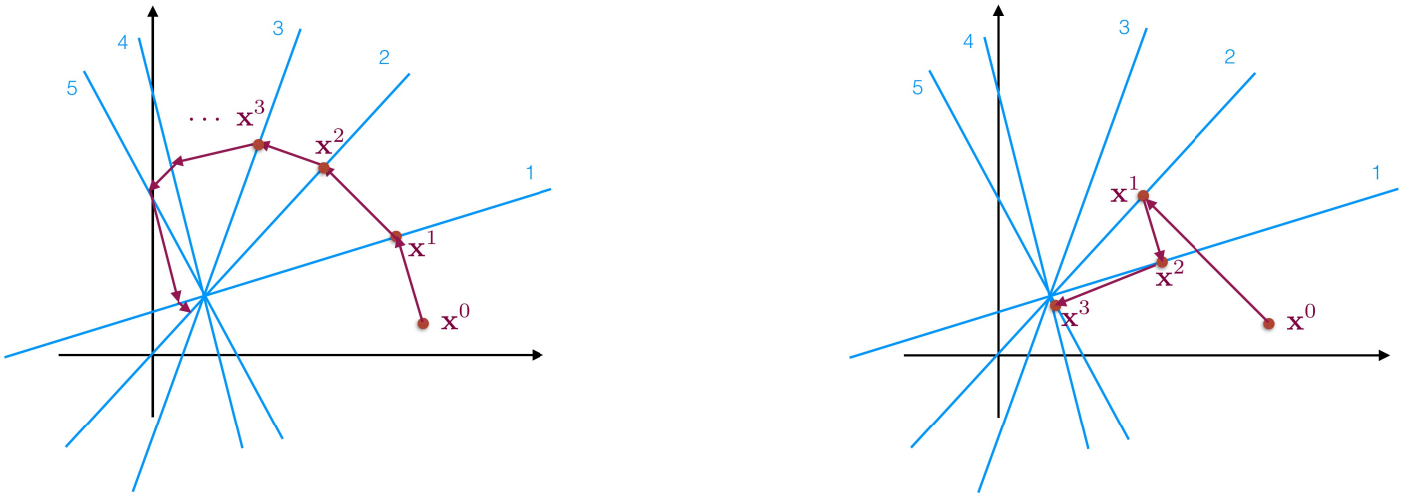
4 Cyclic CD does not always converge

4.1 Powell's example [8]

Consider the following non-convex, continuously differentiable function:
$f\colon \mathbb{R}^3 \to \mathbb{R} : f(x_1, x_2, x_3) = -(x_1x_2 + x_2x_3 + x_3x_1) + \sum_{i=1}^{3}(|x_i| - 1)_+^2$,
where $x_+^2 = \begin{cases} 0, & \text{if } x < 0, \\ x^2, & \text{if } x \geq 0. \end{cases}$
This function has two minimizers at vertices $(1, 1, 1)$ and $(-1, -1, -1)$ of the unit cube. Note that this is a non-convex problem, but can be solved by gradient descent. However, to illustrate the issue with CD consider the cyclic variant with exact minimization. Choose $\mathbf{x}^0$ near one of the vertices of the unit cube other than the solutions. Then, $\mathbf{x}^k$ cycles around the neighborhoods (see Fig.3) of six points that are close to the six non-optimal vertices [11].

5 Kaczmarz algorithm

Kaczmarz algorithm is a classical iterative method for solving linear systems of equations, $\mathbf{Ax} = \mathbf{b}$. A recent twist on this method [2] looks at the link with coordinate descent. Let us consider a consistent system (e.g., a system that admits a solution) such that $\mathbf{A} \in \mathbb{R}^{n \times p}$. We also have $\|\mathbf{a}_i\|_2 = 1$ for $i = 1, \ldots, p$, where $\mathbf{a}_i^T$ is the ith row of $\mathbf{A}$. Note that we can pre-process $\mathbf{A}$ to satisfy this property.



(a) Worst case behavior of cyclic variant.

(b) Randomized variant behaves better in expectation.

Figure 4: Illustration of cyclic vs randomized variants for Kaczmarz algorithm.

Then the algorithm is as follows:

Choose $\mathbf{x}^0 \in \mathbb{R}^p$
For $k = 0, 1, \ldots$ perform:
  Choose $i_k \in \{1, \ldots, n\}$ randomly with equal probability.
  Update $\mathbf{x}^{k+1} = \mathbf{x}^k - (\langle \mathbf{a}_{i_k}, \mathbf{x}^k \rangle - b_{i_k})\mathbf{a}_{i_k}$

Kaczmarz algorithm chooses a single equation from the system at each iteration (or a block of equations for the block Kaczmarz algorithm), and projects the current iterate to the solution space of this equation.

5.1 Kaczmarz algorithm and coordinate descent

We seek a least-norm solution of the system $\mathbf{Ax} = \mathbf{b}$:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x}\|_2^2, \text{ s.t. } \mathbf{Ax} = \mathbf{b}.$$

We can derive the Lagrange dual as follows:

The Lagrange function is $L(x, y) = \frac{1}{2}\|\mathbf{x}\|_2^2 - y^T A\mathbf{x} + y^T b$. Then, for $\mathbf{x}'$ that minimizes the Lagrange function with respect to $\mathbf{x}$ we have $\nabla_x L(\mathbf{x}', y) = 0$ which gives us $\mathbf{x}' = A^T y$. Next, $g(y) = L(\mathbf{x}', y) = \frac{1}{2}\|A^T y\|_2^2 - \|A^T y\|_2^2 + y^T b = -\frac{1}{2}\|A^T y\|_2^2 + y^T b$. Finally,

$$max_{\mathbf{y}\in\mathbb{R}^p}\ g(y) = \min_{\mathbf{y}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{A}^T\mathbf{y}\|_2^2 - \mathbf{b}^T\mathbf{y}.$$

The coordinate descent step on this dual formulation with step $\gamma_k = 1$ gives:

$$\mathbf{y}^{k+1} = \mathbf{y}^k - (\langle \mathbf{a}_{i_k}, \mathbf{A}^T\mathbf{y}^k \rangle - b_{i_k})\boldsymbol{e}_{i_k}.$$

When we multiply both sides by $\mathbf{A}^T$, we get:

$$\mathbf{A}^T\mathbf{y}^{k+1} = \mathbf{A}^T\mathbf{y}^k - (\langle \mathbf{a}_{i_k}, \mathbf{A}^T\mathbf{y}^k \rangle - b_{i_k})\mathbf{a}_{i_k}.$$

Finally, the change of variable $\mathbf{x}^k = \mathbf{A}^T\mathbf{y}^k$ yields Kaczmarz algorithm.

5.2 Kaczmarz algorithm cyclic vs randomized strategy

We want to emphasize that the convergence behavior depends heavily on the selection of i_k. In the worst case for the cyclic variant, the expected behavior is not well captured. However, the randomized variant performs better in expectation. We illustrate this in Fig.4

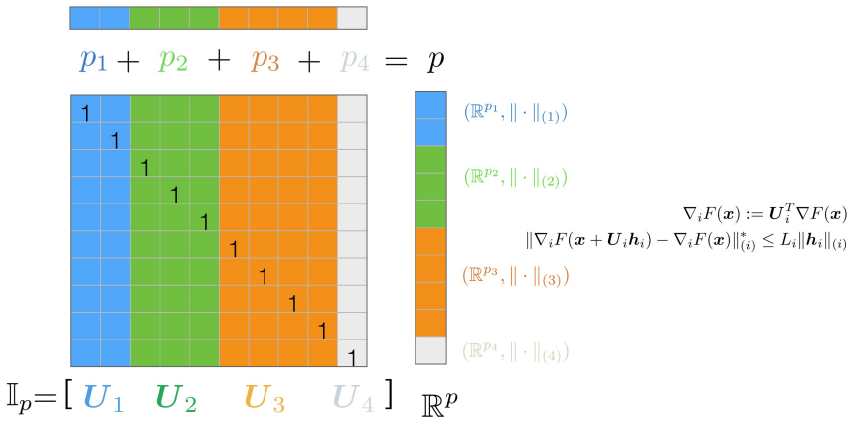


Figure 5: Randomized CD optimizes with respect to block of coordinates sampled by the Lipschitz along columns.

6 Randomized coordinate descent algorithm

CD algorithm is as follows:

Choose $\theta \in \mathbb{R}$ and $\mathbf{x}^0 \in \mathbb{R}^p$
For $k = 0, 1, \ldots$ perform:
  Choose $i_k = \mathcal{A}_\theta$
  Update $\mathbf{x}^{k+1} = \mathbf{x}^k - L_{i_k}^{-1} \boldsymbol{U}_{i_k} [\nabla_{i_k} f(\mathbf{x}^k)]^\#$

Where we define the *Sharp-operator* as $[\mathbf{x}]^\# = \arg\max_{\boldsymbol{s} \in \mathbb{R}^p} \langle \mathbf{x}, \boldsymbol{s} \rangle - (1/2)\|\boldsymbol{s}\|^2$. As an example, for the ℓ_2 norm we have $[\mathbf{x}]^\# = \mathbf{x}$. Then, we have $\mathcal{A}_\theta$ that generates $i \in \{1, \ldots, s\}$ with probability $L_i^\theta / \sum_{j=1}^s L_j^\theta$. This means that for $\theta = 0$ we get a uniform distribution.

Theorem 6.1 (Convergence of randomized CD. Nesterov, 2012.)**.** *The following theorem is due to Nesterov in [6], similar work is surveyed in [11]. We split the analysis in two cases according to convexity.*

1. *Without strong convexity:*

$$\mathbb{E}[f(\boldsymbol{x}^k) - f^\star] \leq \begin{cases} \frac{\sum_{j=1}^s L_j^\theta}{k+4} R_{1-\theta}^2(\boldsymbol{x}^0), & \ell_2 - norm, \\ \frac{s}{k+s}(R_1^2(\boldsymbol{x}_0)/2 + f(\boldsymbol{x}_0) - f^\star), & \theta = 0. \end{cases}$$

where $R_\theta(\boldsymbol{x}^0) = \max_{\{(x, x^\star) | f(x) \leq f(\boldsymbol{x}^0)\}} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_{[\theta]}$ *and* $\|\boldsymbol{x}\|_{[\theta]}^2 = \sum_{i=1}^s L_i^\theta \|\boldsymbol{x}_i\|_{(i)}^2$.

2. *With strong convexity: Suppose that f is strongly convex with respect to the norm $\|\cdot\|_{[1-\theta]}$ with convexity parameter $\mu_{1-\theta} > 0$. Then*

$$\mathbb{E}[f(\boldsymbol{x}^k) - f^\star] \leq \begin{cases} (1 - \mu_{1-\theta}/S_\theta)^k (f(\boldsymbol{x}^0) - f^\star), & \ell_2 - norm, \\ (1 - 2\sigma/(s(1+\sigma)))^k (R_1^2(\boldsymbol{x}_0) + f(\boldsymbol{x}_0) - f^\star), & \theta = 0. \end{cases}$$

where $S_\theta = \sum_{i=1}^s L_i^\theta$.

Recall that SPGM only gets the rate of $O(1/\sqrt{k})$ for non strongly convex problems and $O(1/k)$ for strongly convex problems. One needs the condition that the level set of f defined by $\mathbf{x}_0$ is bounded.

6.1 An illustration on the Least squares problem

Next, we demonstrate the performance of randomized coordinate the descent on the classic Least Squares problem:

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}$$

Choosing $\theta = 0$, so coordinates are chosen uniformly random. Then, $\mathbf{A} := \text{randn}(n, p)$ (e.g. standard gaussian). And, $\mathbf{x}^\natural \in \mathbb{R}^p$ with Gaussian i.i.d. entries, normalized to $\|\mathbf{x}^\natural\|_2 = 1$. Finally, $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w}$ is Gaussian white noise. SNR is 30dB.

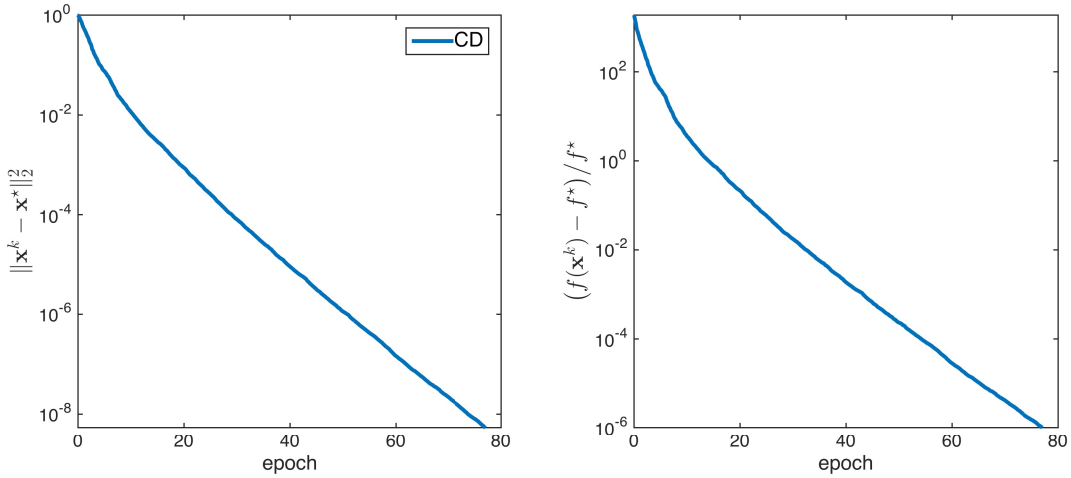


Figure 6: Least squares experiment for randomized coordinate descent.

7 Randomized accelerated CD

Randomized accelerated CD algorithm 1, RACD1:

Choose $\boldsymbol{v}^0 = \mathbf{x}^0 \in \mathbb{R}^p$, $a_0 = 1/s$, $b_0 = 2$.

For $k = 0, 1, \ldots$ perform:

Compute $\gamma_k \geq 1/s$ from equation $\gamma_k^2 - \frac{\gamma_k}{s} = (1 - \frac{\gamma_k \mu}{s})\frac{a_k^2}{b_k^2}$ and set $\alpha_k = \frac{s - \gamma_k \mu}{\gamma_k (s^2 - \mu)}$ and $\beta_k = 1 - \frac{\gamma_k \mu}{s}$.

Compute $\mathbf{y}^k = \alpha_k \boldsymbol{v}^k + (1 - \alpha_k)\mathbf{x}^k$.

Choose $i_k \in \{1, \ldots, s\}$ uniformly at random.

Update $\begin{cases} \mathbf{x}^{k+1} = \mathbf{y}^k - \frac{1}{L_{i_k}} \boldsymbol{U}_{i_k} [\nabla_{i_k} f(\mathbf{y}^k)]^{\#}, \\ \boldsymbol{v}^{k+1} = \beta_k \boldsymbol{v}^k + (1 - \beta_k)\mathbf{y}^k - \frac{\gamma_k}{L_{i_k}} \boldsymbol{U}_{i_k} [\nabla_{i_k} f(\mathbf{y}^k)]^{\#}. \end{cases}$

Update parameters $b_{k+1} = b_k / \sqrt{\beta_k}$ and $a_{k+1} = \gamma_k b_{k+1}$

Recall that s is the number of blocks, L_i the Lipschitz constant of $\nabla_i f$, and μ is the strong convexity constant of f. While the sharp operator is defined as $[\mathbf{x}]^{\#} = \arg\max_{s \in \mathbb{R}^p} \langle \mathbf{x}, s \rangle - (1/2)\|s\|^2$.

Theorem 7.1 (Convergence of RACD1. Nesterov, 2012)**.** *The following theorem is due to Nesterov in [6].*

$$\mathbb{E}[f(\boldsymbol{x}^k) - f^\star] \leq \Big(\frac{s}{k+1}\Big)^2 \Big[2\|\boldsymbol{x}^0 - \boldsymbol{x}^\star\|_{[1]}^2 + \frac{1}{s^2}(f(\boldsymbol{x}^0) - f^\star)\Big],$$

where $\|\boldsymbol{x}\|_{[1]} = \Big(\sum_{i=1}^s L_i \|\boldsymbol{x}_i\|_{(i)}^2\Big)^{1/2}$ *and* L_i *is Lipschitz constant of* $\nabla_i f$.

The expected complexity of RACD1 for finding an ε-solution is of the order $O\Big(\frac{s}{\sqrt{\varepsilon}} \max_{1 \leq i \leq s} L_i\Big)$ we emphasize that this bound depends on the dimension. In the next section we present another variant that is dimension independent. The next algorithm is motivated in situations when producing high accuracy solutions can become prohibitively time-consuming. This is a common scenario in large-scale problems where a much cheaper way to exploit the structure is preferred over accuracy.

8 Randomized accelerated CD: Dimension independence

Randomized accelerated CD algorithm 2 (RACD2)

Choose $\theta \in \mathbb{R}$, $\boldsymbol{v}^0 = \mathbf{x}^0 \in \mathbb{R}^p$, $a_0 = 1/s$, $b_0 = 1$, and $\sigma = \theta/2$

For $k = 0, 1, \ldots$ perform:

Choose $i_k = \mathcal{A}_\sigma$

Compute $\gamma_{k+1} > 0$ from equation $\gamma_{k+1}^2 S_\beta^2 = a_{k+1} b_{k+1}$ where $a_{k+1} = a_t + \gamma_{k+1}$ and $b_{k+1} = b_k + \mu_{1-\theta}\gamma_{k+1}$.

Compute $\alpha_k = \frac{\gamma_{k+1}}{a_{k+1}}$, $\beta_k = \frac{\mu_{1-\alpha} a_{k+1}}{b_{k+1}}$, $\mathbf{y}^k = \alpha_k \boldsymbol{v}^k + (1-\alpha_k)\mathbf{x}^k$.

Update

$$\begin{cases} \mathbf{x}^{k+1} = \mathbf{y}^k - \frac{1}{L_{i_k}} \boldsymbol{U}_{i_k} B_{i_k}^{-1} \nabla_{i_k} f(\mathbf{y}^k), \\ \boldsymbol{v}^{k+1} = \beta_k \mathbf{y}^k + (1-\beta_k)\boldsymbol{v}^k - \frac{\gamma_{k+1} \sum_{j=1}^s L_j^\sigma}{L_{i_k}^{1-\theta/2} b_{t+1}} \boldsymbol{U}_{i_k} B_{i_k}^{-1} \nabla_{i_k} f(\mathbf{y}^k). \end{cases}$$

The expected complexity of RACD2 for finding an ε-solution is of the order $O\Big(\frac{1}{\sqrt{\varepsilon}} \sum_{i=1}^s L_i^{1/2}\Big)$. In contrast to RACD1, this complexity is independent of the number of blocks.

Next, we demonstrate in Fig 7 the performance of RACD1 with uniform coordinate sampling and varying the strong convexity constant μ. The red curve performs on known μ We have taken vanilla coordinate descent as a baseline.

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}$$

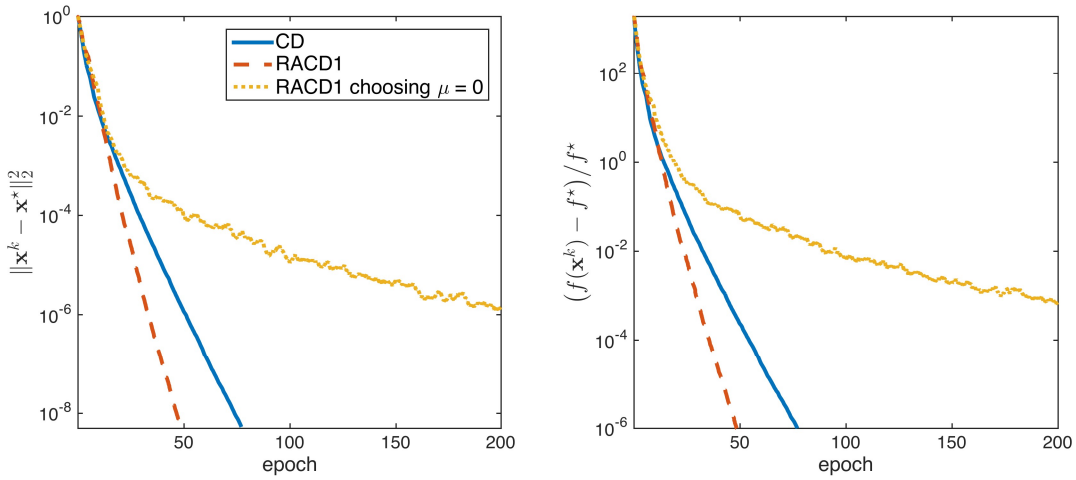


Figure 7: Experiment for the performance of RACD1 by varying strong convexity constant

For this experiment we set $\mathbf{A} :=$ randn(n, p) - standard Gaussian $\mathcal{N}(0, \mathbb{I})$, with $n = 1000$, $p = 500$. The parameters $\mathbf{x}^\natural \in \mathbb{R}^p$ with Gaussian i.i.d. entries, normalized to $\|\mathbf{x}^\natural\|_2 = 1$. Finally, we have $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w}$ is Gaussian white noise. SNR is 30dB.

From the results in Fig. 7 we observe the vanilla coordinate descent (CD) adapts to the strong convexity without requiring μ as an input. RACD requires μ to be known in advance. Otherwise, the convergence rate becomes sub-linear. Recall that this is also the case for gradient descent and its accelerated variants.

9 Coordinate descent for composite minimization problem

9.1 Composite convex minimization

Consider the following unconstrained composite convex minimization problem:

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$

Where f and g are both proper, closed, and convex. Further assumptions are that ∇f is L-Lipschitz continuous. While g is possibly non-smooth. Lastly, The solution set $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ is nonempty.

Nonetheless, the following examples illustrate the need of an additional assumption for CD to work for composite problems.

9.2 Coordinate descent does not always converge for composite convex problems

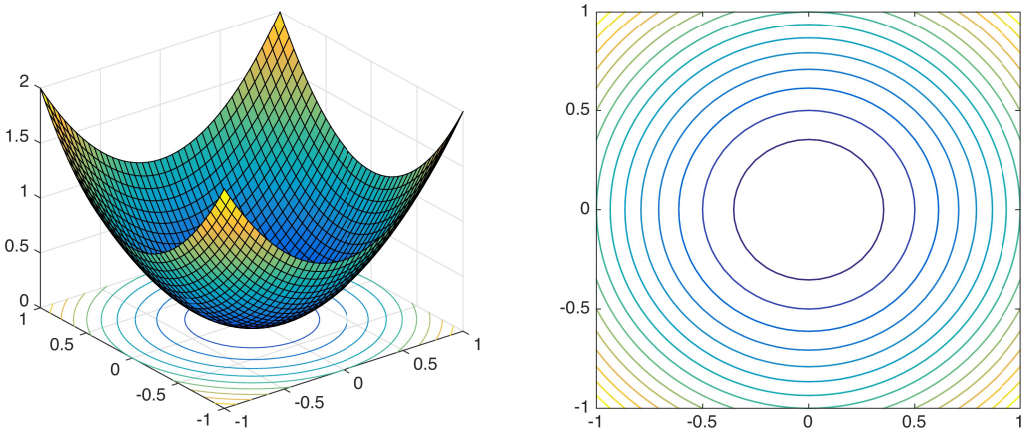


Figure 8: Smooth objective function: $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$

Consider the objective function: $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$. $f(\mathbf{x})$ is minimized along each coordinate axis, if and only if $\mathbf{x}$ is the global optimum (Fig.8). By the first order condition we get:

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = 0, \quad \text{for } i = 1, \ldots, p \quad \Longrightarrow \quad \nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_p}\right]^T = \mathbf{0}.$$

However, the natural question is whether we can consider the same optimality criterion for non-smooth $f(\mathbf{x})$.

Now, for composite (non-smooth) objective function (Fig.9 $F(\mathbf{x}) = \|\mathbf{x}\|_2^2 + |x_1 - x_2|$ the above criterion is no longer valid. Despite a unique global optimum at $(0, 0)$ we can still take points along the non-smooth axis that satisfy the first order condition mentioned above, consider point $(0.5, 0.5)$ for example.

Next, we illustrate the form of non-smooth composite functions for which the first order optimality criterion holds. It turns out that the key property to be able to use first order condition is that one term be *separable*. Consider the function $F(\mathbf{x}) = \|\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1$ (Fig.10).

Denote $f(\mathbf{x}) := \|\mathbf{x}\|_2^2$ the smooth part of $F(\mathbf{x})$ and $g(\mathbf{x}) := \|\mathbf{x}\|_1$ the non-smooth part. Assume that the non-smooth part is separable: $g(\mathbf{x}) = \sum_{i=1}^p g_i(x_i)$. We observe in Fig. 10 that the non-smooth axis are now aligned with the coordinate axis, making first order condition valid.

Then, $F(\mathbf{x})$ is minimized along each coordinate axis, if and only if $\mathbf{x}$ is the global optimum.

$$F(\hat{\mathbf{x}}) - F(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle + \sum_{i=1}^p [g(\hat{x}_i) - g(x_i)] \geq 0, \quad \forall \hat{\mathbf{x}} \in \mathbb{R}^p.$$

10 Coordinate descent for composite minimization problem

10.1 Composite convex minimization

Consider the following unconstrained composite convex minimization problem:

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$

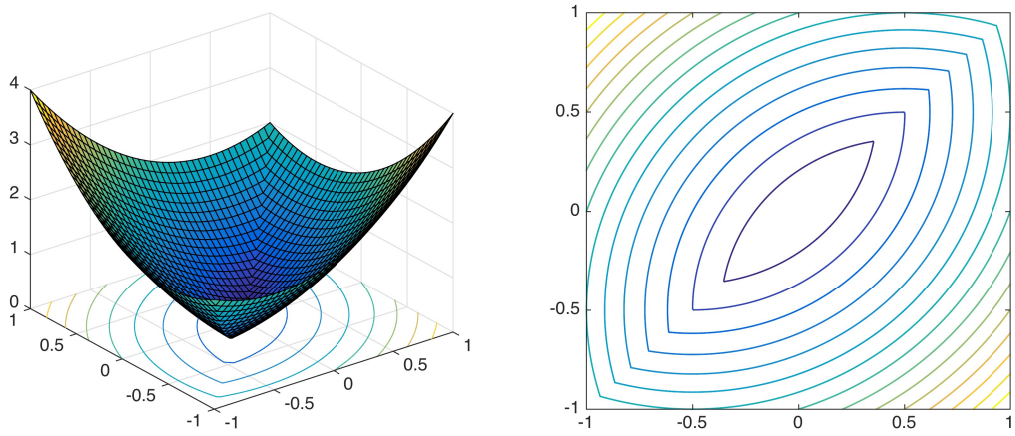


Figure 9: Objective function: $F(\mathbf{x}) = \|\mathbf{x}\|_2^2 + |x_1 - x_2|$

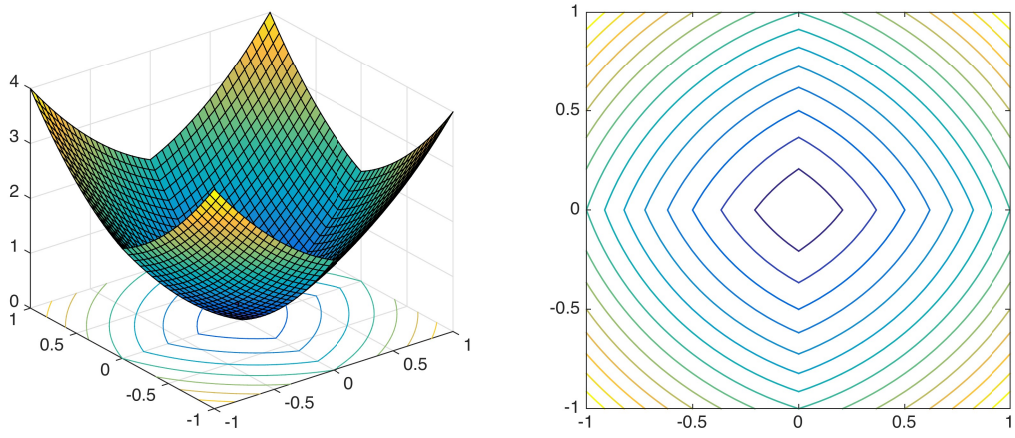


Figure 10: Non-smooth function: $F(\mathbf{x}) = \|\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1$

- f and g are both proper, closed, and convex.
- ∇f is L-Lipschitz continuous.
- g is possibly non-smooth.
- $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\} \neq \emptyset$.

As mentioned in section 9.2 we need $g(\mathbf{x})$ be separable. That is, $g(\mathbf{x}) = \sum_{i=1}^{p} g_i(x_i)$, where $g_i \colon \mathbb{R} \to \mathbb{R}$ for all i. This form includes unconstrained optimization ($g(\mathbf{x})$ = constant), box constrained: $g(\mathbf{x}) = \sum_{i=1}^{s} \mathbb{1}_{[a_i,b_i]}(x_i)$ and ℓ_q norm regularization: $g(\mathbf{x}) = \|\mathbf{x}\|_q^q$ where $q \geq 1$.

We can also extend to g being block-separable: $p \times p$ identity matrix can be partitioned into column sub-matrices $\boldsymbol{U}_i$, $i = 1, \ldots, s$ such that $g(\mathbf{x}) = \sum_{i=1}^{s} g_i(\boldsymbol{U}_i^T \mathbf{x})$.. Block-separable examples include group-sparse regularizers.

10.2 Composite convex problems with separable g

Next, we mention relevant examples of problems in statistics, machine learning and optimization where the framework we presented below, composite convex with a separable term, is applicable.

The following problems are ubiquitous and withing such context coordinate descent was proven quite useful. This is, perhaps, the motivation behind seeking a more formal understanding of CD methods.

10.2.1 LASSO

$$\min_{\mathbf{x}} \underbrace{\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2}_{f(\mathbf{x})} + \underbrace{\lambda \|\mathbf{x}\|_1}_{g(\mathbf{x})}.$$

10.2.2 Support vector machine (SVM) with squared hinge loss

$$\min_{\mathbf{x}} \underbrace{C \sum_i \max\{y_i(\mathbf{w}_i^T \mathbf{x} - b), 0\}^2}_{g(\mathbf{x})} + \underbrace{\frac{1}{2}\|\mathbf{x}\|^2}_{f(\mathbf{x})}.$$

10.2.3 SVM: dual form with bias term

$$\min_{0 \leq \mathbf{x} \leq C\mathbb{1}} \underbrace{\frac{1}{2} \sum_{i,j} x_i x_j y_i y_j \boldsymbol{K}(\mathbf{w}_i, \mathbf{w}_j)}_{f(\mathbf{x})} - \underbrace{\sum_i x_i}_{g(\mathbf{x})}.$$

10.2.4 Logistic regression with ℓ_q norm regularization

$$\min_{\mathbf{x}} \underbrace{\frac{1}{p} \sum_i \log(1 + \exp(-b_i \mathbf{w}_i^T \mathbf{x}))}_{g(\mathbf{x})} + \underbrace{\lambda \|\mathbf{x}\|_q^q}_{f(\mathbf{x})}.$$

10.2.5 Semi-supervised learning with Tikhonov regularization

$$\min_{\mathbf{x}} \underbrace{\sum_{i \in \{\text{labeled data}\}} (\mathbf{x}_i - \mathbf{y}_i)^2}_{g(\mathbf{x})} + \underbrace{\lambda \mathbf{x}^T \boldsymbol{L} \mathbf{x}}_{f(\mathbf{x})}.$$

10.2.6 Relaxed linear programing

$$\min_{\mathbf{x} \geq 0} \boldsymbol{c}^T \mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \;\Rightarrow\; \min_{\mathbf{x} \geq 0} \underbrace{\boldsymbol{c}^T \mathbf{x}}_{g(\mathbf{x})} + \underbrace{\lambda \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2}_{f(\mathbf{x})}.$$

11 Randomized proximal coordinate descent algorithm

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := f(\mathbf{x}) + \sum_{i=1}^{s} g_i(\mathbf{x}_i) \right\}$$

Randomized proximal coordinate descent algorithm (PCD):

Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k\in\mathbb{N}} \in]0, +\infty[^{\mathbb{N}}$.

For $k = 0, 1, \ldots$ perform:

Pick $i_k \in \{1, \ldots, s\}$ uniformly at random.

Update coordinate i_k:

$$\mathbf{x}_{i_k}^{k+1} = \arg\min_{v\in\mathbb{R}^{p_{i_k}}} \; g_i(v) + \left\langle v, \nabla_{i_k} f(\mathbf{x}^k) \right\rangle + \frac{1}{2\alpha_k}\|v - \mathbf{x}_{i_k}^k\|_2^2.$$

Theorem 11.1 (Linear convergence rate [11]). *Suppose that f is uniformly Lipschitz continuously differentiable and strongly convex with modulus $\mu > 0$. Set $L_{\max} = \max_{1\le i\le s} L_i$ and suppose that $\alpha_k = 1/L_{\max}$ for all k. Then*

$$\mathbb{E}[F(x^k) - F^\star] \le \left(1 - \frac{\mu}{sL_{\max}}\right)^k (F(x^0) - F^\star).$$

Recall that we need the following condition: the level set of F defined by $\mathbf{x}_0$ is bounded.
Next, we illustrate performance for proximal coordinate descent for the LASSO problem (Fig.).

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 : \mathbf{x} \in \mathbb{R}^p \right\}$$

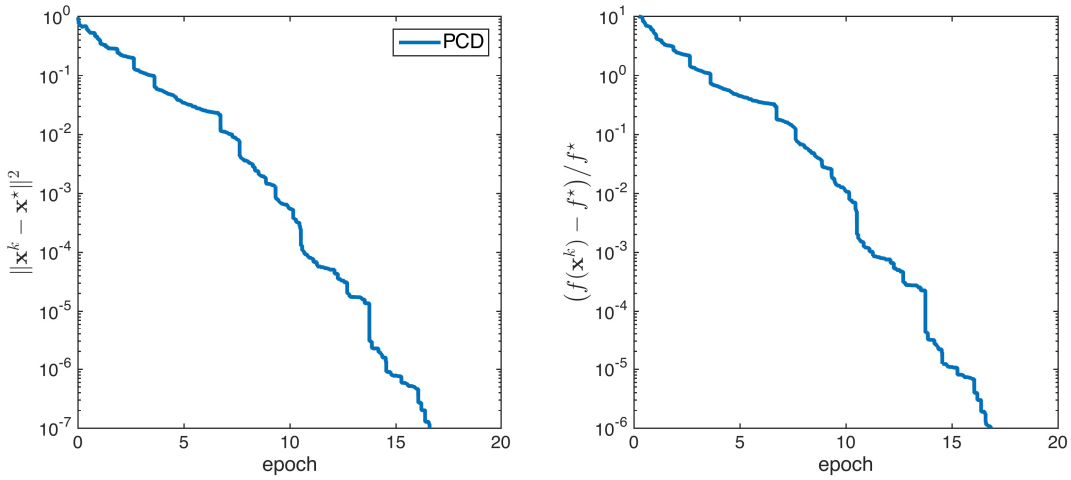


Figure 11: Proximal coordinated descent applied to the LASSO problem.

Synthetic problem setup

- $\mathbf{A}$:= randn(n, p) - standard Gaussian $\mathcal{N}(0, \mathbb{I})$, with $n = 1000$, $p = 500$.
- $\mathbf{x}^\natural \in \mathbb{R}^p$ is 50-sparse with Gaussian i.i.d. entries, normalized to $\|\mathbf{x}^\natural\|_2 = 1$.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w}$ is Gaussian white noise. SNR is 30dB.
- $\theta = 0$, so coordinates are chosen uniformly random.
- $\lambda := 10^{-2}$.

12 Accelerated parallel proximal block-coordinate descent algorithm

• Randomized block-coordinate descent algorithm:

- Strong convexity.
- Level sets are bounded.
- Slow convergence if $L_{\max}$ is large.

• No strong convexity: choose a group of blocks + acceleration.

Accelerated parallel proximal coordinate descent algorithm (APPROX) $\hat{\mathcal{S}} \subset \{1, \dots, s\}$: uniform block sampling, $\tau = \mathbb{E}[|\hat{\mathcal{S}}|]$, $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_s) \in \mathbb{R}^s_+$.

Choose $\boldsymbol{v}^0 = \mathbf{x}^0 \in \mathbb{R}^p$ and $\alpha_0 = \tau/s$.

For $k = 0, 1, \dots$ perform:

Compute $\mathbf{y}^k = (1 - \alpha_k)\mathbf{x}^k + \alpha_k \boldsymbol{v}^k$.

Generate a random set of blocks $\mathcal{S}_k \sim \hat{\mathcal{S}}$.

For $i \in \mathcal{S}_k$, perform:

$$\boldsymbol{v}_i^{k+1} = \operatorname*{arg\,min}_{\boldsymbol{v} \in \mathbb{R}^{p_i}} \{ \left\langle \boldsymbol{v} - \mathbf{y}_i^k, \nabla_i f(\mathbf{y}^k) \right\rangle + \frac{s\alpha_k \sigma_i}{2\tau} \|\boldsymbol{v} - \boldsymbol{v}_i^k\|^2_{(i)} + g_i(\boldsymbol{v}) \}.$$

Update: $\mathbf{x}^{k+1} = \mathbf{y}^k + (s\alpha_k)/\tau(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)$.

Update parameters: $\alpha_{k+1} = (\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2)/2$.

• Uniform block sampling means $(\forall (i, j) \in \{1, \dots, s\}^2) \quad \boldsymbol{P}(i \in \hat{\mathcal{S}}) = \boldsymbol{P}(j \in \hat{\mathcal{S}})$.
• Algorithm can be executed in parallel.

12.1 $O(1/k^2)$ rate convergence

Theorem 12.1 ([4]). *Suppose that the expected separable over-approximation holds, i.e.,*

$$\mathbb{E}[f(\boldsymbol{x} + \boldsymbol{h}_{\hat{\mathcal{S}}})] \le f(\boldsymbol{x}) + \frac{\tau}{s}\Big(\langle \boldsymbol{h}, \nabla f(\boldsymbol{x}) \rangle + \frac{1}{2}\|\boldsymbol{h}\|^2_{\sigma} \Big), \tag{16}$$

where $\boldsymbol{h}_{\hat{\mathcal{S}}} = \sum_{i \in \hat{\mathcal{S}}} \boldsymbol{U}_i h_i$ *and* $\|\boldsymbol{h}\|^2_{\sigma} = \sum_{i=1}^s \sigma_i \|\boldsymbol{h}_i\|^2_{(i)}$. *Then*

$$\mathbb{E}[F(\boldsymbol{x}^k) - F^\star] \le \frac{4s^2}{((k-1)\tau + 2s)^2}\Big(\Big(1 - \frac{\tau}{s}\Big)(F(\boldsymbol{x}^0) - F^\star) + \frac{1}{2}\|\boldsymbol{x}^0 - \boldsymbol{x}^\star\|^2_{\sigma} \Big).$$

• Condition (16) means that the function $\mathbf{x} \mapsto f(\mathbf{x} + \boldsymbol{h}_{\hat{\mathcal{S}}})$ has τ/s-Lipschitz continuous gradient in $\|\cdot\|_{\sigma}$.

13 Coordinate descent primal-dual algorithm

13.1 Composite minimization problem with linear operator

Consider the following composite minimization problem with linear operator

$$\min_{\mathbf{x} \in \mathbb{R}^p} \{ f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{A}\mathbf{x}) \}.$$

- f, g and h are proper, closed and convex.
- ∇f is Lipschitz continuous.
- $\mathbf{A} \in \mathbb{R}^{q \times p}$.

This problem can be transformed to finding saddle points of the Lagrangian function

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{x}) + \langle \mathbf{y}, \mathbf{A}\mathbf{x} \rangle - h^\star(\mathbf{y}),$$

where $h^\star : \mathbf{y} \mapsto \sup_{\mathbf{z}} \langle \mathbf{y}, \mathbf{z} \rangle - h(\mathbf{z})$ is the Fenchel-Legendre transform of h.

13.2 Examples

13.2.1 Total variation + ℓ_1 regularized least squares regression

$$\min_{\mathbf{x}\in\mathbb{R}^p} \underbrace{\frac{1}{2}\|\boldsymbol{M}\mathbf{x}-\mathbf{b}\|_2^2}_{f(\mathbf{x})} + \underbrace{\alpha r\|\mathbf{x}\|_1}_{g(\mathbf{x})} + \underbrace{\alpha(1-r)\|\mathbf{A}\mathbf{x}\|_{2,1}}_{h(\mathbf{A}\mathbf{x})}.$$

13.2.2 Dual SVM

$$\min_{\mathbf{x}\in\mathbb{R}^p} \underbrace{\frac{1}{2\lambda}\|\mathbf{A}D(b)\mathbf{x}\|_2^2 - \boldsymbol{e}^T\mathbf{x}}_{f(\mathbf{x})} + \underbrace{\sum_{i=1}^{p}\iota_{[0,C_i]}(x_i)}_{g(\mathbf{x})} + \underbrace{\iota_{\mathbf{b}^T}(\mathbf{x})}_{h(\mathrm{Id}\,\mathbf{x})}.$$

Coordinate descent primal-dual algorithm

Choose $\sigma = (\sigma_1,\dots,\sigma_q)$, $\tau = (\tau_1,\dots,\tau_p)$, $\mathbf{x}^0 \in \mathbb{R}^p$, $\mathbf{y}^0 \in \mathbb{R}^q$ and initialize

$$\begin{cases} (\forall i \in \{1,\dots,p\}) & \mathbf{w}_i^0 = \sum_{j\in\mathbb{J}(i)} \mathbf{A}_{ji}^T \mathbf{y}_0^j(i). \\ (\forall j \in \{1,\dots,q\}) & \mathbf{z}_j^0 = (1/m_j)\sum_{i\in\mathbb{I}(j)} \mathbf{y}_j^0(i). \end{cases}$$

For $k = 0, 1, \dots$ perform:

Choose $i_k \in \{1,\dots,p\}$ at random and uniformly.

Compute:

$$\begin{cases} \bar{\mathbf{y}}^{k+1} = \mathrm{prox}_{\sigma h^\star}(\mathbf{z}^k + D(\sigma)\mathbf{A}\mathbf{x}^k) \\ \bar{\mathbf{x}}^{k+1} = \mathrm{prox}_{\tau g}(\mathbf{x}^k - D(\tau)(\nabla f(\mathbf{x}^k) + 2\mathbf{A}^T\bar{\mathbf{y}}^{k+1} - \mathbf{w}^k)). \end{cases}$$

Update: For $i = i_{k+1}$ and for each $j \in \mathbb{J}(i_{k+1})$:

$$\begin{cases} \mathbf{x}_i^{k+1} = \bar{\mathbf{x}}_i^{k+1} \\ \mathbf{y}_j^{k+1}(i) = \bar{\mathbf{y}}_j^{k+1}(i) \\ \mathbf{w}_i^{k+1} = \mathbf{w}_i^k + \sum_{j\in\mathbb{J}(i)} \mathbf{A}_{ji}^\star(\mathbf{y}_j^{k+1}(i) - \mathbf{y}_j^k(i)) \\ \mathbf{z}_j^{k+1} = \mathbf{z}_i^k + \frac{1}{m_j}(\mathbf{y}_j^{k+1}(i) - \mathbf{y}_j^k(i)). \end{cases}$$

Otherwise

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k,\ \mathbf{w}_i^{k+1} = \mathbf{w}_i^k,\ \mathbf{z}_j^{k+1} = \mathbf{z}_j^k,\ \mathbf{y}_j^{k+1}(i) = \mathbf{y}_j^{k+1}(i).$$

Remarks • At iteration k, quantities $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ do not need to be explicitly computed and only the coordinates

$$\bar{\mathbf{x}}_{i_{k+1}}^{k+1} \quad \text{and} \quad \bar{\mathbf{y}}_j^{k+1},\ j \in \mathbb{J}(i_{k+1})$$

are needed to perform the update.

• When g is separable, the other coordinates do not need to be computed.

Convergence results

Theorem 13.1 ([3]). *Suppose that*

1. *For every $i \in \{1,\dots,p\}$, there exists $\beta_i \geq 0$ such that*

$$(\forall \boldsymbol{x} \in \mathbb{R}^p)(\forall \boldsymbol{u} \in \mathbb{R}^{p_i}) \quad f(\boldsymbol{x} + \boldsymbol{U}_i\boldsymbol{u}) \leq f(\boldsymbol{x}) + \langle \boldsymbol{U}_i\boldsymbol{u}, \nabla f(\boldsymbol{x})\rangle + \frac{\beta_i}{2}\|\boldsymbol{u}\|_{(i)}^2.$$

2. *For every $i \in \{1,\dots,p\}$,*

$$\tau_i < \frac{1}{\beta_i + \rho(\sum_{j\in\mathbb{J}(i)} m_j\sigma_j\mathbf{A}_{ji}^T\mathbf{A}_{ji})},$$

where $\rho(\boldsymbol{B})$ is the spectral radius of $\boldsymbol{B}$.

Then

1. $\boldsymbol{x}^k \to \boldsymbol{x}^\star$.
2. $\boldsymbol{y}_j^k(i) \to \boldsymbol{y}_j^\star$ *for every* $j \in \{1,\dots,q\}$ *and every* $i \in \mathbb{I}(j)$.

References

[1] A.Beck and L.Tetruashvili. On the convergence of block coordinate descent type methods. 2013.

[2] A.Ramdas. Rows vs columns for linear systems of equations-randomized kaczmarz or coordinate descent? 2014.

[3] O. Fercoq and P. Bianchi. A coordinate descent primal-dual algorithm with large step size and possibly non separable functions,. 2015. http://arxiv.org/abs/1508.04625.

[4] O. Fercoq and P. Richatárik. Accelerated, parallel and proximal coordinate descent,. *SIAM. J. Optim.*, 25:1997–2023, 2016.

[5] Gene Howard Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins studies in the mathematical sciences. The Johns Hopkins University Press, 1996.

[6] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization. *SIAM. J. Optim.*, 22:341–362, 2012.

[7] A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. *NIPS*, pages 1574–1582., 2014.

[8] M.J.D. Powell. On search directions for minimization algorithms. *Math. Program.*, 4:193–201., 1973.

[9] P.Richtarik and M. Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. 2011.

[10] L. Xiao Q. Lin, Z. Lu. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM. J. Optim.*, 25:2244–2273, 2015.

[11] S. J. Wright. Coordinates descent algorithms,. *Math. Program.*, 151:3–34., 2015.